Real-Time Simulation of Dust Behavior Generated by a Fast Traveling Vehicle

JIM X. CHEN, XIAODONG FU, and EDWARD J. WEGMAN
Computer Graphics Laboratory, George Mason University

Simulation of physically realistic complex dust behavior is very useful in training, education, art, advertising, and entertainment. There are no published models for real-time simulation of dust behavior generated by a traveling vehicle. In this paper, we use particle systems, computational fluid dynamics, and behavioral simulation techniques to simulate dust behavior in real time. First, we analyze the forces and factors that affect dust generation and the behavior after dust particles are generated. Then, we construct physically-based empirical models to generate dust particles and control the behavior accordingly. We further simplify the numerical calculations by dividing dust behavior into three stages, and establishing simplified particle system models for each stage. We employ motion blur, particle blending, texture mapping, and other computer graphics techniques to achieve the final results. Our contributions include constructing physically-based empirical models to generate dust behavior and achieving simulation of the behavior in real time.

Categories and Subject Descriptors: I.6.5 [**Simulation and Modeling**]: Model Development—*Modeling methodologies*

General Terms: Simulation, Graphics, Dust

Additional Key Words and Phrases: Physically-based Modeling, Real-time Simulation, Vehicle, Particle Systems, Computational Fluid Dynamics

1. INTRODUCTION

In many virtual environments and distributed interactive simulations, it is desirable to simulate trucks, armored vehicles, bulldozers, and other ground-based moving objects [Balci 1997; Chen et al. 1997; Cremer et al. 1995; Schiavone et al. 1997; IST 1994; Li and Moshell 1993]. Typically,

The work of Dr. Wegman was supported by the Army Research Office under contract DAAH04-94-G-0267 and by a National Science Foundation Group Infrastructure Grant DMS-9631351.

Authors' addresses: J. X. Chen and X. Fu, Computer Graphics Laboratory, George Mason University, Fairfax, VA 22030; email: jchen6@gmu.edu; xfu@gmu.edu; E. J. Wegman, Center for Computational Statistics, George Mason University, Fairfax, VA 22030; email: ewegman@galaxy.gmu.edu.

dust behavior is not simulated when these objects travel on an unpaved road. Dust behavior is caused by different factors (such as natural wind and a fast traveling vehicle) and appears in many situations. Simulating physically realistic, complex dust behavior is very useful in interactive graphics applications, such as computer art, advertising, education, entertainment, and training. However, due to the lack of modeling and simulation techniques, there are currently no published successful real-time simulations of realistic dust behavior. As computers and their graphics systems become faster, many natural phenomena (such as the behaviors of fluids, terrains, trees, fireworks, volcanos, clouds, falling leaves, etc.) may be simulated in real time [Chen and Lobo 1995; Li and Moshell 1993; Loke et al. 1992; Oppenheimer 1986; Reeves 1983; Reynolds 1987; Roth and Guritz 1995; Sims 1990; Stam and Fiume 1993; Wejchert and Haumann 1991]. We believe it is appropriate now to include dust behavior into real-time simulation.

Hsu and Wong [1995] introduced a dust accumulation model. Their model presents static appearance of dust accumulation without behavior and animation. Cowherd [1989], Williams [Williams and Davis 1988], and other researchers studied dust and the mechanisms of dust generation. Their purpose was to study and measure the density of the dust in the real battlefield rather than simulating the dust behavior graphically. Today, military training using graphics and distributed interactive simulation is an important topic of research and applications [IST 1994], and generating dust behavior in real time can significantly increase the realism of the simulated training environment.

In this paper, we introduce a method for simulating the dust behavior caused by a fast traveling vehicle in real time. Specifically, we can calculate and render the dust behavior at more than four frames per second (with 2000 particles animated at the same time) on a SGI OnyxII graphics computer. The method exploits a combination of particle systems, rigid body particle dynamics, computational fluid dynamics (CFD), and behavioral simulation techniques. The *particle systems* technique was first introduced to computer graphics by Reeves [1983], and is now widely used to simulate fuzzy or dynamic objects, such as fire, grass, explosions, clouds, water, and trees. These objects have no fixed shape and change their shapes and behavior stochastically. They have ill-defined boundaries that make surface-based modeling impractical. Dust behavior behind a moving vehicle belongs to this category. CFD methods are employed to generate the dynamics of air flow behind a moving vehicle. We also employ motion blur for small and fast-moving particles, particle blending instead of hidden-surface removal, texture mapping, and other graphics techniques to achieve both better performance and also better appearance in the final results.

In order to construct a physically based realistic simulation, we first analyze the forces and factors that affect dust generation and the behavior after dust particles are generated. Then we construct physically based empirical models for generating dust particles and for controlling the

behavior. The particle systems are integrated with CFD to achieve better realism. However, the models are time-consuming and inefficient. Based on these general models and the analysis of forces, we simplify the numerical calculations by dividing dust behavior into three stages, and establishing simplified particle system models for each stage. The resulting models are satisfactory for real-time simulation as well as for achieving realistic dust behavior.

The rest of the paper is organized as follows: In Section 2, we briefly describe the particle systems technique that is employed in our dust simulation. In Section 3, we discuss the generation of a dust particle, the turbulent air flow affecting the behavior after a dust particle is generated, and the dynamics of a dust particle. In Section 4, we simplify the models, and divide the dust particle system into three stages (*fluid turbulence*, *particle momentum*, and *airborne drift*). In Section 5, we discuss some rendering issues and present some simulation results with different parameters. Finally, in Section 6, we summarize and describe several avenues of future research.

2. PARTICLE SYSTEMS

The original particle systems method proposed by Reeves [1983] was based on stochastic processes. Each particle is independent and moves according to its own characteristics. He hypothesized no interaction among particles in a particle system. Structured particle systems, which are used to simulate objects that are more structured (such as trees, grass, etc.), were also proposed by Reeves. Sims [1990] extended Reeves' work and allowed the independent particles to interact with the environment. Today, there are numerous publications on particle systems. The interactions among particles in a particle system and among particle systems are very important for simulating some phenomena. For example, a particle system representing fire could interact with another particle system representing water. This will result in a new particle system representing steam.

2.1 Properties

A particle-based simulation may include a number of independent particle systems. Each particle system is made of many particles and each particle has its own parameters that significantly influences the particles' properties. In general, a particle system and its particles have very similar parameters, but with different values:

—Position (including orientation in 3D space and center location x, y, and z)

—Movement (including velocity, rotation, acceleration, etc.)

—Color (RGB)

—Transparency (alpha)

—Shape (point, line, sphere, cube, rectangle, etc.)

—Volume

—Density

—Mass

—Lifetime (only for particles)

—Blur head and rear pointers (only for particles)

The position, shape, and size of a *particle system* determine the initial positions of the *particles* and their range of movement. The movements of the particles are restricted within the range defined by their associated particle system. The shape of a particle system can be a point, line segment, sphere, box, or cylinder. The movement of a particle system is affected by internal or external forces, and the results of the rotations and accelerations of the particles as a whole. A particle system may change its shape, size, color, transparency, or some other attributes as it evolves.

The lifetime defines how many time slices (frames) a particle will be active. A particle has both a head position and a tail position. The head position is usually animated and the tail position follows along for motion blur. The simulation looks more dynamic and has more particles with motion blur, at the cost of longer rendering time.

