Cloud Modeling for Computer Graphics

Gustav Taxén

February 23, 1999

Abstract

Realistic rendered images of clouds are important in architectural design, weather visualization, flight simulation, film, art and entertainment. The extremely varied visual appearance of clouds and the complexity of the meteorological and optical processes that create clouds makes modeling and rendering difficult. This thesis presents a new visually-orientated technique for creating realistic images of convective clouds. The new technique is simple and very fast compared to physically-based methods, which makes it suitable for use on desktop workstations. The thesis also contains an overview of the meteorology and optics of clouds and a review of previous modeling and rendering techniques.

Molnmodellering för datorgrafik

Referat

Det är viktigt att ha tillgång till realistiska datorgenererade bilder av moln inom arkitekturdesign, vädervisualisering, flygsimulering, film, konst och underhållning. Moln är dock svåra att modellera och rendera eftersom deras utseende varierar kraftigt. Samtidigt är det svårt att simulera de komplicerade fysiska processer som ger uppkomst till moln. I den här rapporten presenteras en ny, visuellt orienterad metod för att skapa realistiska bilder av cumulusmoln. Den nya tekniken är enkel och effektiv jämfört med simuleringsbaserade metoder, vilket gör att den kan utnyttjas på persondatorer. Rapporten innehåller också en översikt över de delar av meteorologin och optiken som rör moln samt en genomgång av tidigare använda renderingsmetoder.

Preface

Many thanks to my advisor Lars Kjelldahl for always being supportive, helpful and willing to give up precious time for our meetings.

Thanks to Robert Sigg, Gunilla Svensson and Harald Lejenäs at the Department of Meteorology, Stockholm University, and Jesper Oppelstrup at the Department of Numerical Analysis and Computer Science, KTH, who gave me valuable advice and support.

I would also like to thank my family for their support and patience.

Contents

Chapter 1

Introduction

Clouds have fascinated and vexed computer graphics researchers for many years. The visual appearance of clouds is very complex and extremely varied, yet it is very easy to recognize an "incorrect" cloud model, probably because we see clouds in one form or another every day. This makes clouds extremely difficult to model and render.

There are two main approaches to cloud modeling and rendering, *physically-based techniques* and *ontogenetic techniques*. The physically-based techniques attempt to simulate the meteorological processes that create clouds and the interaction between light and cloudy air. Ontogenetic techniques try to capture the visual appearance of clouds without simulating the actual physical processes.

Truly realistic generated images of outdoor scenes require that clouds are present. Rendered images of clouds are important in architectural design, weather visualization, flight simulation, film, art and entertainment.

Chapter 2 deals with the modeling of clouds. Section 2.1 contains an overview of the meteorological processes in the atmosphere that give rise to clouds, while Section 2.2 reviews previous ontogenetic modeling methods and presents a new technique for modeling convective clouds.

Chapter 3 is about cloud rendering. Section 3.1 gives an overview of the interaction between light and cloudy air and reviews previous techniques for simulating this interaction. Section 3.2 contains a new, simple and efficient ontogenetic approach to cloud rendering.

Chapter 4 contains images created by combining the new modeling and rendering techniques.

Chapter 5 discusses the main results of the thesis and presents some ideas for future research and Chapter 6 concludes and summarizes the thesis.

Chapter 2

Cloud modeling

The purpose of a model of an entity is to allow people to visualize and understand the structure or behaviour of the entity [10]. For clouds and gases the model is often implemented as a *density function*, $\rho(\vec{x})$, $\vec{x} \in \mathbf{R}^3$ that, for each point in space, evaluates to the amount of cloud matter that exists at that point. We shall refer to *cloud modeling* as the task of defining such a density function. What should be regarded as cloud matter is an issue that is open for interpretation. Often, ρ is assumed to be a measure of the amount of cloud water or the number of cloud droplets per unit volume that exist at a point in space.

Section 2.1 is an overview of the physical processes that creates clouds. Section 2.2 gives an overview of ontogenetic, non-physically-based cloud models and presents a new and efficient modeling technique for convective clouds.

2.1 Physically-based cloud modeling

When modeling a natural phenomenon for computer graphics, it often makes sense to try to simulate the physical processes that cause the phenomenon. For atmospheric clouds, this implies that such things as atmospheric composition and stability, thermodynamics and droplet formation by condensation are taken into consideration. These subjects are studied and described by the science of cloud physics, a branch of physical meteorolgy. A good introduction to cloud physics that we shall follow in this section can be found in [40].

Cloud formation is a phenomenon that takes place on a wide range of scales, from global weather circulation patterns that are more than 1000 km in size to individual cloud elements in the range from about 1 to 100 km. The cloud elements consist of individual droplets or ice crystals that range from 10 to 100 micrometers in radius. The processes on the micrometer scale influence the processes on larger scales.

Convective clouds are formed in warm, moist air that is chilled in ascent. When the air temperature drops beyond a certain point, the water vapor starts to condense on microscopic particles and forms small droplets. Given the right conditions, these droplets can continue to grow by condensation and coagulation and may eventually become large enough to fall out of the cloud as rain drops.

Clouds are normally classified according to their visual appearance from the ground by using an internationally adopted system that was proposed by Luke Howard in 1803. It consists of four major cloud types: *cumulus* (clouds with vertical development), *stratus* (clouds in flat-appearing layers), *cirrus* (fibrous or hair-like) and *nimbus* (rain clouds). The complete classification consists of many sub-types that includes combinations of the main types and clouds notable for their development. [19]

2.1.1 Thermodynamics of dry and moist air

In meteorology, it is common to treat air as a mixture of two ideal gases: *dry air* and *water vapor*. The mixture is called *moist air* and its properties are obtained by combining the thermodynamic behaviours of dry air and water vapor.

The equation of state for dry air

It is possible to express the relationship between pressure, p, volume, V and temperature, T, of an ideal gas in thermal equilibrium. This relationship is usually called the *equation of state* and can be written as

$$pV = R^*T \tag{2.1}$$

where R^* is the *universal gas constant*.[1] If we denote the mass of the gas by M, the molecular weight of the gas by m, and the specific volume of the gas by $\alpha = V/M$, (2.1) can also be written

$$p\alpha = R'T \tag{2.2}$$

where $R' = R^*/m$ is called the *individual gas constant*.

The first law of thermodynamics

The first law of thermodynamics states that heat is a form of energy and that energy is conserved. The second statement can be expressed in the following way: of the total amount of heat, dq, added to a unit mass of gas, some tends to increase the energy of the gas by the amount du and the remainder will cause an amount of work, dw, to be done by the gas. In other words,

$$dq = du + dw \tag{2.3}$$

[1] Constants, symbols and units are listed in the appendices.

The work done by the gas can be written as

$$dw = pd\alpha \tag{2.4}$$

For ideal gases, an increase in internal energy appears as a change in temperature according to

$$dT = \frac{1}{c}dq \tag{2.5}$$

where c is called the *specific heat capacity*. c is not constant but depends on whether work is done while the heat is added. If no work is done, $dw = 0$, which, by (2.4), implies $d\alpha = 0$, and we can define the *specific heat at constant volume* as

$$c_v = \left(\frac{dq}{dT}\right)_\alpha \tag{2.6}$$

Similarly, we define the *specific heat at constant pressure* as

$$c_p = \left(\frac{dq}{dT}\right)_p \tag{2.7}$$

Combining (2.3), (2.4) and (2.5),

$$dq = c_v dT + pd\alpha \tag{2.8}$$

(2.8) can also be written as

$$dq = c_p dT - \alpha dp \tag{2.9}$$

The adiabatic process

We can use the first law of thermodynamics to describe a number of different processes. An important process in meteorology is a process where the gas does not exchange heat with its environment, that is, $dq = 0$ and from (2.8) and (2.9),

$$c_v dT = -pd\alpha \tag{2.10}$$

$$c_p dT = \alpha dp \tag{2.11}$$

This process is called the *adiabatic process*. Many of the temperature changes in the atmosphere can be approximated as adiabatic.

By combining the equation of state with (2.9),

$$c_p dT = R'T\frac{dp}{p} \tag{2.12}$$

which can be integrated to give

$$\left(\frac{T}{T_0}\right) = \left(\frac{p}{p_0}\right)^k \tag{2.13}$$

where $k = R'/c_p$. (2.13) is called *Poisson's equation for adiabatic processes* and we define the *potential temperature*, θ, of dry air as

$$\theta = T\left(\frac{100 \text{ kPa}}{p}\right)^k \tag{2.14}$$

θ is the temperature that a parcel of dry air would have if, starting at temperature T and pressure p, it was adiabatically expanded or compressed to a pressure of 100 kPa. In any adiabatic process, θ is constant.

The equation of state for water vapor

We now turn to the thermodynamic properties of water vapor. The equation of state for water vapor is

$$e = \varrho_v R_v T \tag{2.15}$$

where e is the vapor pressure, ϱ_v is the vapor density and R_v is the individual gas constant for water vapor. It can also be written

$$e = \varrho_v \frac{R'}{\varepsilon} T \tag{2.16}$$

where $\varepsilon = R'/R_v$.

Saturation vapor pressure

Consider a water surface. A stream of molecules are breaking free from the surface as vapor molecules. This process is called *evaporation*. The rate of evaporation depends on temperature - an increase in temperature yields an increased rate of evaporation. At the same time, the reversed process, *condensation*, is taking place. When equilibrium is reached and the rates of evaporation and condensation are equal, the air above the surface is said to be *saturated* and the partial pressure due to water vapor is called the *saturation vapor pressure*, e_s. It turns out that e_s depends only on temperature and is described by a differential equation known as the *Clausius-Clapeyron equation*. An approximation of the saturation vapor pressure is

$$e_s(T) = Ae^{-B/T} \tag{2.17}$$

where $A = 2.53 \cdot 10^8$ kPa and $B = 5.42 \cdot 10^3$ K.

Water vapor content in moist air

The water vapor content in moist air can be described in several ways:

- Partial pressure of the water vapor, e.
- Density of the water vapor, ϱ_v. ϱ_v is also called *absolute humidity*.
- *Mixing ratio*, w, the mass of water vapor per unit mass of dry air. $w = M_v/M_d = \varrho_v/\varrho_d$. From the respective equations of state, we have $\varrho_v = e/R_vT$ and $\varrho_d = (p-e)/R'T$, and we can write

$$w = \varepsilon \frac{e}{p-e} \approx \varepsilon \frac{e}{p} \tag{2.18}$$

- *Specific humidity*, q, the mass of water vapor per unit mass of moist air.

$$q = \frac{\varrho_v}{\varrho_d + \varrho_v} = \varepsilon \frac{e}{p-(1-\varepsilon)e} \approx \varepsilon \frac{e}{p} \tag{2.19}$$

- *Relative humidity*, f.

$$f = \frac{w}{w_s} \approx \frac{e}{e_s} \tag{2.20}$$

  The relative humidity is often expressed in per cent.

- *Virtual temperature*, T_v, the temperature of dry air having the same density as a sample of moist air at the same pressure.

$$T_v = T\left(\frac{1+w/e}{1+w}\right) \approx T(1+0.6w) \tag{2.21}$$

  We can apply the equation of state for dry air to moist air if we replace the temperature with the virtual temperature.

We define the *saturation mixing ratio* and *saturation specific humidity* by replacing e by e_s in the definitions above.

Saturation temperatures

If the vapor mixing ratio and pressure of an air sample is held constant while the temperature is decreased, the air will eventually reach saturation. The temperature at which this happens is called the *dew point temperature* and is denoted by T_d. An approximation for the dew point temperature follows directly from (2.17) and (2.18):

$$T_d(w,p) = \frac{B}{\ln(A\varepsilon/wp)} \tag{2.22}$$

When air is cooled by radiation close to the ground and the wind is weak, the air may reach saturation and form a cloud, a *fog*. Fogs can also form when warm air is moved to an area where the ground is cold or by air mixing.

Consider a sample of air that is cooled adiabatically with w held constant. The temperature at which saturation is reached is called the *isentropic condensation temperature* and is denoted by T_c. If we denote the pressure at which saturation is reached by p_c we can write $T_c = T_d(w, p_c)$. We can then write Poisson's equation (2.13) as

$$\frac{T_c}{T_0} = \left(\frac{p_c}{p_0}\right)^k \Leftrightarrow p_c = p_0 \left(\frac{T_c}{T_0}\right)^{1/k} \tag{2.23}$$

and combine to get

$$T_c = B/\ln\left(\frac{A\varepsilon}{wp_0}\left(\frac{T_0}{T_c}\right)^{1/k}\right) \tag{2.24}$$

Equation (2.24) is solved by iteration.