2.2 Simulation

In general, particle systems are first initialized with each particle having its original position, velocity, color, transparency, shape, size, mass, and lifetime. After the initialization, for each simulation frame, some parameters of the particles are updated using a rule base, and the resulting particle systems are rendered. An outline of a general simulation loop for a particle system is as follows:

—Create particle systems: ranges of particle activities; data structures to hold particles

—Initialize each particle's parameters (position, velocity, color, transparency, shape)

—For each simulation frame, do the following:
  —Create and remove particles
  —Calculate and update each particle's parameters (position, velocity)
  —Adjust each particle's motion blur (head and rear) pointers
  —Render the particles in the particle systems
  —Update the particle systems' ranges of activities

2.3 Applications

Figure 1 summarizes applications using particle systems or particle-based simulation in computer graphics.

As shown above, a particle system has different shape, color, transparency properties, and movement governed by certain underlying processes.

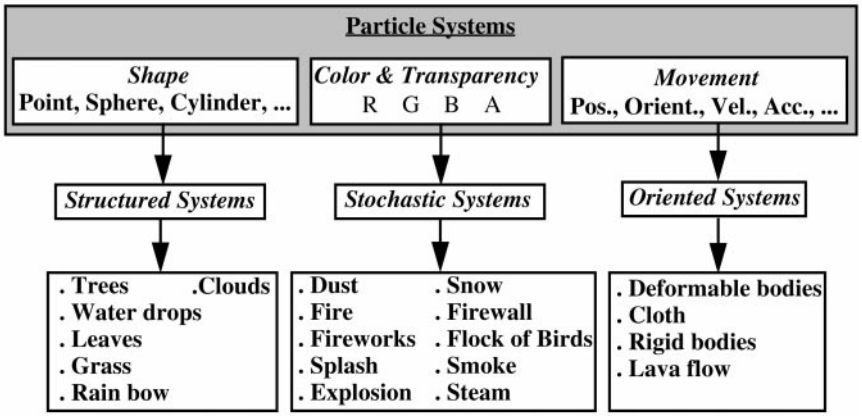


Fig. 1. The applications of particle systems.

Structured particle systems are often used to model trees, water drops, leaves, grass, rainbow, and clouds. Stochastic particle systems are often used to model fireworks, explosions, snow, and so forth. Oriented particle systems are often used to model deformable and rigid bodies such as cloth, lava flow, etc.

Figure 2 is an example of using particle systems to model grass from the Center for Computational Statistics, George Mason University.

Dust behavior is fuzzy and unstructured and, thus, belongs to the category of stochastic particle systems as indicated in Figure 1. In the following section, we construct a stochastic particle system to simulate dust behavior caused by a fast-traveling vehicle.

3. DUST BEHAVIORS

In this section, we introduce the methods for the generation of dust particles, discuss the dynamics of air flow around the vehicle that affect the dust behavior, and analyze the forces acting on a dust particle in order to establish corresponding physically based empirical models.

3.1 Dust Generation

There are many factors affecting the generation of dust particles (number of particles, initial locations, and velocities) as well as affecting the behavior of the particles after their generation. Here we develop correlations among the factors and the dust generation process, so that when we have a different vehicle or the same vehicle in a different environment, we can change the simulations just by adjusting the corresponding parameters.

As a vehicle wheel passes over an unpaved surface with velocity ($\mathbf{V_{car}}$) vertical pressure, due to the weight of the vehicle (WT_{car}), will produce ground surface vibration and deformation, crushing large particles into smaller ones, and splashing particles into the air (Figure 3(a), left). WT_{car}

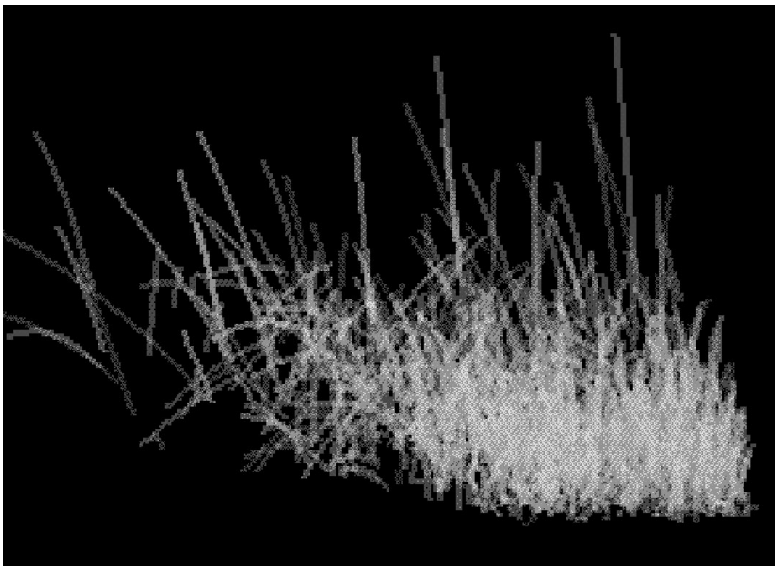


Fig. 2. Grass.

and $\mathbf{V_{car}}$ both affect the number of particles generated. $\mathbf{V_{car}}$ also affects the initial velocity of the particle generated at the bottom of the tire.

Horizontal stress and friction are largely due to the driving power that sustains the velocity ($\mathbf{V_{car}}$) and acceleration of the vehicle and will further pulverize the particles and carry them on the surface of the tire. The slippage between the tire and the ground surface will lift particles of different sizes due to the adhesive and shear forces, and eject them at different places on the tire surface due to the centrifugal force and air drag force (Figure 3(a), right). $\mathbf{V_{car}}$ decides the initial speed of the particle on the tire surface. Angular velocity of the tire ($\omega_{\mathbf{tire}}$), radius of the tire (R_{tire}), particle volume or size (ϑ_p), particle density (ρ_p), and particle stickiness (σ_p) affect the location of the particle leaving the tire surface.

The pressure gradient formed by the moving air underneath and behind the vehicle will also lift fine particles (Figure 3(b)). The initial velocity of the particle depends on the energy associated with the pressure gradient in the vertical direction. The length (L_{car}), height (H_{car}), and width (W_{car}) of the vehicle are used together to calculate the pressures and velocities in the fluid region (introduced in the next section).

There are many other important factors associated with the conditions of the environment that affect dust generation: the environmental wind ($\mathbf{V_{wd}}$), the density of dust on the ground (ρ_{gd}), the average volume or size of particles on the ground (ϑ_{gd}), and the adhesion and wetness of the ground (σ_{gd}). If the ground is wet and the average size of the particles is large, there will be fewer particles. If the dust density of the ground surface is high, there will be more particles.

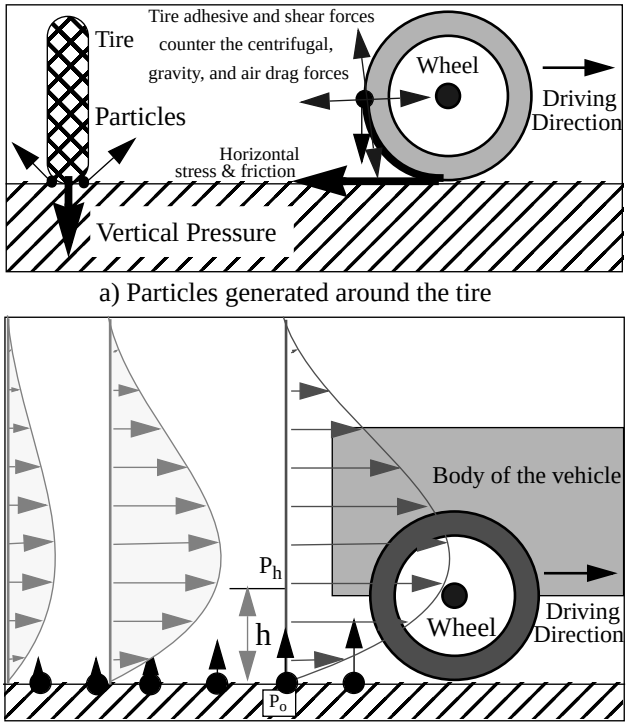


Fig. 3. Forces between the wheels and the ground.

In summary, most dust particles are generated right behind and alongside of the wheels. Some fine dust particles are lifted from the ground surface because of the pressure gradient underneath and behind the vehicle. Each particle is generated with its own initial density, size, position, and velocity. In addition to the parameters that are used to affect the dust generation as listed in Table 1, a particle's shape affects the direction of its velocity [Wejchert and Haumann 1991], which is included as random perturbation of the velocity. Particle mass and size are randomly generated with a Poisson distribution.

After a particle is generated with its mass ($m_p = \rho_p \vartheta_p$), stickiness, initial position and velocity, it is then entrained in the turbulent air behind the vehicle. It will eventually return to the ground depending on its properties and environment conditions. Bigger particles will fall back to the ground surface more rapidly while the fine ones will remain suspended in the air. Small stones and blocks of muds may fall back to the ground immediately after ejection from the tires. The turbulent air around and behind the vehicle is calculated according to fluid dynamics as described in the next section.

Table I. Parameters affecting the dust generation

Items	Parameters	Description
Vehicle	Velocity - $\mathbf{V}_{car}$ Weight - WT_{car} Tire radius - R_{tire} Height - H_{car} Width - W_{car} Length - L_{car}	Decide the velocity of the fluid and boundary of the car where the turbulence is generated; determine the pressures and velocities of the fluid field; Decide where a dust particle leaves the tire
Dust Particle	Volume (size) - ϑp Density - ρ_p Stickiness - σ_p Shape (included as random perturbation of velocities)	Decide where a dust particle leaves the tire and ground; affect the number of particles generated
Environment	Ground density - ρ_{gd} Average Volume (size) - ϑ_{gd} Ground wetness - σ_{gd} Environmental Wind - $\mathbf{V}_{wd}$	Influence the number of particles generated and their initial velocities.

3.2 Fluid Dynamics Affecting Dust Behaviors

The physical problem of the air flow around a moving vehicle belongs to the well-studied subject of aeromechanics which is called separated flow [Burggraf 1966; Prandtl and Tietjens 1934]. Burggraf [1966] summarized the various regimes of flow experienced by circular cylinder in an incompressible fluid (Figure 4). The appearance of a wake first occurs at Reynolds Number [1] (Re) of the order of 1, and the flow separates from the rear of the cylinder, forming a recirculating eddy for Reynolds number greater than 5; the steady recirculating eddy persists at Reynolds number about the order of 100; the steady flow then breaks down into the Karman vortex street; finally the stream develops into a completely turbulent wake for Reynolds number greater than the order of 10^5.

Considering a vehicle at speed $\mathbf{V_{car}} = 60$ km/hour, vehicle size of the order of $L = 1$ meter, and an ordinary air kinematics viscosity of 10^{-5}, the Reynolds numbers (Re) at the boundary and right behind the vehicle are then in the order of 10^6. Therefore, turbulence is the main feature of the flow at the boundary of the vehicle movement in our model. The study of turbulence is still not satisfactory [Burggraf 1966]. The situation is worse in simulation with numerical models: even the most advanced models of today, such as direct numerical simulation (DNS), large eddy simulation (LES), or Reynolds average equations (RANS), are deficient for large Reynolds numbers [Gatski 1996]. As for graphics simulation applica-

[1] $Re = VL/v$ where L and V are characteristic length and velocity respectively, and v is the kinematic viscosity. For example, given fluid flow inside a pipe, then L can be the diameter of the pipe and V the velocity of the fluid flow.

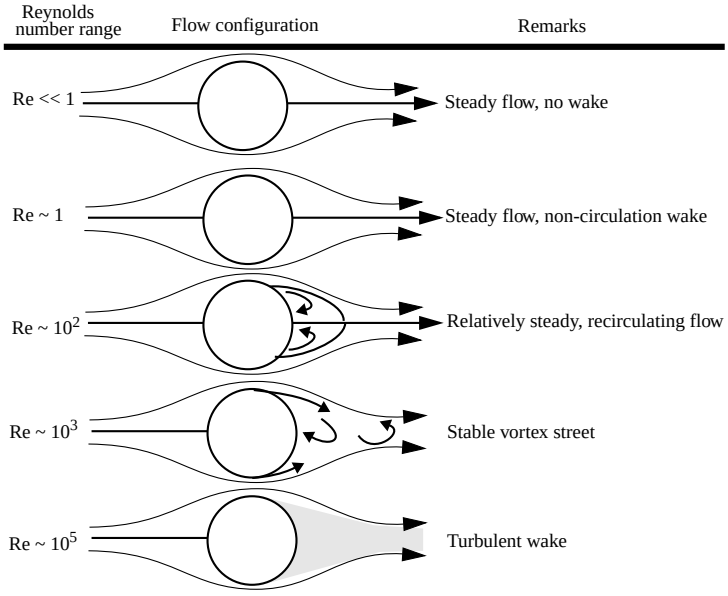


Fig. 4. Flow regimes for circular cylinder (from Burggraf [1966]).

tions, it is not necessary to simulate turbulence in such detail, which is almost random motion in higher order scales. Therefore, in this paper, we assume that the flow around a moving vehicle can be approximated as a steady, low Reynolds external laminar flow modified by a random velocity in the high Reynolds turbulent region. Here we choose the separation between the turbulent flow and the laminar flow by the tangential direction of downstream along the boundary of the vehicle (Figure 5).

Our assumptions on the flow in the low Re region (in terms of physical nature) are steady, low speed, and incompressible. Gravity force of the air is ignored. The incompressible Navier-Stokes equations [Batchelor 1967; Burggraf 1966] in nondimensional form for a Cartesian coordinate system are then given by continuity equation and momentum equations along x, y, and z coordinates:

$$u_x + v_y + w_z = 0 \tag{1}$$

$$u_t + uu_x + vu_y + wu_z = -\rho_x + \frac{1}{Re}(u_{xx} + u_{yy} + u_{zz}) \tag{2}$$

$$v_t + uv_x + vv_y + wv_z = -\rho_y + \frac{1}{Re}(v_{xx} + v_{yy} + v_{zz}) \tag{3}$$

$$w_t + uw_x + vw_y + ww_z = -\rho_z + \frac{1}{Re}(w_{xx} + w_{yy} + w_{zz}) \tag{4}$$

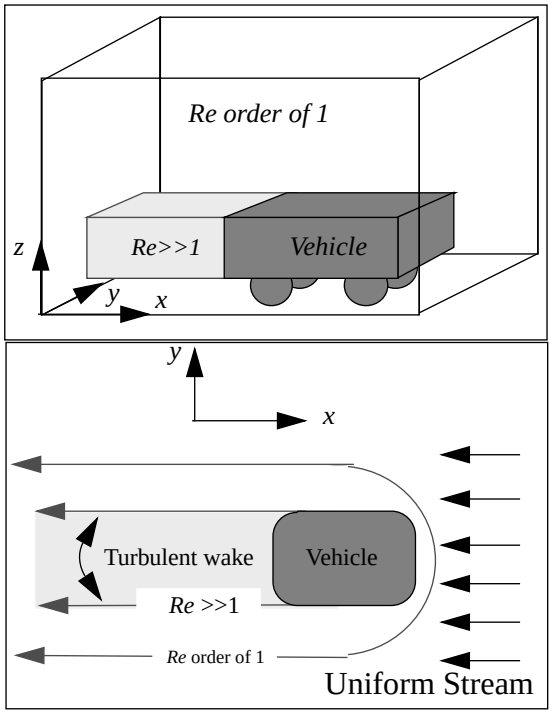


Fig. 5. Regions of different dynamics of flow around a moving vehicle.

where $u = dx/dt$, $v = dy/dt$ and $w = dz/dt$ are velocities along x, y, and z axes; $u_x = \partial u/\partial x$, $u_{xx} = \partial^2 u/\partial x^2$, and so forth similarly for partial derivatives along y and z and for v and w; p is the pressure and $\rho_x = \partial p/\partial_x$, Re is the Reynolds number.