The pseudoadiabatic process

If a parcel of air is ascending adiabatically in the atmosphere and its temperature drops beyond the isentropic condensation point, the water vapor in the parcel will start to condense and latent heat is released. The latent heat will warm the air and the temperature will decrease at a lower rate than before. This process is called the *pseudoadiabatic process*. If it is assumed that all condensed water is removed from the cloud immediately, it is possible to write an approximation of the dependence of T on p as

$$\frac{dT}{T} = k\frac{dp}{p} - \frac{L}{Tc_p}dw_s \tag{2.25}$$

where L is the *latent heat of vaporization*. L depends on temperature and is tabulated in appendix A. It is possible to use equation (2.17) and the approximation for w_s to solve for dT/dp (see appendix B). The result is

$$\frac{dT}{dp} = \frac{\left(\frac{k}{p} + \frac{L\varepsilon Ae^{-B/T}}{Tc_p p^2}\right)}{\left(\frac{1}{T} + \frac{L\varepsilon ABe^{-B/T}}{T^3 c_p p}\right)} \tag{2.26}$$

(2.26) can be integrated numerically to give the temperature as a function of pressure. Figure 2.1 shows the result of such an integration. The integration starts at 261.83° K and 50 kPa.

2.1.2 Cloud droplets

It is not obvious that condensation should start when moist air reaches saturation. In clean supersaturated air, water molecules must collide to form embryonic droplets on which the condensation can take place. Theory and

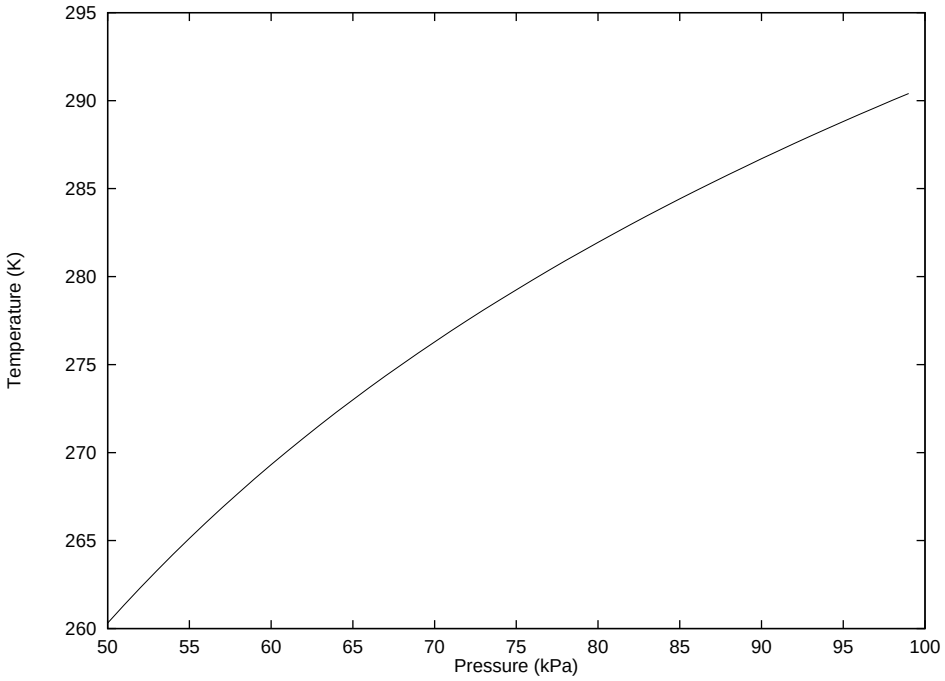

Figure 2.1: Pseudoadiabatic process.

experiments show that this requires relative humidities of several hundred percent. In the atmosphere, condensation starts as soon as the air exceeds the saturation level, and the relative humidity rarely exceeds 2%. The reason is that particles are present on which droplets can form. The particles are usually called *condensation nuclei* and their sizes range from about 10^{-3} micrometers to over 10 micrometers. Common sources of condensation nuclei include wind-generated dust, sea spray, forest fires, combustion and other industrial operations.

Critical droplet radius

We define the *saturation ratio* as

$$S = \frac{e}{e_s} \tag{2.27}$$

A cloud droplet that is forming on a condensation nucleus is stable until its radius reaches a critical value r^*. Before reaching r^*, the droplet radius follows changes in saturation ratio gently and the droplet is in equilibrium with its environment. After the drop radius reaches the critical value the growth process is unstable. The growth still follows changes in saturation ratio, but the droplet will deviate further and further from the equilibrium value.

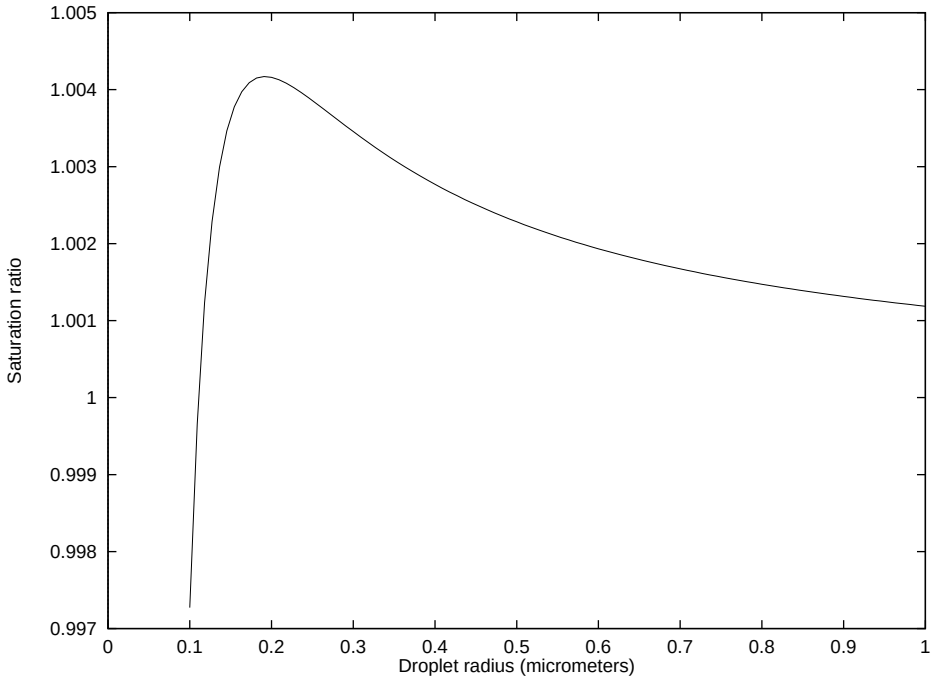

Figure 2.2: Saturation ratio as a function of droplet radius.

It can be shown that, for a given temperature T, nuclei mass M and nuclei molecular weight m_s, the relationship between saturation ratio and cloud droplet radius is

$$S(r) = \left(1 - \frac{b}{r^3}\right) e^{a/r} \tag{2.28}$$

where $a \approx 3.3 \cdot 10^{-5}/T$ (cm) and $b \approx 8.6M/m_s$ (cm^3). Figure 2.2 shows the saturation ratio as a function of radius for a droplet forming on a NaCl nucleus of mass 10^{-16} g. The temperature is 273° K. The critical radius r^* is at about 0.19 micrometers. Before the critical radius is reached, the droplet is in equilibrium with its surroundings and an increase in saturation ratio will cause the droplet to grow until it reaches equilibrium again. If the saturation ratio is increased beyond the peak of the curve at r^*, the droplet will become unstable and continue to grow without the need for further increase in saturation ratio. The droplet is said to be *activated* and will eventually grow large enough to become a *cloud drop*. Stable droplets with radius less than r^* are called *haze particles*. Activated droplets do not grow uncontrollably since there are always many droplets present that compete for the available water vapor.

An approximate expression for r^* is

$$r^* = \sqrt{3b/a} \tag{2.29}$$

and an approximate expression for the saturation ratio at which the critical radius is reached is

$$S^* = 1 + \sqrt{4a^3/27b} \tag{2.30}$$

Droplet growth

The rate of growth of a single, activated droplet can be shown to be approximately

$$r\frac{dr}{dt} = \frac{(S-1) - \dfrac{a}{r} + \dfrac{b}{r^3}}{\left(\dfrac{L}{R_v T} - 1\right)\dfrac{L\varrho_L}{KT} + \dfrac{\varrho_L R_v T}{De_s}} \tag{2.31}$$

where ϱ_L is the density of water, K is the *coefficient of thermal conductivity of air* and D is the *dynamic viscosity of air*. K and D depend on temperature and are tabulated in appendix A.

If we assume that the water vapor in a developing cloud is provided by saturated air that is cooled in ascent and is lost by condensation on growing droplets, it is possible to derive an expression for the rate of change of S,

$$\frac{dS}{dt} = \frac{1}{T}\left(\frac{\varepsilon L g}{R' c_p T} - \frac{g}{R'}\right)\frac{dz}{dt} - \frac{p}{R'T}\left(\frac{R'T}{\varepsilon e_s} + \frac{\varepsilon L^2}{pTc_p}\right)\frac{d\chi}{dt} \tag{2.32}$$

where dz/dt is the vertical air velocity, $d\chi/dt$ is the rate of condensation and g is the acceleration due to gravity. By coupling (2.31) and (2.32), it is possible to simulate the growth of a droplet population in a developing cloud. The simulation procedure is

1. Calculate the height z from vertical air velocity.
2. Obtain pressure p at height z from, for example, the ICAO Standard Atmosphere tables (see appendix A).
3. Calculate the temperature T by making use of (2.26) or the pseudo-adiabatic temperature lapse rate (see Section 2.1.3).
4. For each droplet in the population, obtain the new radius from (2.31).
5. Use the change in droplet radius to estimate $d\chi/dt$ and obtain the new saturation ratio from (2.32).
6. Repeat from step 1.

Figures 2.3 and 2.4 show the results of such a simulation. The altitude at time $t = 0$ was 2000 m above mean sea level, dz/dt was kept constant at 0.15 m/s, initial temperature, pressure and saturation ratio were, respectively, 260.31° K, 80 kPa and 1.0. The droplet population was assumed to consist of 10^8 identical specimens with an initial radius of 0.75 μm.

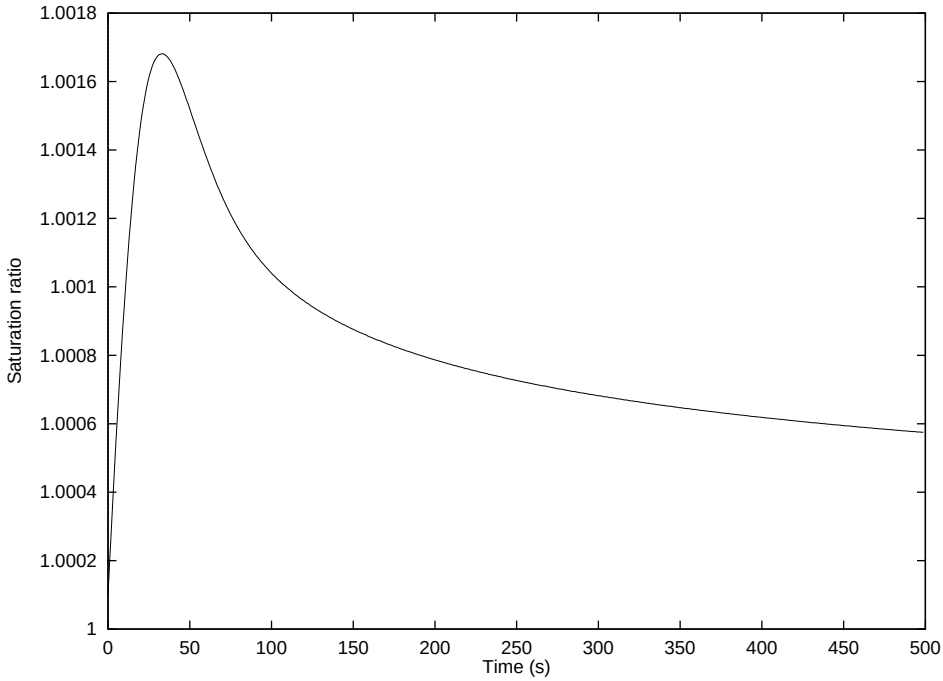

Figure 2.3: Saturation ratio as a function of time in a droplet growth simulation.

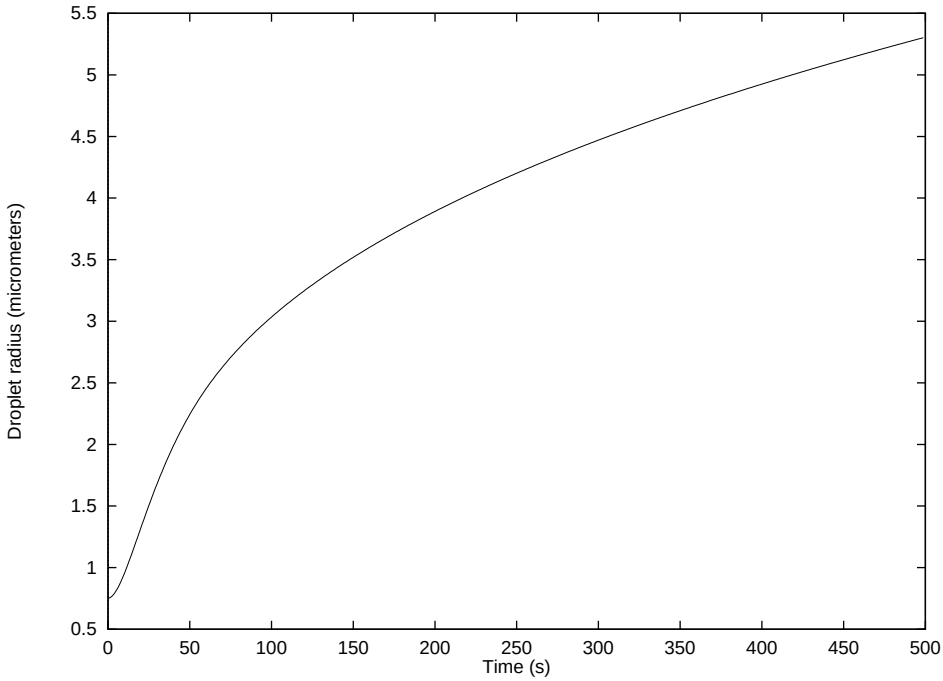

Figure 2.4: Droplet radius as a function of time in a droplet growth simulation.

The saturation ratio increases sharply while the droplet radius is small but eventually settles down towards a constant value. The droplet growth rate decreases as the saturation ratio moves towards the constant value. Note that since the saturation ratio controls the number of droplets that become active, the entire cloud drop population is determined in the early stages of cloud formation.

2.1.3 Atmospheric stability

Air is a mixture of several gases, mainly nitrogen, oxygen and argon, and different solid and liquid particles. The proportional amounts are close to constant up to an altitude of about 90 km. Variable amounts of water vapor, carbon dioxide and ozone are also present.

The atmosphere is divided into layers with different temperature variation characteristics. The layer between ground level and approximately 12 km is called the *troposphere* and is the only layer in which convective clouds can form. The mean temperature lapse rate in the troposphere is about $0.6 - 0.7°\mathrm{C}$ / 100 m, but limited regions where the temperature increases with height, *inversions*, can be present.

Temperature lapse rates

A parcel of air is said to be in *hydrostatic equilibrium* when the vertical pressure gradient force on the air exactly balances the force of gravity, that is,

$$\frac{\partial p}{\partial z} = -\frac{pg}{R'T} \tag{2.33}$$

(2.33) is called the *hydrostatic equation*.

Consider a parcel of air that is ascending adiabatically in the atmosphere. By differentiating (2.12) with respect to altitude z, we get

$$\frac{dT}{dz} = \frac{R'}{c_p}\frac{T}{p}\frac{dp}{dz} \tag{2.34}$$

If we assume that the parcel pressure immediately adjusts to the ambient pressure, we have from the hydrostatic equation

$$\frac{dp}{dz} = \frac{\partial p}{\partial z} = -\frac{pg}{R'T'} \tag{2.35}$$

where T' is the ambient temperature. Thus,

$$\frac{dT}{dz} = -\frac{g}{c_p}\frac{T}{T'} \tag{2.36}$$

If we assume that $T/T' \approx 1$, we can write

$$\frac{dT}{dz} \equiv -\Gamma \approx \frac{g}{c_p} \approx -10^{-2}\ \mathrm{K/m} \tag{2.37}$$

Γ is called the *dry adiabatic lapse rate* and denotes the rate at which temperature falls off with height in the dry adiabatic process. Note that Γ is a process lapse rate and should not be confused with the temperature lapse rate in the (ambient) atmosphere.

It is possible to derive a corresponding expression for the pseudoadiabatic lapse rate by differentiating (2.25) with respect to height and using the hydrostatic equation and the Clausius-Clapeyron equation. The result is

$$\Gamma_s \equiv -\frac{dT}{dz} = \Gamma \frac{1 + \dfrac{Lw_s}{R'T}}{1 + \dfrac{L^2 \varepsilon w_s}{R' c_p T^2}} \tag{2.38}$$

Buoyancy

An air parcel that is warmer than its surroundings tends to accelerate upwards. It can be shown that the buoyant force (i.e. the upward force) per unit mass is

$$F_B = g\left(\frac{T - T'}{T'}\right) \tag{2.39}$$

From (2.39) it follows that if the ambient lapse rate γ is lower than Γ, an air parcel that has been displaced upwards will experience a restoring force and the situation is said to be *stable*. Conversely, if γ is greater than Γ the parcel is accelerated upwards and the situation is said to be *unstable*. When $\gamma = \Gamma$ the situation is *neutral*.

F_B is not the only force acting on the parcel. Horizontal forces such as the Coriolis force (the centrifugal force due to the Earth's rotation) and wind forces from high pressures to low pressures will displace the parcel slantwise. Vertical forces due to the burden of condensed water and aerodynamic resistance tend to reduce the vertical acceleration of the parcel. In convective clouds, mixing of air at the parcel boundaries is taking place and the assumption of adiabatic ascent does not hold. Also, the atmosphere is mass continuous: air is neither created nor destroyed anywhere, which implies that air must descend somewhere to replace air that is lifted vertically. The ascending air parcel may have its temperature affected by air descending around it. Thus, F_B may be considered as an upper limit of the force acting on the parcel.

Movements in the atmosphere

Vertical air movements can be *settled* or *unsettled*. A settled air movement is a gentle vertical movement of a horizontal air layer. As a result of the low velocity, the rate of condensation is low. There are two main types of unsettled movements. *Dynamic turbulence* is caused by friction and an

uneven ground level when air is flowing along the ground. *Convection* is a process where the air closest to the ground is heated and rises in the form of bubbles or plumes. Figure 2.5 illustrates settled and unsettled air movements.

From (2.39) it is clear that inversions act as a barrier for vertical air movements. Often, only strong updrafts are able to penetrate an inversion layer. As a consequence, clouds growing vertically that reach an inversion will often show horizontal development. This is illustrated in figure 2.6.

The horizontal flow over high terrain may force the air layers closest to the ground to reach saturation and form cumulus-like clouds. If the winds are strong, the clouds may show substantial vertical development with heavy precipitation. Figure 2.7 illustrates this situation.

2.1.4 Numerical cloud models

Cloud development is a process that takes place on a wide range of scales and involves microphysics, thermodynamics and dynamics. A useful tool for examining the formation of clouds is numerical simulation. The equations used for the simulation vary depending on the application, but most systems share some basic features: Newton's law of motion applied to air, a continuity equation for air, equations for temperature and conservation of water in its different states (solid, liquid and vapor). Depending on the application and available computer resources, the movement of air is described in one, two or three dimensions.

A large number of papers with different numerical models has been published. A small selection includes [15], [45], [46], [16], [17], [4], [52] and [44]. Very few of the publications include implementation details.

In the field of computer graphics, a well-known paper by Kajiya and Von Herzen [20] includes a meteorological model of cumulus convection. I have not been able to find any other computer graphics papers that use meteorological models.

It would appear that a three-dimensional cloud model such as [44] is suitable for generating clouds for computer graphics applications. Unfortunately, the heavy computational requirements makes it necessary to use a low spatial resolution for the simulation grid (on the order of 1 km for a grid covering $50 \times 50 \times 10$ km in [44]). As a consequence, essential visual information is missing in the resulting clouds. Obviously a meteorological simulation could be used to provide a rough cloud shape that could be improved in a later step by using carefully selected random noise. However, it is one of the findings in this thesis that spending time solving the system of equations is unnecessary since the simple modeling techniques presented in the next section can provide realistic cloud shapes even on a desktop workstation.

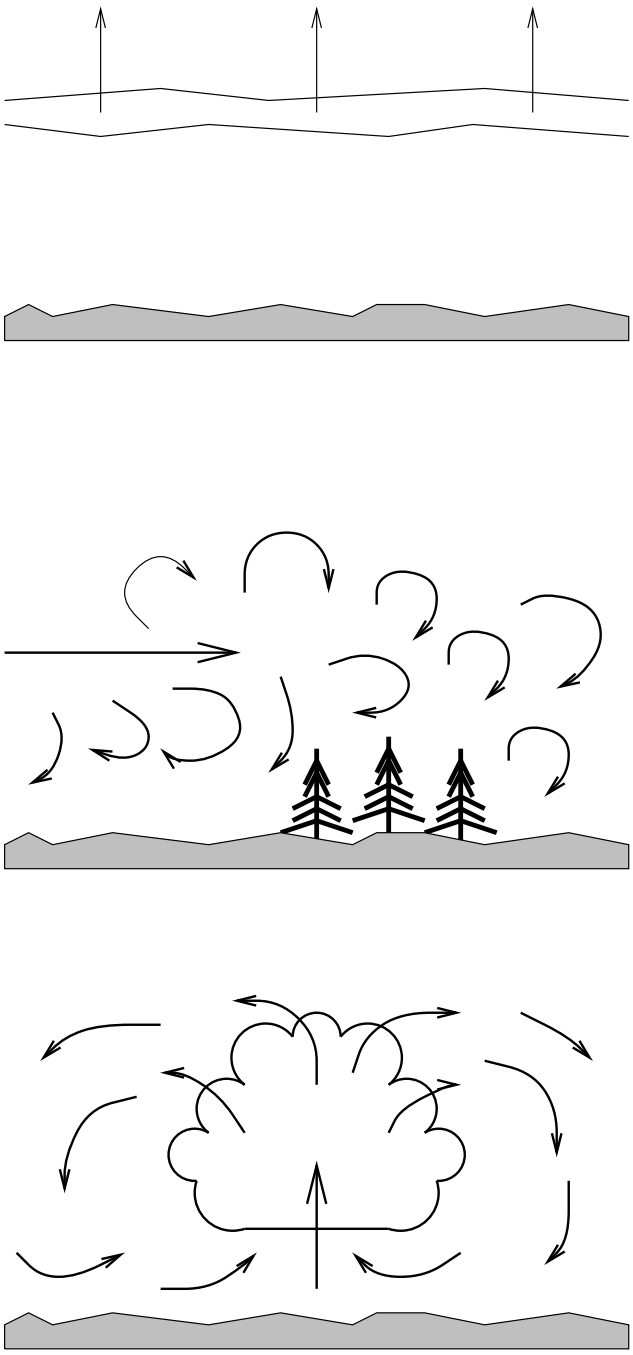

Figure 2.5: Main types of vertical movement in the atmosphere: settled movement (top), dynamic turbulence (middle) and convection (bottom). (After [24].)

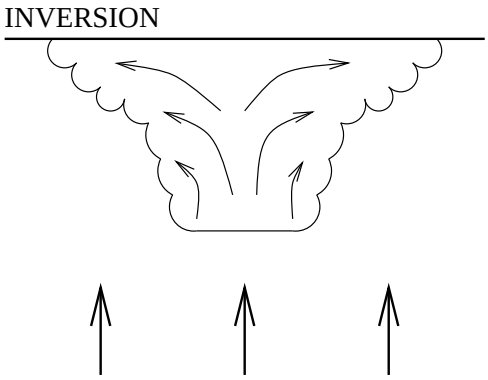

Figure 2.6: Horizontal cloud development due to an inversion layer. (After [24].)

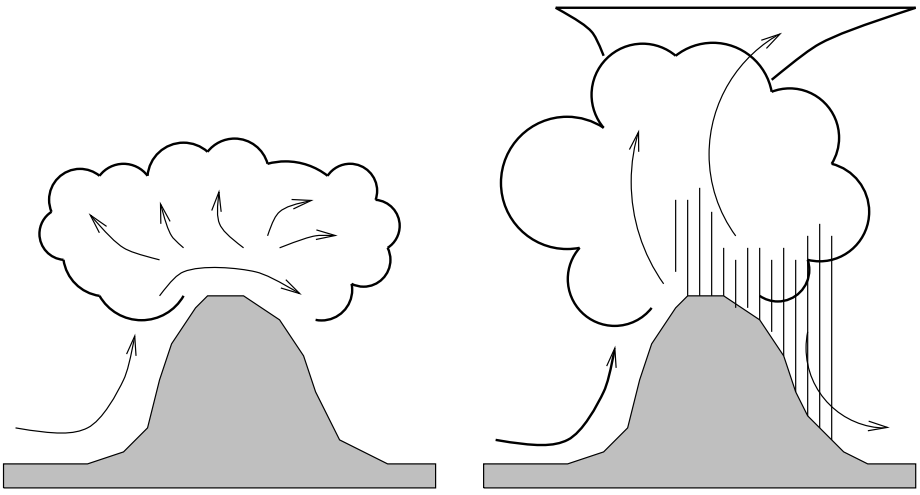

Figure 2.7: Cloud development in air forced upwards by high terrain: cumulus-like clouds (left) and cumulonimbus-like clouds with precipitation (right). (After [24].)