One of the most common approaches to solve the above Navier-Stokes equations uses the Poisson equation instead of the continuity equation to calculate the pressures [Tannehill and Pletcher 1984]. This separates the majority of the "pressure effects" from the partial derivatives to become a single equation so that the elliptic nature of the flow can be suitably modeled. Here,

$$\Delta^2 \rho = D_p - \frac{\partial(u_x + v_x + w_x)}{\partial t} \tag{5}$$

where

$$D_p = \frac{d}{dx}\left(- (uu_x + vu_y + wu_z) + \frac{1}{Re}(u_{xx} + u_{yy} + u_{zz})\right)$$

$$+ \frac{d}{dy}\left(- (uv_x + vv_y + wv_z) + \frac{1}{Re}(v_{xx} + v_{yy} + v_{zz})\right)$$

$$+ \frac{d}{dz}\left(- (uw_x + vw_y + ww_z) + \frac{1}{Re}(w_{xx} + w_{yy} + w_{zz})\right). \tag{6}$$

We employ Ghia's approach [Ghia et al. 1977; 1979] to solve Eqs. (5) and (6) numerically. In Ghia's approach, an Alternating-Direction Implicit (ADI) scheme is applied to the momentum equations and a Successive-Over-Relaxation (SOR) method is then used to solve the Poisson pressure equation. The solution is stable.

The boundary of the vehicle is considered as in Figure 5. For the momentum equations (2) to (4), the normal velocities are zero at the vehicle surface, while the tangential velocities satisfy the condition of zero slip at all walls. The velocities of inflow, outflow, and on the top boundary are fixed. For the Poisson equation (5), the boundary conditions consist of the normal gradient ($\partial p / \partial n$) evaluated at the interior boundary [Ghia et al. 1977].

The fluid dynamics results describe the laminar flow around the vehicle, where the velocity (relative to the vehicle) $\mathbf{V}$ and pressure p at any point in the flow volume are calculated. We have

$$\mathbf{V} = u\mathbf{i} + v\mathbf{j} + w\mathbf{k} \tag{7}$$

$$\text{and } V = |\mathbf{V}| = \sqrt{u^2 + v^2 + w^2}. \tag{8}$$

When a vehicle travels quickly, it sweeps out a volume in which the air pressure is lower than the surrounding areas and the flow is highly turbulent. In order to reproduce the turbulent nature of the separated flow, we add random velocities to the steady flow. The random velocities are uniformly distributed in all directions, and their expected values are zero. This implies that the turbulence is homogeneous and isotropic. The variances of the random velocities are proportional to the ensemble average of the turbulence energies

$$Var(\mathbf{V}') = \frac{1}{2}\langle V'_i V'_j \rangle \tag{9}$$

where $\langle V'_i V'_j \rangle$ is the turbulence stress, which depends on the shear structure of the fluid. For example, in Prandtl's boundary layer theory [Prandtl and Tietjens 1934], the magnitude is approximated by $|L(dV/dy)|^2$, , where L is the size (L_{car}, W_{car}, H_{car}) of the vehicle. Therefore, the velocity of the air without considering the environmental wind and the velocity of the vehicle is:

$$\mathbf{V}_{\mathbf{air}} = \mathbf{V} + \mathbf{e}\left|L\frac{dV}{dy}\right|^2. \tag{10}$$

Here e is a random unit vector.

As shown in Figure 5, we have to admit that our treatment of the vehicle's shape as a box is a crude approximation. The actual vehicle shape is complex. Although the approximation may cause unrealistic flow close to

the vehicle's outside boundary, the error influencing dust particle behavior is at particle initialization. The error effect only lasts for a very short period and so we choose to neglect it for simplification purposes.

3.3 Dynamics of a Dust Particle

The computation of the dynamics of a dust particle includes three stages: dust generation, dust movement, and dust extinction. A dust particle is generated with its initial position and velocity, then it will be entrained in the fluid flow. When a dust particle falls back to the ground, we consider the dust particle to be extinct.

The aspects of dust generation are discussed in Section 3.1. Here we calculate the initial velocity of a particle generated by the pressure gradient that is formed by the moving air (Figure 3(b)). The pressure of the moving air behind and underneath a vehicle follows Bernoulli's theorem:

$$\frac{p_h}{\rho_{air}} + gh + \frac{1}{2}V_h^2 = \frac{p_0}{\rho_{air}}. \tag{11}$$

where p_h and V_h are the pressure and the velocity at height h; p_0 is the pressure on the ground. p_h and V_h are known from our CFD results and the velocity of the vehicle. Thus, between the ground and the bottom of the vehicle of height h, there is a pressure gradient:

$$p_0 - p_h = \rho_{air}\left(gh + \frac{1}{2}V_h^2\right). \tag{12}$$

Our treatment for the lifting process is to convert the pressure gradient energy to the dust initial disturbance kinetic energy. The pressure gradient will, according to the energy conservation law, raise the dust particles from the ground. For instance, a particle will obtain momentum energy from the pressure gradient force when it moves from ground to height h and the total momentum energy obtained is proportional to the pressure difference between the two levels.

$$\frac{1}{2}m_p V_0^2 = \alpha A h(p_0 - p_h), \tag{13}$$

where α, a number between 0 and 1, is a chosen ratio of the momentum energy from the pressure gradient applied to the particle. A is the cross-section area of the particle; therefore, we have the initial speed of the particle:

$$V_0 = \sqrt{\frac{\alpha A h \rho_{air}(2gh + V_h^2)}{\rho_p \vartheta_p}}. \tag{14}$$

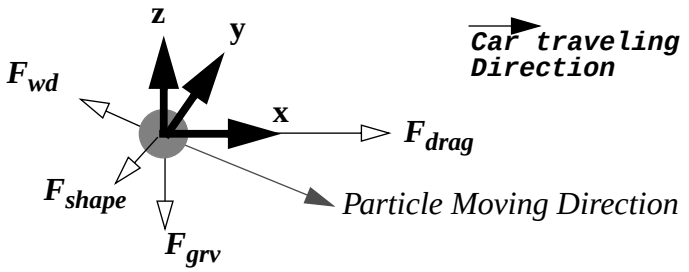


Fig. 6. Forces acting on a dust particle.

The direction of the initial velocity is perpendicular to the ground because the pressure is vertical. The initial velocity of a particle behind the vehicle is also calculated at the same height. From the above equation, the initial velocity of a particle will be larger if the ratio of the air density to the particle density is larger. This means that the low density dust particles will rise from the ground faster.