2.1.5 Summary

The formation of clouds is the result of a large number of linked processes on a wide variety of scales. Atmospheric instability causes vertical displacements of air masses. The air is chilled in ascent and eventually becomes saturated. Cloud drops start forming on condensation nuclei and may, given the right conditions, grow to cloud droplet size. The condensation process releases latent heat that in turn influences the air movement.

By using numerical methods it is possible to simulate the development of clouds, but the current meteorological cloud models are not suitable for computer graphics due to the substantial computational requirements and lack of visual detail.

2.2 Ontogenetic cloud modeling

It was argued in the previous section that the current meteorological models of convection are difficult to adopt for computer graphics applications. An alternative option is to try to create models that gives *a visual impression of clouds*, even if the modeling algoritm in itself has little or no correspondence to the actual physical processes that create clouds.

We shall refer to *ontogenetic modeling*, a term coined by Musgrave, as *modeling based on visible morphological characters*. In [8], he writes:

> The underlying idea of ontogenetic modeling is that, in the field of image synthesis, it is a legitimate engineering strategy to construct models based on subjective morphological (or other) *semblance*. This as opposed to, for instance, pursuing precise (e.g., mathematical) veracity, as is a goal in scientific models, or constructing them such that they and their behaviour issue from first scientific principles, as in the reductionist tradition.

Since our goal is to engineer visually realistic cloud models rather than to visualize the results of "real" cloud formation processes, we may adopt an ontogenetic modeling approach.

2.2.1 Previous work

A substantial number of researchers have been involved in work on ontogenetic cloud models. Voss [47] used pure fractal noise on $\mathbf{R}^3$ as a density function. Gardner [11], [12] used an approximation of fractal noise to modify the translucence of quadric surfaces. Max [27] used quadratic polynomials of the form

$$d - \frac{(x-x_0)^2}{a^2} - \frac{(y-y_0)^2}{b^2} - \frac{(z-z_0)^2}{c^2} \tag{2.40}$$

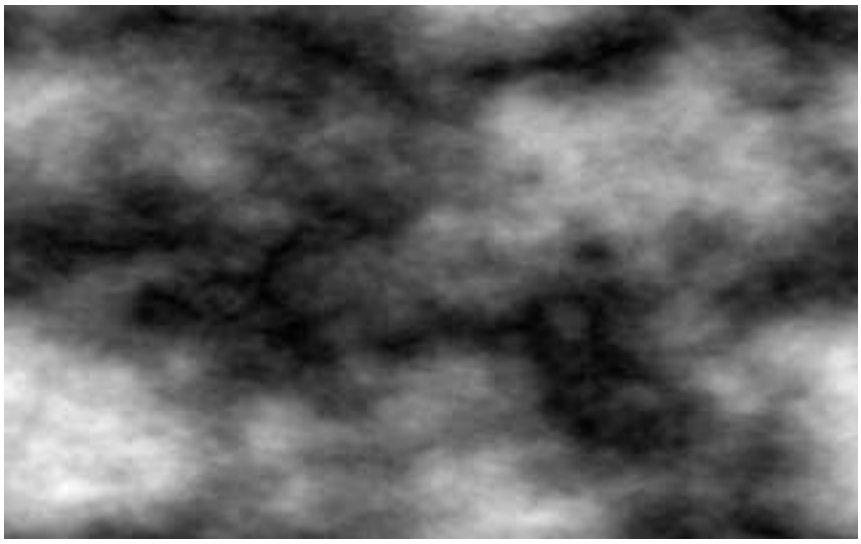

Figure 2.8: Fractal noise generated through spectral syntesis. The fractal dimension is 1.2.

to create unions of ellipsoid-shaped volumes on a 3D grid and then added fractal noise to these volumes. Ebert [8] used implicit spherical primitives and a density blending function to define cloud macrostructure and fractal noise within the primitives for cloud detail.

2.2.2 Fractal noise

Because of its self-similarity properties, most (if not all) ontogenetic cloud models are based on fractal noise or variations of fractal noise. Musgrave defines a fractal as *a geometrically complex object, the complexity of which arises through the repetition of form over some range of scale.* [8] A mathematical definition of fractal noise can be found in [25]. There are many techniques available for generating fractal noise. An excellent overview of many of these techniques is given in [35]. Figure 2.8 shows fractal noise generated by spectral syntesis.

Typically, unaltered or altered fractal noise is used as a cloud density function on a three-dimensional grid (for volume rendering) or as the basis for a texture (for controlling surface translucency) or some combination of these two approaches. When the value of the noise function drops below a certain threshold, the cloud density is assumed to be 0. The threshold is often increased at the imagined cloud boundaries to phase out the cloud matter smoothly and avoid sharp cloud edges.

In his 1985 paper, Gardner used the following construction for generating

a fractal-like noise.

$$t(x,y,z) = k\sum_{i=1}^{n}(c_i \sin(fx_i x + px_i) + t_0)\sum_{i=1}^{n}(c_i \sin(fy_i y + py_i) + t_0) \quad (2.41)$$

where

$$fx_{i+1} = 2fx_i \quad (2.42)$$

$$fy_{i+1} = 2fy_i \quad (2.43)$$

$$c_{i+1} = 0.707c_i \quad (2.44)$$

$$px_i = (\frac{\pi}{2}\sin(fy_{i-1}y)) + \pi\sin(\frac{fx_i z}{2}) \quad (2.45)$$

$$py_i = (\frac{\pi}{2}\sin(fx_{i-1}x)) + \pi\sin(\frac{fx_i z}{2}) \quad (2.46)$$

He then used the result to modify the translucency of ellipsoid and plane surfaces.

The main drawback of this technique is that the values for n, k, t_0, c_0, fx_0, fy_0, px_0 and py_0 have to be selected very carefully in order to avoid visual artefacts. The selection of initial values is difficult because the parameters do not correspond directly to visual features. Unfortunately, Gardner did not provide the values used to create his images. Nevertheless, the images are among the most impressive and realistic images of clouds ever produced and were the state of the art for almost ten years.

In a famous 1985 paper, Perlin described an extremely useful type of noise, usually referred to as *Noise()* [37]. It is defined as a function from $\mathbf{R}^3$ to the interval $[0,1]$ and has the following properties:

- Statistical invariance under rotation.
- A narrow bandpass limit in frequency.
- Statistical invariance under translation.

Noise() can be used to create a turbulence-like approximation of fractal noise by using the following algorithm:

```
float
Turbulence (vector p)
{
   t = 0;
   scale = 1;
   while (scale > pixelsize)
   {
      t += abs (Noise (p / scale) * scale);
      scale /= 2;
   }
   return t;
}
```

Several different implementations of *Noise()* can be found in [8].

2.2.3 Extensions to previous techniques

The two main drawbacks of Gardner's method are the noise function and the fact that the resulting clouds are hollow. Using ellipsoids as a basis for the cloud macrostructure is beneficial since they can be rendered quickly, both by ray tracing and scan-line methods. As we shall see, it is also relatively simple to describe the main shape of a cloud by using unions of ellipsoids.

Ebert's method partly addresses the problems in Gardner's modeling technique, but it is inefficient since it requires traversing the entire length of each eye ray, evaluating the density blending function at each point.

A new combination of Ebert's and Gardner's techniques is presented that is both efficient and produces convective clouds with a three-dimensional density distribution (that is, the clouds are non-hollow).

Density function

Perlin's *Turbulence()* function as described above is a suitable fractal noise generator for clouds. The *Noise()* implementation is from the CD-ROM accompanying [8]. The macrostructure of a cloud is defined as the union of a number of ellipsoids. To allow for fractured cloud boundaries, the amount of cloud matter $\rho_i(\vec{P_i})$ at a point $\vec{P_i}$ inside an ellipsoid i with center point at $\vec{c_i}$ is defined as

$$\rho_i(\vec{P_i}) = \begin{cases} Turbulence(\vec{P_i} + \vec{c_i}) & \text{if } Turbulence(\vec{P_i} + \vec{c_i}) > t \\ 0 & \text{otherwise} \end{cases} \tag{2.47}$$

where $\vec{P_i}$ is defined so that $\|\vec{P_i}\| = 1$ at the ellipsoid boundaries and $\|\vec{P_i}\| = 0$ at $\vec{c_i}$. t is a threshold value defined as

$$t = t_B \|\vec{P_i}\|^n \tag{2.48}$$

where n and t_B are user-specified values controlling the density fall-off and threshold at the ellipsoid boundary, respectively.

The total cloud density $\rho(\vec{P})$ is simply the sum over the set $E_{\vec{P}}$ of ellipsoids that contains $\vec{P}$:

$$\rho(\vec{P}) = \sum_{i \in E_{\vec{P}}} \rho_i(\vec{P_i}) \tag{2.49}$$

Note that ρ is independent of the rendering method: it can be used in ray tracing, scan-line techniques or volume rendering.

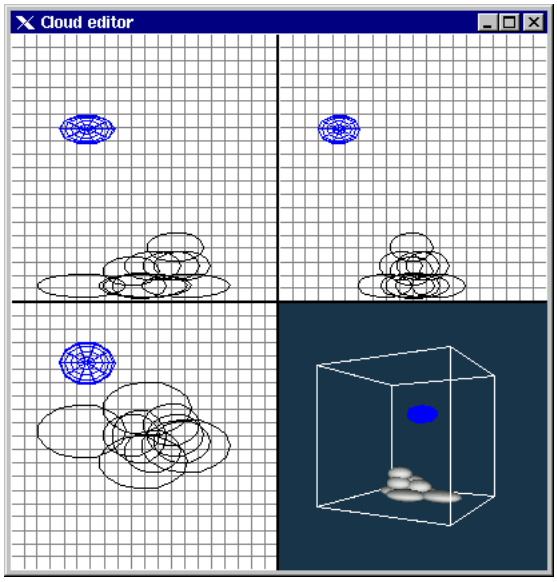

Figure 2.9: Cloud macrostructure editor.

Cloud macrostructure editor

An editor was developed to let users define the cloud macrostructure. Figure 2.9 shows a screenshot of the editor. The main window contains four subwindows. Three show the cloud from the side, front and top directions, respectively. The fourth subwindow shows a rotating, shaded perspective view of the cloud. A cumulus humilis has been drawn using ten ellipsoid primitives. Figure 2.10 shows a rendering of the cloud.

The clouds created with the editor can be used as templates for a cloud randomization program. Consider a cloud template containing N ellipsoids with center points at $[x_i\ y_i\ z_i]$ and radii a_i, b_i and c_i, $i = 1 \ldots N$. A random variation of the template is defined by

$$x_i' = x_i + (a_i * 0.5 * random(-1, 1)) \tag{2.50}$$

$$y_i' = y_i + (b_i * 0.5 * random(-1, 1)) \tag{2.51}$$

$$z_i' = z_i + (c_i * 0.5 * random(-1, 1)) \tag{2.52}$$

$$a_i' = a_i + (a_i * 0.1 * random(-1, 1)) \tag{2.53}$$

$$b_i' = b_i + (b_i * 0.1 * random(-1, 1)) \tag{2.54}$$

$$c_i' = c_i + (c_i * 0.1 * random(-1, 1)) \tag{2.55}$$

Figure 2.11 shows a number of such random variations of the cloud macrostructure in figure 2.9.

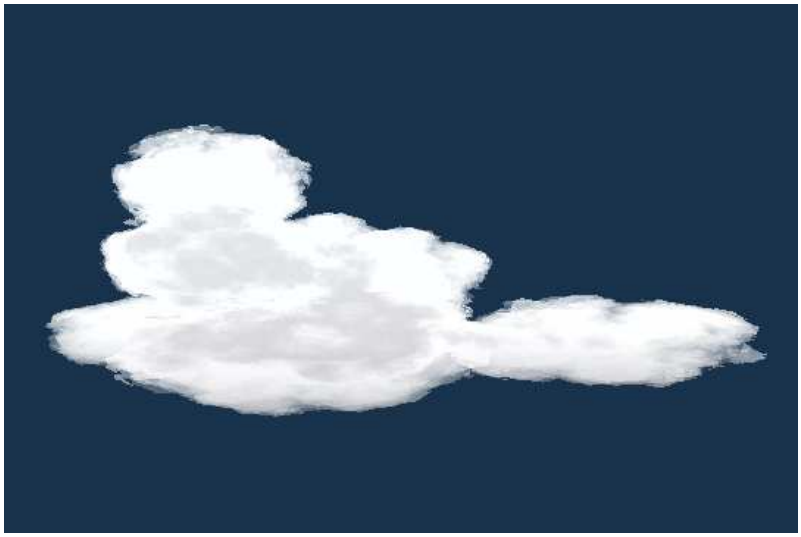

Figure 2.10: Rendering of the cloud macrostructure shown in figure 2.9.