Once a dust particle enters the air, the motion of the particle will obey Newton's law. The external forces include the gravity force and the air friction drag because of the turbulent air flow around the vehicle and the environmental wind. In our treatment, the environmental wind is superimposed on top of the fluid flow. Therefore, we consider it separately from the air friction drag because of the moving vehicle. As discussed by Wejchert and Haumann [1991], the velocity of an object in fluid flow is affected by its shape. The shape of the particle is not considered here, but a small randomly generated force is added to account for the effect of the particle shape. Therefore, the forces acting on a particle include

—The turbulent air friction drag force because of the vehicle movement: $\mathbf{F}_{\mathbf{drag}}$

—The wind: $\mathbf{F}_{\mathbf{wd}}$ which is imposed on top of the turbulent *air friction drag*

—The dust particle's gravity force: $\mathbf{F}_{\mathbf{grv}}$.

—The dust particle's shape: $\mathbf{F}_{\mathbf{shape}}$

In summary, all the forces acting on a particle are shown in Figure 6. Here we ignore any collisions among the dust particles.

We use Batchelor's expression [Batchelor 1967] that for a three-dimensional body of surface area A in a stream of speed U, the total friction drag (F_{drag}) is

$$F_{drag} = k\rho_p U^2 A Re^{-\frac{1}{2}}, \tag{15}$$

where k is a number depending only on the shape of the particle. We use $k = 1$ in the present implementation to approximate a simple shape. $Re = LU/v$ and U is the relative speed between the air and the particle:

$$U = |(\mathbf{V_{air}} + \mathbf{V_{car}} + \mathbf{V_{wd}} - \mathbf{V_p})|, \tag{16}$$

where $\mathbf{V_{wd}}$ is the velocity of the wind at the point of the particle relative to the ground, $\mathbf{V_p}$ is the current velocity of the particle, $\mathbf{V_{air}}$ is the current velocity of the fluid at the point of the particle relative to the car, and $\mathbf{V_{car}}$ is the velocity of the car. The drag force corresponds to the particle's acceleration. Therefore, the magnitude as well as direction of the drag on the particle is:

$$\mathbf{F_{drag}} = \frac{(\mathbf{V_{air}} + \mathbf{V_{car}} + \mathbf{V_{wd}}) - \mathbf{V_p}}{U} F_{drag}. \tag{17}$$

Let $\mathbf{F_p}$ be the total force acting on a dust particle, $\mathbf{P}$ the position, $\mathbf{V_p}$ the velocity, $\mathbf{A_p}$ the acceleration, and $m_p = \rho_p \vartheta_p$ the mass of the particle. Then a dust particle's behavior is described by the following equations:

$$\mathbf{F_p} = \mathbf{F_{drag}} + \mathbf{F_{shape}} + \mathbf{F_{grv}} \tag{18}$$

$$\mathbf{A_p} = \frac{\mathbf{F_p}}{m_p}. \tag{19}$$

The wind itself can be a velocity field. It is also changeable every frame. Therefore

$$\mathbf{V_p} = \mathbf{V_0} + \mathbf{V_{wd}} + \int_{t_0}^{t_1} \mathbf{A_p} dt \tag{20}$$

$$\mathbf{P} = \mathbf{P_0} + \int_{t_0}^{t_1} \mathbf{V_p} dt. \tag{21}$$

To simplify the calculation, we use Euler's method to approximate the particle's next state:

$$\mathbf{V_i} = \mathbf{V_{i-1}} + \mathbf{A_p} \cdot \Delta t \tag{22}$$

$$\mathbf{P_i} = \mathbf{P_{i-1}} + \mathbf{V_i} \cdot \Delta t. \tag{23}$$

The algorithm to compute the solution to the dust behavior then is as follows: For the known current state of a particle $\{V_{i-1}, P_{i-1}\}$, the next state $\{V_i, P_i\}$, after Δt time, is calculated by Eqs. (22) and (23). These equations use Eqs. (15) to (19). The initial velocity of the particle under and behind the vehicle is given by Eq. (14) which is added to the current wind velocity. Given a vehicle model, we define a finite grid coordinate in a box volume. The grid box volume will be the 3D flow field for fluid dynamics computation. The width of the box volume is about 3-5 times of the vehicle's

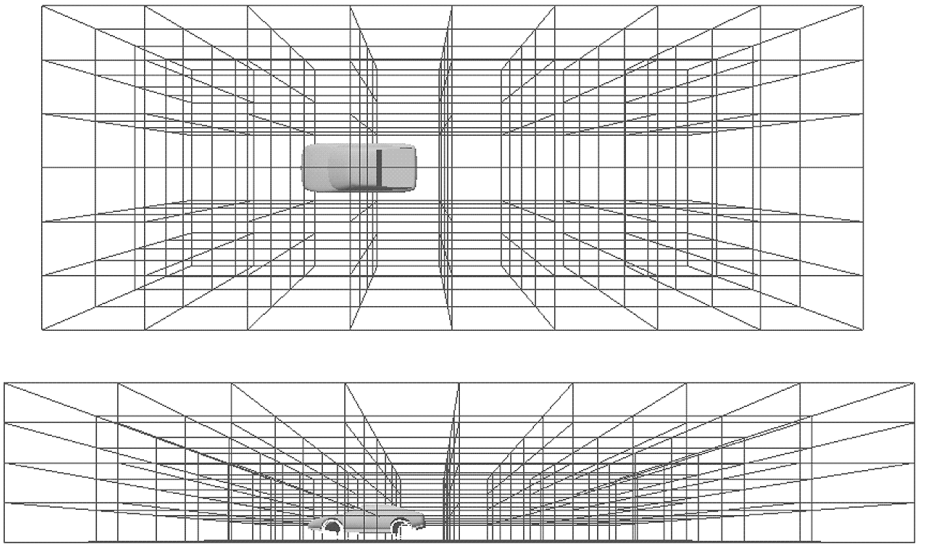


Fig. 7. Top and Front view of the finite difference grid volume for solving Navier-Stokes equations (one line per 5 grid points).

width, and the length is about 8-10 times of the vehicle's length. For example, we define an $81 \times 61 \times 41$ grid volume (Figure 7). The finite grid volume will move along with the car. The flow pattern formed by the movement of the car can then be treated as an inflow passing by the static car. The inflow is determined at the boundary of the grid system. We can specify that the inflow is always opposite to the normal direction of the front boundary and has the same magnitude at the car's speed.

To solve the Navier-Stokes equations (2) to (5), we use a time interval size according to the Courant-Friedrichs-Lewy (CFL) condition that the numerical integration is stable. In each time step, we use a three-step Alternating-Direction Implicit (ADI) method for the momentum equations and a Successive-Over-Relaxation (SOR) method for the solution of the pressure equation [Ghia et al. 1977]. The second-order accurate central differences are used for all spatial derivatives. When the velocity of the car stays constant, we use the same CFD results to calculate the dust particle behavior. If the velocity of the car changes, for example, an abrupt starting or braking, the fluid behavior is recalculated according to the current vehicle's velocity. This takes significant amount of time. Once the new fluid behavior is achieved according to the CFD computation, the dust particles are ejected from the ground and the simulation continues with the particle behavior calculated according to the new CFD results.