Figure 2.11: Rendering of random variations of cloud macrostructure shown in figure 2.9.

2.2.4 Summary

The use of ontogenetic modeling techniques is important in computer grapics. Often, the physical processes behind a phenomenon are too difficult or costly to simulate, which makes it necessary to create a model with a *visual semblance* to the phenomenon.

A substantial number of ontogenetic models for convective clouds have been presented by previous resarchers, each with particular benefits and drawbacks. I have attempted to bring together the advantages of two of these models. The result is a simple and effective cloud modeling technique that can be used as input for several different rendering methods. The main benefits of the new method are:

- The user is given explicit control of the appearance of the cloud macrostructure.
- A user-created cloud specification can be used as a template for random cloud variations. The randomization process is automatic.
- The user is given control over cloud microstructure, with parameters that are intuitive and easy to use.
- The noise function does not have to be evaluated outside the boundaries of the cloud macrostructure.
- The density function can be used for volumetric rendering as well as for ray tracing and scan-line methods.

Chapter 3

Cloud rendering

In the previous chapter, different techniques for obtaining cloud density functions $\rho(\vec{x})$, $\vec{x} \in \mathbf{R}^3$ were discussed. This chapter describes how such functions can be rendered on a computer display.

We shall refer to *cloud rendering* as the task of transforming the density function into color values suitable for presentation on a computer monitor or other similar devices. Typically, this means calculating an RGB color triplet for each pixel in a color buffer.

As with cloud modeling, we can approach cloud rendering in two ways. We can try to simulate the physical interaction between light and cloud drops or we can use an ontogenetic approach that gives a visual impression of cloudy air with little or no connection to the actual physical processes. An overview of physically-based techniques are given in Section 3.1. A new ontogenetic approach for cloud rendering is presented in Section 3.2.

3.1 Physically-based rendering

In this section, we review some of the optical properties of the atmosphere and cloudy air. We also give a short overview of previous research.

3.1.1 Interaction between light and atmospheric particles

Light traveling through a dispersive medium such as atmospheric particles or cloudy air is scattered by the medium. The amount of scattering in different directions depends on the size of the particles with respect to the wavelength λ of the light. Figure 3.1 shows a single scattering event. A ray of light is traveling in the positive z direction and is scattered at the origin. The intensity $I(\lambda)$ of light that remains after such a scattering event can be written

$$I(\lambda) = I_0(\lambda)\beta_{sc}(\lambda, \theta, \phi) \tag{3.1}$$

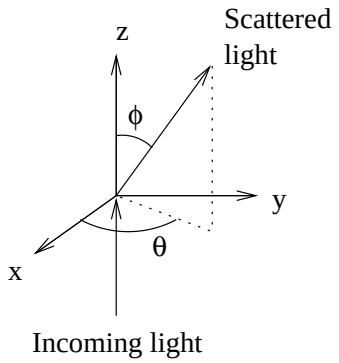

Figure 3.1: A single scattering event. (After [22].)

where $I_0(\lambda)$ is the intensity of the incoming light, θ and ϕ are the scattering angles and $\beta_{sc}(\lambda, \theta, \phi)$ is called the *scattering coefficient* or the *scattering function*. For small spherical particles, β_{sc} is symmetric about the direction of the incoming light and we can write

$$I(\lambda) = I_0(\lambda)\beta_{sc}(\lambda, \phi) \tag{3.2}$$

Equation (3.2) applies to air molecules, water vapor, haze particles, cloud drops and small raindrops.

The light is also attenuated by the dispersive medium. The rate of extinction depends on the sizes and the density of the particles. We define the *extinction coefficient*, $\beta_{ex}(\lambda)$ as the total amount of light lost per unit distance per particle.

If we assume that scattering occurs only once, it is possible to derive a simple expression for the amount of light that remains after attenuation and scattering. Consider a ray of light with intensity $I_0(\lambda)$ that travels through a dispersive medium and reaches the eye point as in figure 3.2. The light enters the medium at P_b and exits at P_a. Sunlight with intensity $I_{sun}(\lambda)$ enters the medium at P_c and is scattered in the direction of the eye point at P. The amount of light that reaches the eye point can then be written

$$I(\lambda) = I_{ab}(\lambda) + I_{sc}(\lambda) \tag{3.3}$$

where $I_{ab}(\lambda)$ is the amount of $I_0(\lambda)$ that remains after absorption. $I_{sc}(\lambda)$ is the total amount of scattered light that remains after absorption. $I_{ab}(\lambda)$ is given by

$$I_{ab}(\lambda) = I_0(\lambda)\exp(-\tau(P_a, P_b, \lambda)) \tag{3.4}$$

where τ is the *optical depth*

$$\tau(a, b, \lambda) = \int_a^b \beta_{ex}(\lambda)\rho(s)ds \tag{3.5}$$

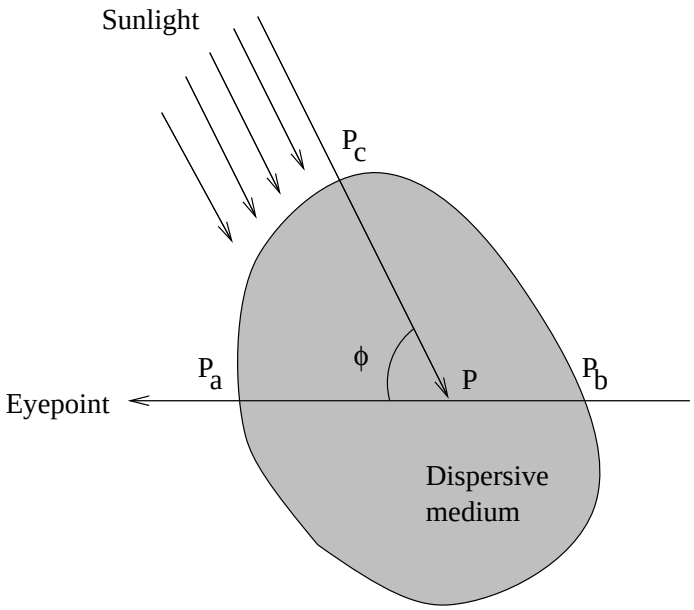

Figure 3.2: Geometry of light traveling through a dispersive medium. (After [33].)

$\rho(s)$ is the density of the medium at s. I_{sc} is given by

$$I_{sc}(\lambda) = I_{sun}(\lambda) \int_{P_a}^{P_b} \beta_{sc}(\lambda, \phi) \exp(-\tau(P_c, l, \lambda) - \tau(l, P_a, \lambda))dl \tag{3.6}$$

When the resulting $I(\lambda)$ has been calculated, it must typically be converted to an RGB triplet [38], [29], [3]. This can be done by first converting the intensity spectrum to a *CIE XYZ coordinate triplet* by integrating it over the visible wavelenghts λ_v:

$$X = k \int_{\lambda_v} I(\lambda)\overline{x}(\lambda)d\lambda \tag{3.7}$$

$$Y = k \int_{\lambda_v} I(\lambda)\overline{y}(\lambda)d\lambda \tag{3.8}$$

$$Z = k \int_{\lambda_v} I(\lambda)\overline{z}(\lambda)d\lambda \tag{3.9}$$

$\overline{x}$, $\overline{y}$ and $\overline{z}$ are the *CIE color matching functions*, tabulated in [38]. k is a normalization constant. The corresponding RGB triplet can then be obtained by multiplying the XYZ coordinate with a transformation matrix.

Scattering functions

Given a scattering function and an extinction coefficient, (3.3) can be integrated numerically. For atmospheric particles and molecules with radii up

to about 0.05λ, *Rayleigh scattering* predominates. The extinction coefficient per molecule and unit length is approximately

$$\beta_{ex}(\lambda) \approx \frac{8\pi^3}{3}\frac{(n^2-1)^2}{\lambda^4 N} \tag{3.10}$$

where N is the molecular number density of the atmosphere and n is the refractive index of air.

The scattering function per molecule is approximately

$$\beta_{sc}(\lambda,\phi) \approx \frac{\pi^2}{2}\frac{(n^2-1)^2}{\lambda^4 N}(1+\cos^2(\phi)) \tag{3.11}$$

Although N and n vary with temperature and pressure, β_{ex} and β_{sc} are often assumed to depend only on λ and on λ and ϕ respectively [36]. For computer graphics purposes, this does not seem to cause visible artefacts—see, for example, the results in [33].

The $1/\lambda^4$ dependence of the scattering and extinction coefficients explains the blue color of the sky during the day and the red color at sunrise and sunset. The blue color is scattered light in the blue wavelengths and the red color is light that remains after the out-scattering of the blue wavelengths.

For larger particles, such as haze and cloud droplets, Mie scattering predominates. The extinction and scattering coefficients does not show the same selectivity with respect to wavelength as in Rayleigh scattering and are often assumed to be independent of wavelength. Because of its complexity, the details of Mie scattering theory is beyond the scope of this work.

According to [6], the Mie extinction coefficient per kilometer for a typical cloud is

$$\beta_{ex} = 16.33 \tag{3.12}$$

An approximation of the scattering function, cited in [33] is

$$\beta_{sc}(\phi) \propto \frac{3(1-g^2)}{2(2+g^2)}\frac{(1+\cos^2\phi)}{(1+g^2-2g\cos\phi)^{3/2}} \tag{3.13}$$

$$g = \frac{5}{9}u - \left(\frac{4}{3} - \frac{25}{81}u^2\right)x^{-1/3} + x^{1/3} \tag{3.14}$$

$$x = \frac{5}{9}u + \frac{125}{729}u^3 + \left(\frac{64}{27} - \frac{325}{243}u^2 + \frac{1250}{2187}u^4\right)^{1/2} \tag{3.15}$$

where u varies from 0.7 to 0.85.

Tabulated values of the extinction and scattering coefficients can be found in a number of publications (see, for example, the references in [22], [33] and [34]).

3.1.2 Optical models for clouds used in computer graphics

Blinn was among the first researchers to use models of scattering for visualization of density fields [1]. The case in study was the rings of Saturn, which he assumed to consist of dust layers with a low albedo. He then integrated a set of single-scattering formuale across the layers. His paper also contains an overview of scattering coefficients.

In an often-cited paper [20], Kajiya and von Herzen improved Blinn's formulae and presented a single scattering visualization method for ray tracing. They then used the method to render a cloud created from a meteorological model. They also presented a method where multiple scattering was taken into account, but unfortunately it was never implemented.

Klassen [22] used a single-scattering model to simulate the color of the sky during day, sunrise and sunset. A similar model was used by Nishita *et al.* to create images of the Earth from space [33].

Since clouds have a very high albedo, the effects of multiple scattering cannot be ignored [27], [28], [34]. Several researchers have tried to create realistic images of clouds by modeling multiple scattering. In these methods, the density field is typically defined at discrete points on a grid. A set of radiation transport equations between different points is then solved by numerical integration.

Max [27], [28] used direction bins and the discrete ordinates method to solve the equations. Whitehouse [50] used a parallel supercomputer to solve the equations. Nishita *et al.* [34] used a combination of numerical integration and Monte Carlo ray tracing.

Because of the heavy computational demands of these methods, the number of points on the density grid must be kept small, typically in the range 20^3 to 100^3. Such a low number of grid points are insufficient for displaying the small and complex visual details of clouds. As a consequence, the resulting images show very realistic lighting conditions but the cloud models themselves are fuzzy and too homogenous. Musgrave, Lee and Max have presented promising images from currently unpublished research that address this issue [8]. Unfortunately, the details of the method and the times required for rendering are not available.

3.1.3 Summary

The interaction between light and atmospheric particles is complex and simulating it requires time-consuming calculations. The development of heavy-duty workstations and parallel supercomputers has made it possible to perform these calculations on medium-sized density grids in a matter of minutes. The lighting in the resulting images is very realistic but the grid sizes are not large enough faithfully to reproduce the complex visual detail of clouds.

3.2 Ontogenetic rendering

It was argued in the previous section that the rendering of truly realistic images of clouds requires very fast computers, preferably parallel supercomputers. The question is, then, can we render realistic images of clouds on slower workstations? By examining the visual appearance of clouds under different lighting conditions, we can design efficient ontogenetic lighting models that give good results for these conditions. One such method is presented in this section.

3.2.1 Previous work

I have not been able to find any research paper that deals exclusively with ontogenetic rendering of clouds. A few resarchers have used rendering methods that are not based on physics. Gardner [11], [12] used a variation of fractal noise inside the limb curves of ellipsoids projected onto the image plane. Ebert [8] used the single-scattering method of [20] and added an ambient term to account for multiple scattering.

3.2.2 An ontogenetic rendering method for clouds

The visual appearance of clouds is extremely varied and depends on the intensity and color of sunlight, light reflected from the ground and the thickness and topology of the clouds themselves. I have examined different photographs of clouds and tried to recreate their appearance by using a flexible ontogenetic model. The model assumes that the position of the sun is stationary for each set of model parameters and that clouds do not cast shadows onto other clouds.

Model description

The model divides the color of a cloud into two categories: the color of sunlit cloudy air, $I_c = (R_c, G_c, B_c)$, and the color of cloudy air in shadow, $I_s = (R_s, G_s, B_s)$. Both colors are provided by the user. I_c and I_s are then combined with the color of the background, $I_{bg} = (R_{bg}, G_{bg}, B_{bg})$ to give the color $I = (R, G, B)$ of an eye ray according to

$$I = k_{bg}(o)I_{bg} + k_c(o)I_c + k_s(o)I_s \tag{3.16}$$

where $k_{bg}(o)$, $k_c(o)$ and $k_s(o)$ are weighting functions of the *opacity*

$$o = 1 - \exp(-\kappa \int_a^b \rho(s)ds) \tag{3.17}$$

where κ is a user-provided constant and a and b are the points where the ray enters and exits the cloud, respectively. In the current implementation,

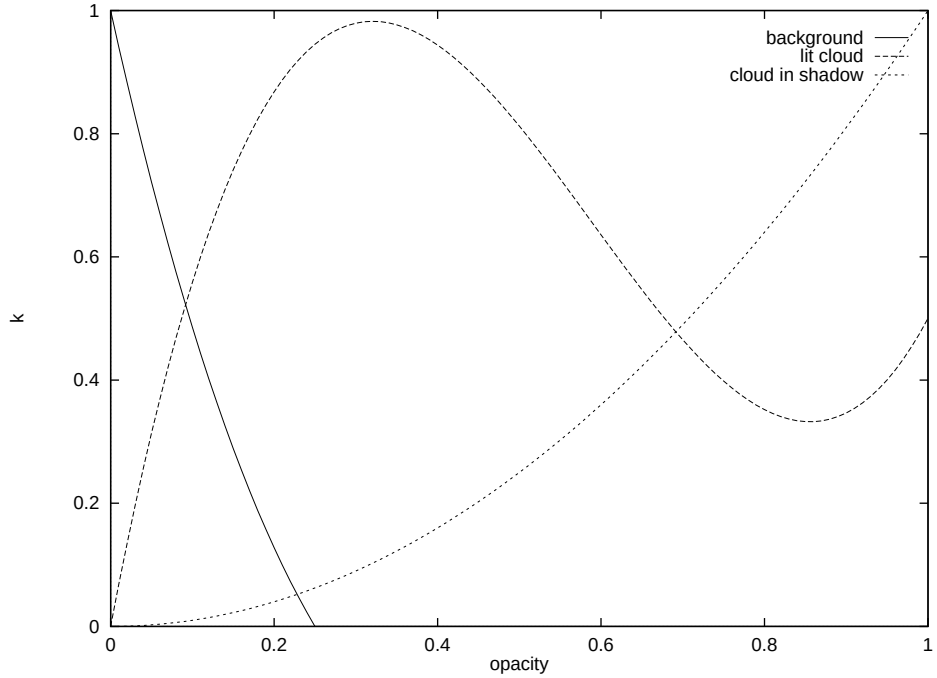

Figure 3.3: Weighting functions of opacity for background color, sunlit cloudy air and cloudy air in shadow.)

the weighting functions are Hermite splines of the form

$$k(t) = (2t^3 - 3t^2 + 1)p_0 + (-2t^3 + 3t^2)p_1 + (t^3 - 2t^2 + t)r_0 + (t^3 - t^2)r_1 \quad (3.18)$$

where p_0 and p_1 denotes the value of $k(t)$ at $t = 0$ and $t = 1$, respectively and r_0 and r_1 denotes the derivative of $k(t)$ at $t = 0$ and $t = 1$, respectively. Figure 3.3 shows the weighting functions used for my images.

The lower parts of a cloud are often in shadow, so we modify I by blending in a user-provided shadow color I_l

$$I = (1 - a)I + (a_k \cdot a)I_l \quad (3.19)$$

$$a = alt^{a_e} \quad (3.20)$$

where a_k and a_e are user-provided constants and alt is a normalized altitude such that $alt = 0$ at the bottom of the cloud and $alt = 1$ at the top of the cloud.

Ray tracing software

In order to try different approaches to cloud rendering, a generic ray tracing library was developed. The library is suitable for all applications that

require ray tracing. It consists of a set of core functions for managing rays, transformation matrices, hit lists, primitives, lights, cameras, texture maps and images. Primitive instances (such as boxes, spheres, density fields and height fields) and shaders are written as plug-in libraries that are linked together with the core library to form an application program. The ray tracing library is written in portable ANSI C. A C++ version is planned.

3.2.3 Summary

In this section, a new ontogenetic approach to cloud rendering was presented. Compared to the physically-based methods described in the previous section, it is flexible, simple and very efficient. Although it is not claimed that the results are photorealistic, it is felt that they are good enough for visualization, illustration and educational purposes where short rendering times are important.

Chapter 4

Results

This chapter contains results of combining the cloud model from Section 2.2.3 and the rendering method from Section 3.2. All images have the dimensions 540×360 and were rendered on a Pentium 90 system running RedHat Linux 4.2.

The ocean is a bump-mapped plane. The bump map was created by a program that implements the spectral synthesis method of Mastin *et al.* [26].

Figure 4.1 shows a field of cumulus humilis clouds. The time required for rendering was 400 seconds.

Figure 4.2 shows a field of cumulus mediocris clouds. The time required for rendering was 2260 seconds.

Figure 4.3 shows a field of cumulus congestus clouds. The time required for rendering was 1432 seconds.

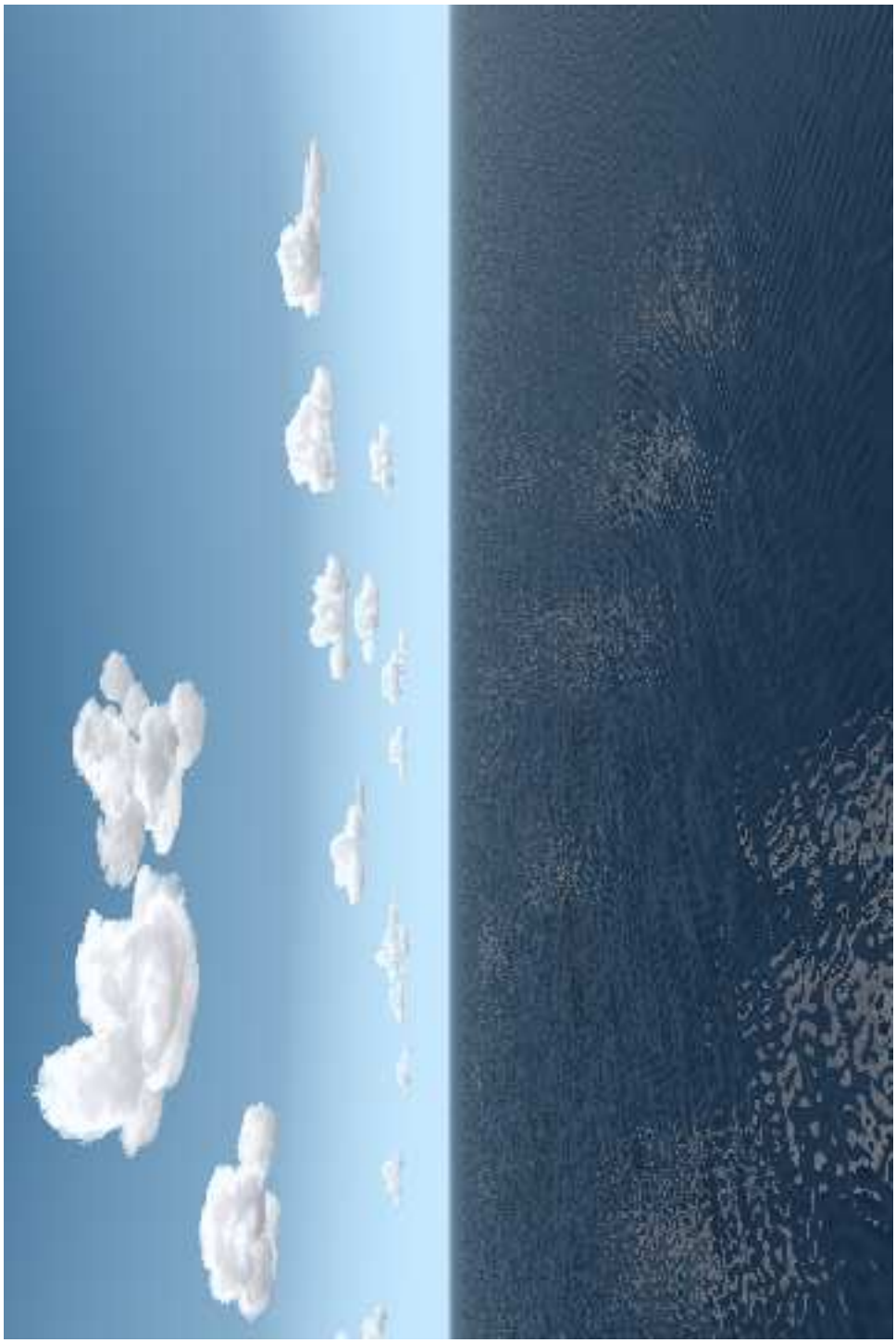

Figure 4.1: Cumulus humilis.

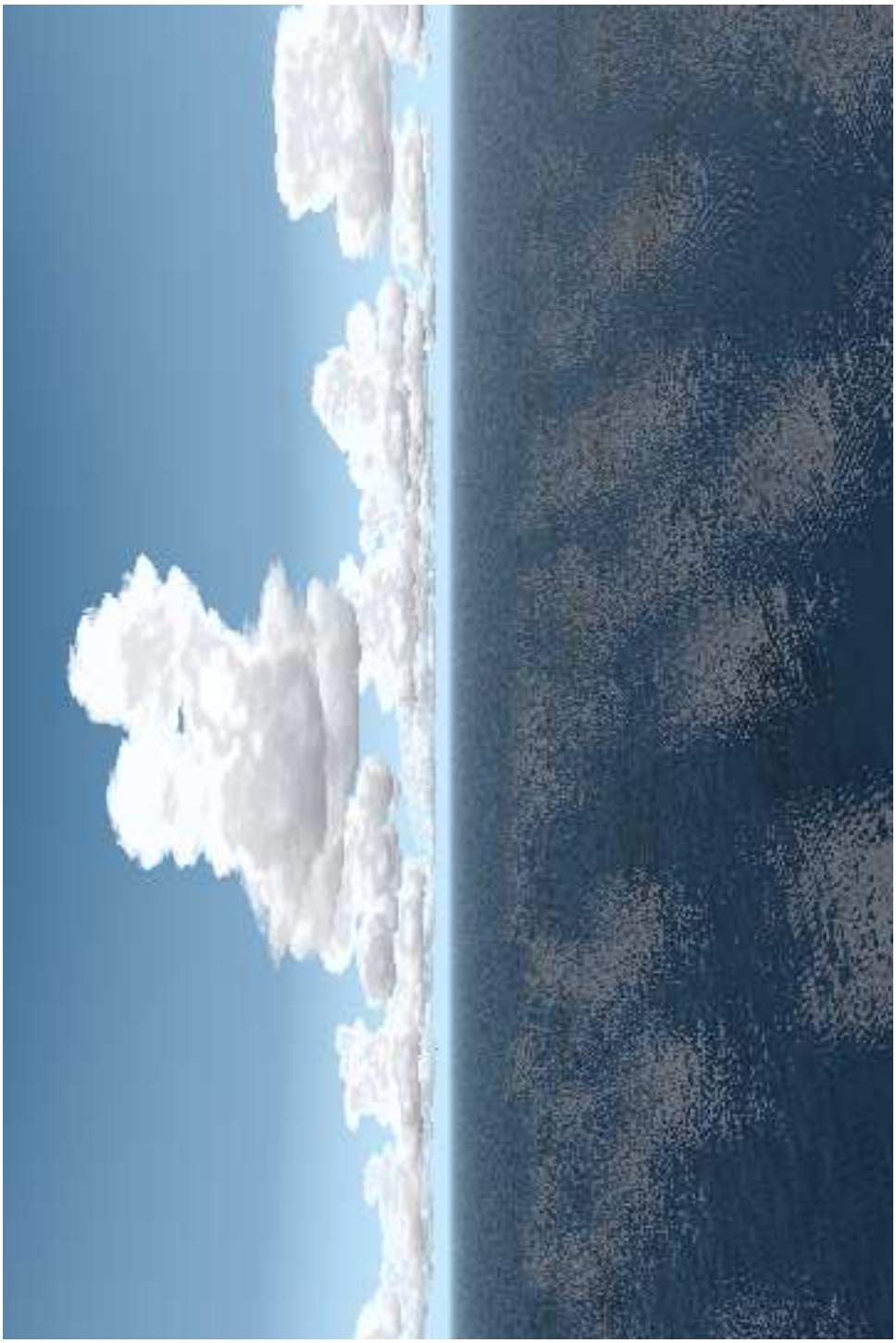

Figure 4.2: Cumulus mediocris.