The number of the particles is proportional to the predefined ground dust density, size, and wetness. We approximate the standard deviation of turbulence velocities using Prandtl theory. The tires also eject particles. The maximum glue force of the tire is defined empirically and by random

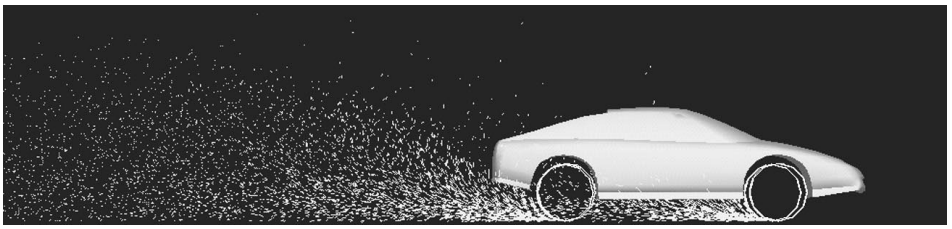


Fig. 8. A frame of dust simulation.

distribution. Some dust particles can be carried to more than a half cycle of the tire, and escape in the front. Figure 8 shows one frame of a dust behavior simulation. The solution to the Navier-Stokes equations takes approximately 7 minutes on a SGI O2 workstation with a MIPS R5000 processor and 64 Mbytes memory. By changing the parameters and boundary conditions, we can achieve different appearances.

4. SIMPLIFIED DUST PARTICLE SYSTEMS

The computation for the physically based dust behavior is time consuming. The CFD computation takes several minutes to complete. There are too many factors in the equations, and the forces on each individual particle have to be calculated during the whole simulation period. There are superfluous calculations because for a remote particle, some forces are reduced to near zero. Therefore, we divide a dust particle's behavior into three stages to simplify the simulation and to achieve real-time interactive simulation. We may consider that we have three different particle systems (models) working together to simulate the dust behavior. The three different systems are called Fluid Turbulence, Particle Momentum, and Airborne Drift.

4.1 Fluid Turbulence

Methods from computational fluid dynamics are used to calculate the fluid flow around the vehicle and turbulent wakes behind the vehicle. However, this approach is computationally complex and expensive, prohibiting simulation in real time. Our approach is to precalculate the laminar flow at any point in the flow volume off-line and save the results in the memory. $\mathbf{V}_{\mathbf{air}}$ is then calculated by Eq. (10) and the laminar portion of $\mathbf{V}_{\mathbf{air}}$ is read from the memory. This way, we avoid calculating CFD on the fly, which is impossible for solving 3D Navier-Stokes equations with much detail. With this method, we can achieve 2-3 frames per second for a single vehicle with about 3000 particles on a SGI O2 workstation with a MIPS R5000 processor and 64 Mbytes memory.

The laminar portion of $\mathbf{V}_{\mathbf{air}}$ is read from a 3D array of vectors in the memory directly; $\mathbf{V}_{\mathbf{wd}}$ is provided, which can be a function of time. Therefore, the drag force on the particle can be calculated immediately from Eq. (16) without CFD computation. In our implementation, we assume the

Reynolds number of the air, the cross section, and density of the particle are constant. So we further simply Eq. (15) as:

$$F_{drag} = cU^2 \tag{24}$$

where c is a precalculated constant:

$$c = k\rho_p A R_e^{-\frac{1}{2}}. \tag{25}$$

The problem with this method is that if the vehicle changes its velocity, the existing fluid data is no longer valid, and it is impossible to have all the solutions saved. The compromise solution is to precalculate several key frames of the velocity field with different vehicle velocities. Once there is a change in the vehicle's velocity, the closest frames are used to interpolate the current results. We are also working on generating another process that will calculate the new CFD results with the current vehicle's velocity. The new results will be integrated into the interpolation process.

4.2 Particle Momentum

The force F_{drag} acting on a particle was calculated in Eq. (16). Here we rewrite in vector form:

$$\mathbf{U} = (\mathbf{V_{air}} + \mathbf{V_{car}}) - \mathbf{V'_p} \tag{26}$$

where

$$\mathbf{V'_p} = \mathbf{V_{wd}} - \mathbf{V_p}. \tag{27}$$

As the vehicle travels and time passes, the location of the particle is farther away from the vehicle. Therefore, the effect of $(\mathbf{V_{air}} + \mathbf{V_{car}})$ on the particle is reduced and the air turbulence at the particle calms down. It takes a significant amount of time to calculate $\mathbf{V_{air}}$, only to find that $(\mathbf{V_{air}} + \mathbf{V_{car}})$ becomes a less important factor in the particle's driving force. Eventually, the air will return to a normal state, similar to the state before the vehicle generated the turbulence. When the effect of $\mathbf{V_{air}}$ and $\mathbf{V_{car}}$ is much reduced, that is, $\mathbf{V_{air}} + \mathbf{V_{car}} \approx \mathbf{0}$ we have

$$\mathbf{U} = -\mathbf{V'_p}. \tag{28}$$

As in Eq. (17), we have

$$\mathbf{F_{drag}} = \frac{\mathbf{U}}{U} F_{drag}. \tag{29}$$

From Eqs. (24), (28), and (29), we have:

$$\mathbf{F_{drag}} = -c\mathbf{V'_p}U \tag{30}$$

where c is a constant. At the moment that $(\mathbf{V_{air}} + \mathbf{V_{car}})$ vanishes, the particle enters what we call Particle Momentum stage. In this stage, the forces $\mathbf{F_{grv}}$ and $\mathbf{F_{drag}}$ are the primary forces governing its behavior. This stage will continue until the dust particle's velocity is reduced such that the particle's velocity is very close to the wind's velocity.

4.3 Airborne Drift

When the turbulence is reduced and the total force acting on a particle becomes very small, the dust particle begins to move with constant velocity towards the ground. The gravity force and the air drag force are balanced. At the same time, small low-density particles flow in the wind with very little resistance. Some low-density particles suspend in the air and drift in the wind for a long time. After a period of time, the remaining drifting particles are small in size (volume) and low in density. We further simplify and consider these particles' velocity to be the wind velocity with some random disturbances. If a dust particle touches the ground, the particle no longer moves, and is excluded from further processing. If a dust particle drifts from the range of the particle systems to the outside area, it is also considered extinct. It is faded away after a few frames of simulations.

Use of the Particle Momentum and Airborne Drift stages significantly simplifies the calculation of the new states of many particles. This gives us an improvement from 2 to 3 frames per second to 4 frames per second for a single vehicle with about 3000 particles on a SGI O2 workstation with a MIPS R5000 processor and 64 Mbytes memory.

5. RENDERING AND RESULTS

5.1 Motion Blur

We use motion blur to achieve better animation effects. To speed up the animation, instead of using accumulation buffers, we save several positions along a dust particle's movement. Each dust particle has a head pointer, which is the current position, and a tail pointer, which is the fading position. A particle is drawn a number of times into the buffer with increasing fading coefficients. The head is drawn at its current position with the particle's original color, and the tail is drawn at the earliest position with a much dimmer color. The particles in between are drawn with increasing fading coefficients.

5.2 Blending

We use Reeves's method to render the particle system: every particle is treated as a point light source when it is displayed. Each particle adds a bit of light to the pixels that it covers with distance attenuation. A particle behind another particle is not obscured but rather adds more light to the pixels covered.

In order to speed up the rendering process, we restrict our rendering area to be a box volume larger than the volume shown in Figure 7. As the

vehicle travels ahead, the box volume also moves ahead the same distance. The fluid dynamics calculations are restricted to this volume, as described at the last paragraph of Section 3.3. Any dust particle outside the box volume is considered extinct. It will be just rendered a few times to blend with the background and removed completely from the memory. This approach reduces the number of dust particles needed and thus reduces the memory needed to save all the particles. This method also allows us to have a background texture that is not updated outside the box area. As the vehicle moves, we cut a background corresponding to the box area, and the dust particles are rendered and blended with this piece of background. We only need to calculate and render the dust particles within this box area, and the rest of the background in the entire environment is untouched.