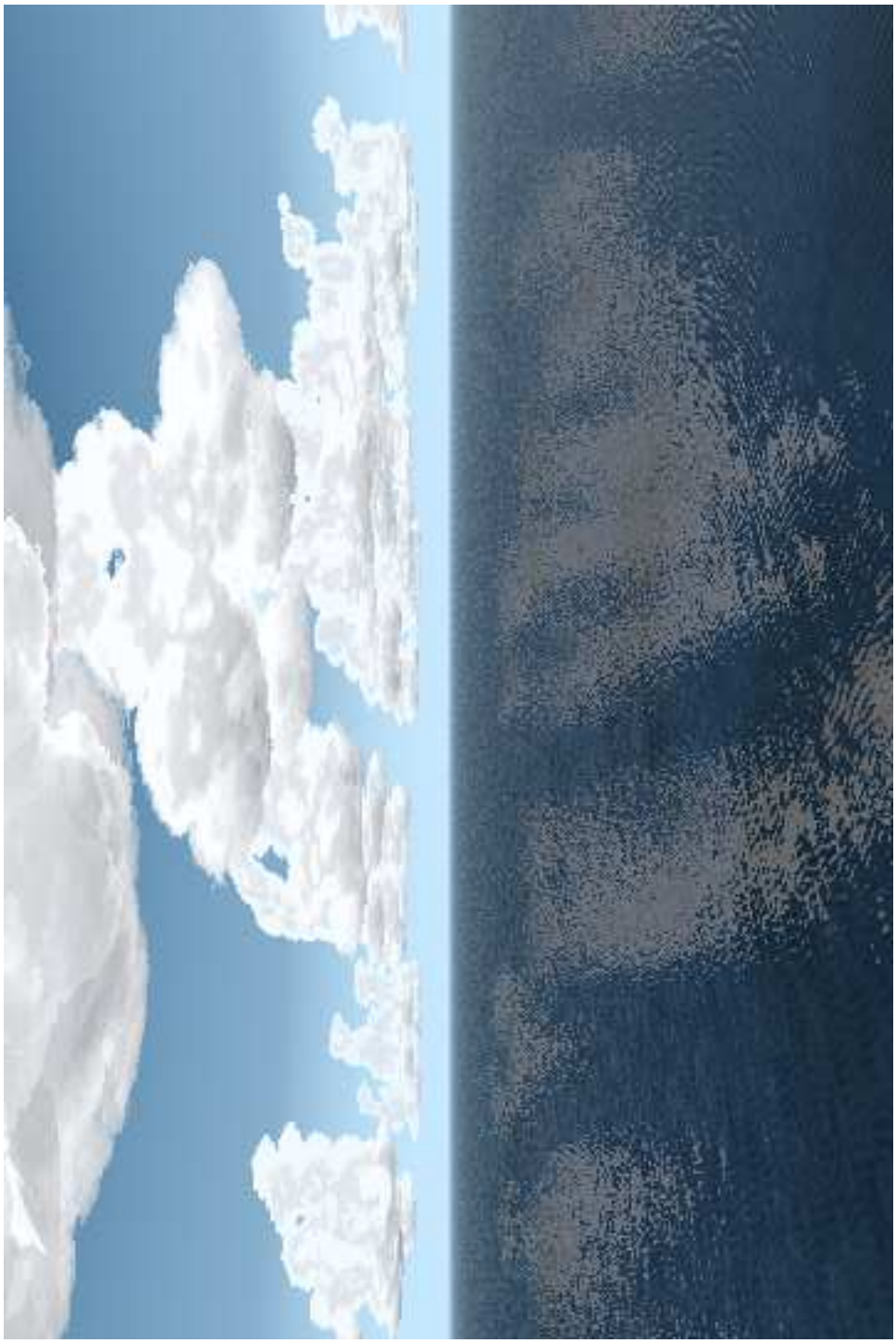

Figure 4.3: Cumulus congestus.

Chapter 5

Discussion and future work

Creating realistic images of clouds has been desirable in computer graphics for many years. The main reasons why this has been difficult to achieve are:

- The physical processes that create clouds are extremely complex and occur over a wide range of scales in three dimensions. The requirement for a high level of detail in computer graphics makes them difficult and very time-consuming to simulate.
- The high albedo of clouds makes it necessary to account for multiple scattering when simulating the interaction between light and cloudy air. Solving the radiation transfer equations for multiple scattering in three dimensions is very time-consuming.
- Clouds are familiar to everyone. It is very easy to spot visual artefacts in rendered cloud images.

One of the main points of this thesis is that while the current physically-based methods can produce very good results, they are far too time-consuming and complex to be usable on desktop workstations. Even when the systems of equations are solved by using massively parallel supercomputers, the level of detail is not high enough to remove visible artefacts if the rendering times are to be kept to an acceptable level. Therefore, I have chosen to examine different ontogenetic modeling and rendering techniques.

The new ontogenetic modeling approach presented in Section 2.2.3 is a straightforward combination of two previous techniques. It is simple and very fast compared to simulating the physics of cloud creation. It also gives the user explicit control over the appearance of the cloud macrostructure, which makes it easy to model different types of clouds. Random variations of clouds created by the user can be generated automatically. These can easily be combined to form cloud fields and different cloud fields can be combined to produce varied and realistic scenes.

It has also been noted in this thesis that previous research on the subject of ontogenetic rendering methods for clouds is very limited. The general approach to creating images of clouds seems to be to use simple modeling techniques and render the models by simulating physical processes. The simulation process creates realistic lighting but is very time-consuming and requires fast workstations or supercomputers. Therefore, I have chosen to use an ontogenetic approach for rendering. The result is a simple and efficient technique that gives the user a high level of control over the appearance of the clouds.

The ontogenetic modeling and rendering methods presented in this thesis are useful but require refinement. Apart from improving the models themselves, it would be interesting to try to bring together elements from physically-based techniques and ontogenetic techniques.

Creating dynamic clouds and animations of cloud development is one possible area for future research. The cloud macrostructure ellipsoids can be animated in a particle system environment where forces and wind flows are obtained from physically-based simulations. The cloud microstructure could be animated by using the technique presented in [43]. It would also be interesting to try to let physically-based simulations influence the cloud microstructure.

The ontogenetic renderer should be extended to allow clouds to cast shadows onto other clouds. One way to accomplish this is to trace a shadow ray towards the light source for points on the cloud macrostructure boundary (i.e. the point where $\varrho \neq 0$ for the first time when moving along an eye ray). If an obstacle is hit, the cloud color could be augmented or set to a user-defined color. Another interesting issue is to attempt to introduce physically-based features into the ontogenetic rendering model. By examining the results of physically-based models, it may be possible to make assumptions about the color contribution of a point inside the cloud based on its location with respect to the cloud boundary.

Chapter 6

Conclusion

The modeling and rendering of clouds are difficult tasks that have vexed computer graphics researchers for many years. The main reason for this is the visual complexity of clouds and the complexity of the physical processes that create them. Physically-based simulations of cloud development and cloud appearance tend to be mathematically advanced and very time-consuming. Therefore, many researchers have developed ontogenetic modeling techniques that try to capture the visual appearance of clouds without the need for simulation. Typically, the resulting clouds are then rendered using physically-based models.

This thesis introduces a new ontogenetic cloud modeling and rendering technique that is simple and very efficient. The user is given explicit control over the visual appearance of the cloud through model parameters that are easy and intuitive to use. Cloud models created by the user can be used as templates for automatic generation of cloud-covered skies. The cloud models can be rendered with both ray-tracing and scan-line methods.

Bibliography

[1] Blinn, J. F., 1982, "Light Reflection Functions for Simulation of Clouds and Dusty Surfaces." *Computer Graphics* 1982: 16(3), pp. 21–29.

[2] Bohren, C. F. (ed.), 1989, *Selected Papers on Scattering in the Atmosphere.* Society of Photo-Optical Instrumentation Engineers (SPIE) Milestone Series Volume MS 7. ISBN 0-8194-0283-4.

[3] Borges, C. F., 1991, "Trichromatic Approximation for Computer Graphics Illumination Models." *Computer Graphics* 1991: 25(4), pp. 101–104.

[4] Clark, T. L., 1979, "Numerical Simulations with a Three-Dimensional Cloud Model: Lateral Boundary Condition Experiments and Multicellular Severe Storm Simulations." *Journal of the Atmospheric Sciences* 1979:36, pp. 2191–2215.

[5] Dahlquist, G., and Björk, Å., 1974, *Numerical Methods.* Prentice-Hall, 573 pages. ISBN 0-13-627315-7.

[6] Deirmendjian, F., 1964, "Scattering and Polarization Properties of Water Clouds and Hazes in the Visible and Infrared." *Applied Optics* 1964: 3(2), pp. 187–196.

[7] Dobashi, Y., Kaneda, K., Yamashita, H., and Nishita, T., 1996, "Method for Calculation of Sky Light Luminance Aiming at an Interactive Architectural Design." *Computer Graphics Forum* 1996: 15(3), pp. 112–118.

[8] Ebert, D. S., Musgrave, F. K., Peachey, D., Perlin, K., and Worley, S., 1998, *Texturing and Modeling. A Procedural Approach.* Second edition. AP Professional, 415 pages. ISBN 0-12-228730-4.

[9] Fermi, E., 1936, *Thermodynamics.* Dover Publications, 160 pages. ISBN 0-486-60361-X.

[10] Foley, J. D., van Dam, A., Feiner, S. K., and Hughes, J. F., 1990, *Computer Graphics. Principles and Practice.* Second edition. Addison-Wesley Publishing Company, 1174 pages. ISBN 0-201-12110-7.

[11] Gardner, G. Y., 1984, "Simulation of Natural Scenes Using Textured Quadric Surfaces." *Computer Graphics* 1984: 18(3), pp. 11–20.

[12] Gardner, G. Y., 1985, "Visual Simulation of Clouds." *Computer Graphics* 1985: 19(3), pp. 297–303.

[13] Glassner, A. S. (ed.), 1989, *An Introduction to Ray Tracing.* Academic Press, 329 pages. ISBN 0-12-286160-4.

[14] Greenler, R., 1980, *Rainbows, Halos, and Glories.* Cambridge University Press, 195 pages. ISBN 0-521-38865-1.

[15] Haltiner, G. J., 1959, "On the Theory of Convective Currents." *Tellus* 1959:11, pp. 4–15.

[16] Hill, G. E., 1974, "Factors Controlling the Size and Spacing of Cumulus Clouds as Revealed by Numerical Experiments." *Journal of the Atmospheric Sciences* 1974:31, pp. 646–673.

[17] Hill, G. E., 1977, "Initiation Mechanisms and Development of Cumulus Convection." *Journal of the Atmospheric Sciences* 1977:34, pp. 1934–1941.

[18] Hulburt, E. O., 1953, "Explanation of the Brightness and Color of the Sky, Particularly the Twilight Sky." *Journal of the Optical Society of America* 1953: 43(2), pp. 113–118. Reprinted in [2].

[19] *International Cloud Atlas.* World Meteorological Organization, 1969.

[20] Kajiya, J. T., and Von Herzen, B. P., 1984, "Ray Tracing Volume Densities." *Computer Graphics* 1984: 18(3), pp. 165–174.

[21] Kilgard, M. J., 1996, *OpenGL Programming for the X Window System.* Addison-Wesley Developers' Press, 542 pages. ISBN 0-201-48359-9.

[22] Klassen, R. V., 1987, "Modeling the Effect of the Atmosphere on Light." *ACM Transactions on Graphics* 1987: 6(3), pp. 215–237.

[23] Lejenäs, H., 1995, *Meteorologi. En introduktion till de meteorologiska skeendena i lufthavet.* Compendium. Meteorologiska institutionen, University of Stockholm, 220 pages.

[24] Liljequist, G. H., 1979, *Moln - deras uppkomst och formationer.* Generalstabens litografiska anstalt, 120 pages. ISBN 91-7058-084-7.

[25] Mandelbrot, B., 1983, *The Fractal Geometry of Nature.* W. H. Freeman and Company, 468 pages. ISBN 0-7167-1186-9.

[26] Mastin, G. A., Watterberg, P. A., and Mareda, J. F., 1987, "Fourier Synthesis of Ocean Scenes." *IEEE Computer Graphics and Applications* March 1987, pp. 16–23.

[27] Max, N., 1995, "Efficient Light Propagation for Multiple Anisotropic Volume Scattering." In Sakas, G., Shirley, P., and Müller, S. (eds.), 1995, *Photorealistic Rendering Techniques.* Springer-Verlag. pp. 87–104.

[28] Max, N., "Optical Models for Direct Volume Rendering." *IEEE Transactions on Visualization and Computer Graphics.* 1995: 1(2), pp. 97–108.

[29] Meyer, G. W., and Greenberg, D. P., 1987, "Perceptual Color Spaces for Computer Graphics." In Durrett, H. J. (ed.), 1987, *Color and the Computer.* Academic Press. ISBN 0-12-225210-1.

[30] Mordy, W., 1959, "Computations of the Growth by Condensation of a Population of Cloud Droplets." *Tellus* 1959:11, pp. 16–44.

[31] Morton, K. W., and Mayers, D. F., 1994, *Numerical Solution of Partial Differential Equations.* Cambridge University Press, 227 pages. ISBN 0-521-42922-6.

[32] Nishita, T., Miyawaki, Y., and Nakamae, E., "A Shading Model for Atmospheric Scattering Considering Luminous Intensity Distribution of Light Sources." *Computer Graphics* 1987: 21(4), pp. 303–310.

[33] Nishita, T., Sirai, T., Tadamura, K., and Nakamae, E., "Display of The Earth Taking into Account Atmospheric Scattering." *Computer Graphics* 1993, Annual SIGGRAPH conference proceedings, pp. 175–182.

[34] Nishita, T., Dobashi, Y., and Nakamae, E., "Display of Clouds Taking into Account Multiple Anisotropic Scattering and Sky Light." *Computer Graphics* 1996, Annual SIGGRAPH conference proceedings, pp. 379–386.

[35] Peitgen, H-O., and Saupe, D. (eds.), 1988, *The Science of Fractal Images.* Springer-Verlag, 312 pages. ISBN 3-540-96608-0.

[36] Penndorf, R., 1957, "Tables of the Refractive Index for Standard Air and the Rayleigh Scattering Coefficient for the Spectral Region between 0.2 and 20.0 μ and Their Application to Atmospheric Optics." *Journal of the Optical Society of America* 1957: 47(2), pp. 176–182. Reprinted in [2].

[37] Perlin, K., 1985, "An Image Synthesizer." *Computer Graphics* 1985: 19(3), pp. 287–296.

[38] Pokorny, J., and Smith, V. C., 1986 "Colorimetry and Color Discrimination." In Boff, K. R., Kaufman, L., and Thomas, J. P. (eds.), 1986, *Handbook of Perception and Human Performance.* John Wiley & Sons, Volume 1, Chapter 8.

[39] Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P., 1992, *Numerical Recipes in C.* Cambridge University Press, 994 pages. ISBN 0-521-43108-5.

[40] Rogers, R. R., and Yau, M. K., 1989, *A Short Course in Cloud Physics.* Third edition. Pergamon Press, 293 pages. ISBN 0-08-034863-7.

[41] Rozenberg, G. V., 1960, "Light Scattering in the Earth's Atmosphere." *Soviet Physics Uspekhi*, 1960: 3(3), pp. 346–371. Reprinted in [2].

[42] Rushmeier, H. E., and Torrance, K. E., "The Zonal Method for Calculating Light Intensities in the Presence of a Participating Medium." *Computer Graphics* 1987: 21(4), pp. 293–302.

[43] Sakas, G., 1993, "Modeling and animating turbulent gaseous phenomena using spectral synthesis." *The Visual Computer* January 1993: 9(4), pp. 200–212.

[44] Smolarkiewicz, P. K., and Clark, T. L., 1985, "Numerical Simulation of the Evolution of a Three-Dimensional Field of Cumulus Clouds. Part I: Model Description, Comparison with Observations and Sensitivity Studies." *Journal of the Atmospheric Sciences* 1985:42, pp. 502–522.

[45] Srivastava, R. C., 1967, "A Study of the Effect of Precipitation on Cumulus Dynamics." *Journal of the Atmospheric Sciences* 1967:24, pp. 36–45.

[46] Steiner, J. T., 1973, "A Three-Dimensional Model of Cumulus Cloud Development." *Journal of the Atmospheric Sciences* 1973:30, pp. 414–435.

[47] Voss, R. F., 1988, "Fractals in nature: From characterization to simulation." In Peitgen, H-O., and Saupe, D. (eds.), 1988, *The Science of Fractal Images.* Springer-Verlag, ISBN 3-540-96608-0. pp. 21–70.

[48] Warner, J., 1969, "The Microstructure of Cumulus Cloud. Part I. General Features of the Droplet Spectrum." *Journal of the Atmospheric Sciences* 1969:26, pp. 1049–1059.

[49] Warner, J., 1969, "The Microstructure of Cumulus Cloud. Part II. The Effect on Droplet Size Distribution of the Cloud Nucleus Spectrum and Updraft Velocity." *Journal of the Atmospheric Sciences* 1969:26, pp. 1272–1282.

[50] Whitehouse, D., 1994, "A Discrete Solution of the Scattering Equation for Computer Graphics.", *Australian National University Supercomputer Facility Technical Report 93-01.* 8 pages.

[51] Woo, M., Neider, J., and Davis, T., 1997, *OpenGL Programming Guide.* Addison-Wesley Developers' Press, 650 pages. ISBN 0-201-46138-2.

[52] Yau, M. K., and Michaud, R., 1982, "Numerical Simulation of a Cumulus Ensemble in Three Dimensions." *Journal of the Atmospheric Sciences* 1982:39, pp. 1062–1079.

Appendix A

Tables

Table A.1: Saturation vapor pressure, latent heat of vaporization, thermal conductivity of air and coefficient of diffusion of water vapor in air. (From [40].)

T (°C)	e_s (Pa)	L (J/g)	K ($\mathrm{Jm^{-1}s^{-1}K^{-1}}$)	D ($\mathrm{m^2s^{-1}}$)
-40	19.05	2603	$2.07 \cdot 10^{-2}$	$1.62 \cdot 10^{-5}$
-35	31.54			
-30	51.06	2575	$2.16 \cdot 10^{-2}$	$1.76 \cdot 10^{-5}$
-25	80.90			
-20	125.63	2549	$2.24 \cdot 10^{-2}$	$1.91 \cdot 10^{-5}$
-15	191.44			
-10	286.57	2525	$2.32 \cdot 10^{-2}$	$2.06 \cdot 10^{-5}$
-5	421.84			
0	611.21	2501	$2.40 \cdot 10^{-2}$	$2.21 \cdot 10^{-5}$
5	872.47	2489		
10	1227.94	2477	$2.48 \cdot 10^{-2}$	$2.36 \cdot 10^{-5}$
15	1705.32	2466		
20	2338.54	2453	$2.55 \cdot 10^{-2}$	$2.52 \cdot 10^{-5}$
25	3168.74	2442		
30	4245.20	2430	$2.63 \cdot 10^{-2}$	$2.69 \cdot 10^{-5}$
35	5626.45	2418		
40	7381.27	2406		

Table A.2: The ICAO 1964 Standard Atmosphere. Altitude is measured in meters above mean sea level. (From [40].)

z (m)	T (K)	p (kPa)	ρ (kg/m^3)
0	288.150	101.325	1.2250
200	286.850	98.945	1.2010
400	285.550	96.611	1.1786
600	284.250	94.322	1.1560
800	282.951	92.077	1.1337
1000	281.651	89.876	1.1117
1200	280.351	87.718	1.0900
1400	279.052	85.602	1.0687
1600	277.753	83.528	1.0476
1800	276.453	81.494	1.0269
2000	275.154	79.501	1.0066
2500	271.906	74.692	0.9570
3000	268.659	70.121	0.9093
3500	265.413	65.780	0.8634
4000	262.166	61.660	0.8194
4500	258.921	57.753	0.7770
5000	255.676	54.048	0.7364
5500	252.431	50.539	0.6975
6000	249.187	47.218	0.6601
6500	245.943	44.075	0.6243
7000	242.700	41.105	0.5900
7500	239.357	38.300	0.5572
8000	236.215	35.652	0.5258
8500	232.974	33.154	0.4958
9000	229.733	30.801	0.4671
9500	226.492	28.585	0.4397
10000	223.252	26.500	0.4135
10500	220.013	24.540	0.3886
11000	216.774	22.700	0.3648
11500	216.650	20.985	0.3374
12000	216.650	19.399	0.3119
12500	216.650	17.934	0.2884
13000	216.650	16.580	0.2666
13500	216.650	15.328	0.2465
14000	216.650	14.170	0.2279
14500	216.650	13.101	0.2107
15000	216.650	12.112	0.1948

Appendix B

The pseudo-adiabatic process

This appendix contains a derivation of dT/dp for the pseudoadiabatic process.

The saturation mixing ratio w_s is defined as

$$w_s = \varepsilon \frac{e_s}{p - e_s} \approx \varepsilon \frac{e_s}{p} \tag{B.1}$$

e_s can be approximated by

$$e_s \approx A e^{-B/T} \tag{B.2}$$

Combining equations (B.1) and (B.2) gives

$$w_s \approx \varepsilon \frac{A e^{-B/T}}{p} \tag{B.3}$$

We have

$$\frac{\partial w_s}{\partial p} \approx -\frac{\varepsilon A e^{-B/T}}{p^2} \tag{B.4}$$

$$\frac{\partial w_s}{\partial T} \approx \frac{\varepsilon A B e^{-B/T}}{p T^2} \tag{B.5}$$

so

$$dw_s \approx \varepsilon A e^{-B/T} \left(-\frac{dp}{p^2} + B \frac{dT}{pT^2} \right) \tag{B.6}$$

By combining the approximation for the pseudoadiabatic process

$$\frac{dT}{T} = k \frac{dp}{p} - \frac{L}{T c_p} dw_s \tag{B.7}$$

with (B.6), we get

$$\frac{1}{T} dT = \frac{k}{p} dp + \frac{L \varepsilon A e^{-B/T}}{T c_p p^2} dp - \frac{L \varepsilon A B e^{-B/T}}{T^3 c_p p} dT \tag{B.8}$$

$$\Leftrightarrow$$

$$\left(\frac{1}{T}+\frac{L\varepsilon ABe^{-B/T}}{T^3c_pp}\right)dT=\left(\frac{k}{p}+\frac{L\varepsilon Ae^{-B/T}}{Tc_pp^2}\right)dp \tag{B.9}$$

$$\Leftrightarrow$$

$$\frac{dT}{dp}=\frac{\left(\frac{k}{p}+\frac{L\varepsilon Ae^{-B/T}}{Tc_pp^2}\right)}{\left(\frac{1}{T}+\frac{L\varepsilon ABe^{-B/T}}{T^3c_pp}\right)} \tag{B.10}$$

Appendix C

Constants

Universal gas constant	$R^* = 8.314$ J mole^{-1} K^{-1}
Individual gas constant for dry air	$R' = 287$ J kg^{-1} K^{-1}
Individual gas constant for water vapor	$R_v = 461.5$ J kg^{-1} K^{-1}
Specific heat capacity at constant pressure	$c_p = 1005$ J kg^{-1} K^{-1}
Specific heat capacity at constant volume	$c_p = 718$ J kg^{-1} K^{-1}
Density of water	$\varrho_L = 1$ g cm^{-3}
Acceleration due to gravity	$g = 9.81$ m s^{-2}
Molecular number density of the standard atmosphere	$N = 2.68731 \cdot 10^{19}$ cm^{-3}
Index of refraction of air	$n = 1.00027$

Appendix D

Symbols and units

Temperature	T	[K]
Potential temperature	θ	[K]
Virtual temperature	T_v	[K]
Dew point temperature	T_d	[K]
Isentropic condensation temperature	T_c	[K]
Dry adiabatic lapse rate	Γ	[K m^{-1}]
Pseudoadiabatic lapse rate	Γ_s	[K m^{-1}]
Pressure	p	[Pa]
Water vapor pressure	e	[Pa]
Saturation vapor pressure	e_s	[Pa]
Water vapor density, absolute humidity	ϱ_v	[kg m^{-3}]
Volume	V	[m^3]
Mass	M	[kg]
Molecular weight	m_s	[kg mole^{-1}]
Specific volume	α	[m^3 kg^{-1}]
Time	t	[s]
Mixing ratio	w	[g kg^{-1}]
Saturation mixing ratio	w_s	[g kg^{-1}]
Specific humidity	q	[g kg^{-1}]
Relative humidity	f	[per cent]
Latent heat of vaporization	L	[J kg^{-1}]
Saturation ratio	S	[per cent]
Droplet radius	r	[m]
Coefficient of thermal conductivity	K	[J m^{-1} s^{-1} K^{-1}]
Dynamic viscosity of air	D	[m^2 s^{-1}]
Altitude	z	[m]
Liquid water content	χ	[g kg^{-1}]
Buoyant force	F_B	[N]
Wavelength of light	λ	[m]
Intensity of light	I	[W]