5.3 Particle Generation

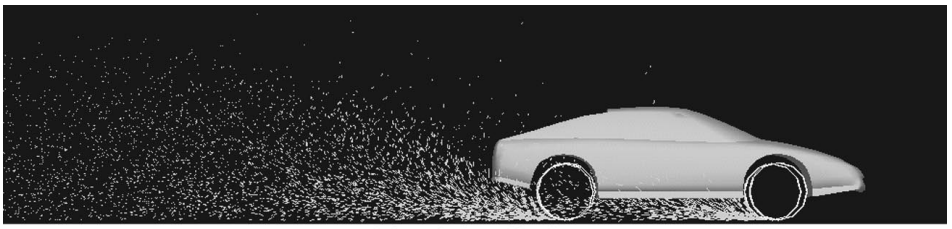
Theoretically, we have to decide where a particle leaves a tire and at what velocity. This process takes time. Because particle generation is not a deterministic process, we simplify the process by assuming that whenever a tire generates a particle, all three other tires generate a particle also with similar property. This way we don't have to recalculate particle generation for the other tires. For each generated particle, we produce four particles for the four tires with the same initial speed but different directions depending on the tires' current turning angles, with some random behavior. This way, we save time for calculating the generation of all particles.

5.4 Simple Rendering Results on a SGI O2 Machine

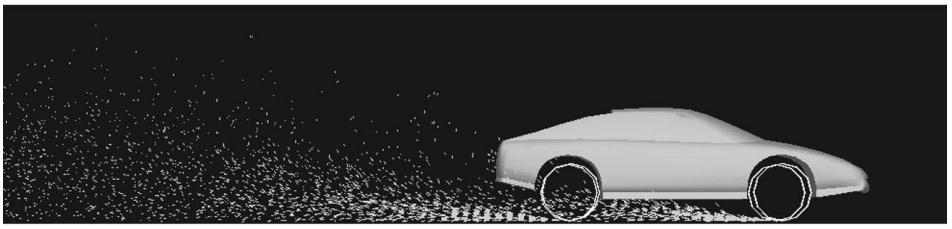
Figure 9 gives several simulation results under the same dust and air properties with different vehicle velocities (45 mph and 90 mph) and environmental wind. The number of particles rendered is about 3000/frame. Currently, the rendering is at 4-5 frames/sec with this number of particles. There are many other parameters that affect the simulation, such as the vehicle, the environment background, etc. The density of the dust displayed is greatly affected by the vehicle's velocity. We are still working on speeding up the rendering. In most cases, the simulation looks better with more dust particles, but the simulation is much slower because all the particles' new velocities and positions must be calculated.

5.5 Hardware-Assisted Rendering and Real-time Simulation on a SGI OnyxII Machine

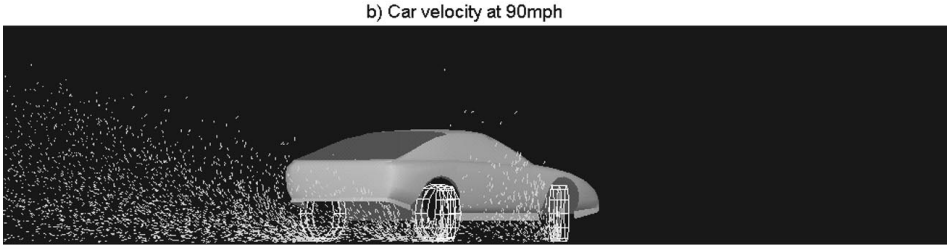
Some high-end computer systems, such as SGI OnyxII machines, are equipped with advanced OpenGL hardware features like texture mapping and alpha blending. We can take advantage of the hardware support, render the dust particles with complex textures and blendings, and achieve more realistic results and real time as well. Since the actual rendering algorithm depends largely on the detail of OpenGL API, here we only give a summary of the main ideas in the implementation. Please notice that while it is possible to implement our approach in pure software, hardware-



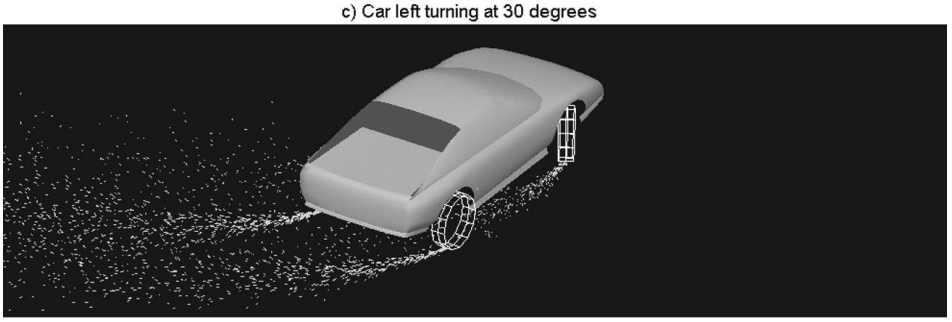
a) Car velocity at 45mph



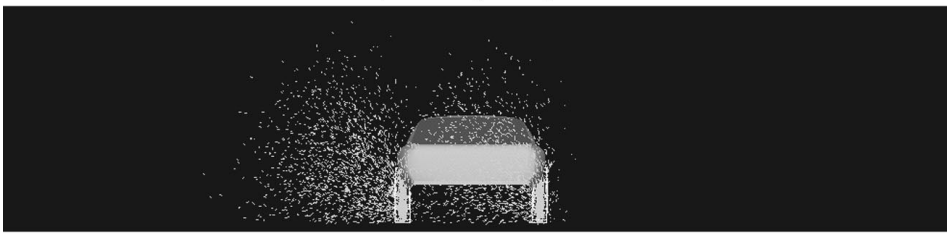
b) Car velocity at 90mph



c) Car left turning at 30 degrees



d) Car velocity at 45mph



e) Car velocity at 30mph and wind at 20mph in perpendicular direction

Fig. 9. Vehicles at different velocities; about 3000 particles are active; rendering at 4 frames/sec.

supported OpenGL provides important features that enable our methods to be computed in real time.

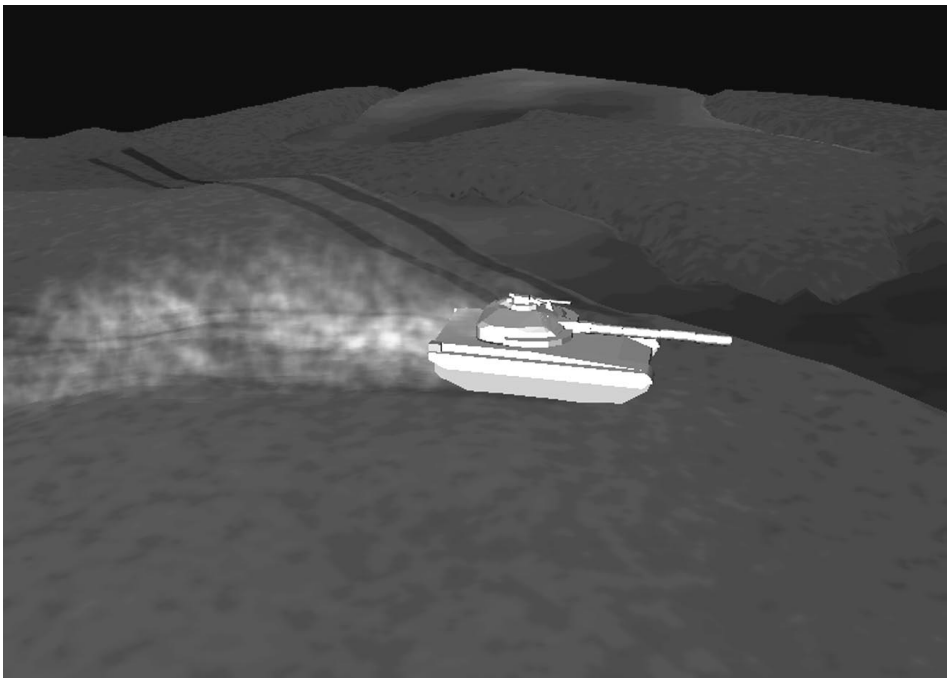


Fig. 10. Real-time simulation of dust on an OnyxII with 2000 particles.

In this implementation, each dust particle is rendered as a so-called bulletin-board object — a small circular area that is always facing the viewer. This small area is mapped with texture of a prerendered cloud-like fractal image containing an intensity-based alpha component. That is, the alpha value is the same as the intensity of the color in the image, so the dark area is very transparent with low alpha value. The cloud texture is generated by a standard procedural textural approach [Ebert et al. 1994]. By varying the parameters of the texture-generating procedure, we can generate a variety of slightly different textures that are used on different particles at different times. This improves the overall appearance and animation of the dust behavior. Therefore, in the rendering process, the particles are replaced by small pieces of textures and the successive overlapping pieces of textures are blended together under the control of the alpha channel.

Figure 10 shows one frame of the real-time simulation of dust behavior behind a tank rendered by the above approach. This simulation has 2000 particles running on a SGI OnyxII with InfinityReality graphics engine (four MIPS R10000 processors and 64MBytes of texture memory). We can achieve rendering and simulation at 4 frames per second.

Figure 11 shows one frame of the more realistic rendering with nearly 20,000 fine particles, which is about 10 times the number of particles for real-time simulation. It takes about 2 minutes to finish one frame of the rendering on the same machine. While our system, with particles at such a

Fig. 11. High-fidelity rendering with 20,000 particles.

number and scale, cannot be rendered in real time, the results are suitable for key-frame animation applications. We have produced several mpeg movies to demonstrate the results of key-frame animations at "http://graphics.gmu.edu/dust/".

6. CONCLUSION AND FUTURE WORK

We have introduced our approach to simulating dust behavior behind a traveling vehicle. We have two primary goals: one is realism of the simulation, the other is real-time computation. In order to achieve realism, we analyze the forces and factors and construct physically based empirical models to generate particles and control the dust behavior accordingly. In order to achieve real time, we further simplify the numerical calculations by precalculating the CFD fluid velocity field (volume), dividing dust behavior into three stages, and establishing simplified particle system models for each stage. We employ motion blur, particle blending, texture mapping, and other techniques in computer graphics to achieve better results. Our work is a useful addition to many applications in simulated virtual environments.

We plan to further consider the interaction between the dust particles, the vehicles, and the environment. The following research topics will be studied:

—integrate the dust behavior with grass and water on the road: when a vehicle passes by, the grass on the two sides of the road will swing back and forth and the water puddle will splash around. We also plan to simulate dust accumulation on the grass;

—integrate with dynamic terrain: dust will be generated while a bulldozer digs a ditch;

—integrate with Distributed Interactive Simulation: develop a practical model to simulate and synchronize the behavior in a networked virtual environment.

ACKNOWLEDGMENT

The authors are very grateful to the anonymous reviewers for their detailed comments on the methods, suggestions for improvements, and corrections on wording and errors. Their effort has shaped our work and enhanced our knowledge.

Thanks to Mr. Yonggao Yang for helping with the initial programming, and Dr. Jingfang Wang for helping with the CFD model.

REFERENCES

BALCI, O. 1997. Guest editorial—Simulation for training: foundations and techniques. *ACM Trans. Model. Comput. Simul. 7*, 3, 291–292.

BATCHELOR, G. K. 1967. *An Introduction to Fluid Dynamics*. Cambridge Univ., Cambridge, MA.

BURGGRAF, O. R. 1966. Analytical and numerical studies of the structure of steady separated flows. *J. Fluid Mech. 24*, 1 (Jan.), 113–151.

CHEN, J. X., LOBO, N. V., HUGHES, C. E., AND MO, J. 1997. Real-time fluid simulation in an networked virtual environment. *IEEE Comput. Graph. Appl. 17*, 3, 52–61.

CHEN, J. X. AND LOBO, N. V. 1995. Toward interactive-rate simulation of fluids with moving obstacles using Navier-Stokes equations. *CVGIP: Graph. Models Image Process. 57*, 2 (Mar.), 107–116.

COWHERD, C., GRELINGER, M., AND ENGLEHART, P. J. 1989. An apparatus and methodology for predicting the dustiness of materials. *American Industrial Hygiene Association Journal 50* (Mar.), 123–130.

CREMER, J., KEARNEY, J., AND PAPELIS, Y. 1995. HCSM: A framework for behavior and scenario control in virtual environments. *ACM Trans. Model. Comput. Simul. 5*, 3 (July 1995), 242–267.

DE REFFYE, P., EDELIN, C., FRANCON, J., JAEGER, M., AND PUECH, C. 1988. Plant models faithful to botanical structure and development. In *Proceedings of the 15th Annual Conference on Computer Graphics* (SIGGRAPH '88, Atlanta, Ga, Aug. 1-5), R. J. Beach, Ed. ACM Press, New York, NY, 151–158.

EBERT, D. S., MUSGRAVE, F. K., PEACHEY, D., PERLIN, K., AND WORLEY, S., Eds, 1994. *Texturing and modeling: A procedural approach*. Academic Press Prof., Inc., San Diego, CA.

GATSKI, T. B. 1996. Turbulent flows: Model Equations and Solution Methodology. In *Handbook of Computational Fluid Mechanics*, R. Peyret, Ed. Academic Press Prof., Inc., San Diego, CA.

GHIA, K. N., HANKEY, W. L. JR., AND HODGE, J. K. 1977. Study of incompressible Navier-Stokes equations in primitive variables using implicit numerical technique. American Isntitute of Aeronatutics and Astronautics (AIAA) Paper 77-648. AIAA, Reston, VA.

GHIA, K. N., HANKEY, W. L. JR., AND HODGE, J. K. 1979. Use of primitive variables in solution of incompressible Navier-Stokes equations. *AIAA J. 17*, 198–301.

HSU, S. AND WONG, T. 1995. Simulating dust accumulation. *IEEE Comput. Graph. Appl. 15*, 1 (Jan.), 18–22.

INSTITUTE FOR SIMULATION AND TRAINING, 1994. The DIS Vision: A Map to the Future of Distributed Simulation. University of Central Florida, Orlando, FL.

LI, X. AND MOSHELL, J. M. 1993. Modeling soil: Real-time dynamic models for soil slippage and manipulation. *SIGGRAPH Comput. Graph. 27*, 4 (Aug.), 361–368.

LOKE, T., TAN, D., AND SEAH, H. 1992. Rendering fireworks displays. *IEEE Comput. Graph. Appl. 12*, 3 (May), 33–43.

OPPENHEIMER, P. E 1986. Real time design and animation of fractal plants and trees. *SIGGRAPH Comput. Graph. 20*, 4 (Aug. 1986), 55–64.

PRANDTL, L. AND TIETJENS, O. G. 1934. *Applied Hydro- and Aeromechanics*. McGraw-Hill, Inc., New York, NY.

REEVES, W. T. 1983. Particle systems: A technique for modeling a class of fuzzy objects. *ACM Trans. Graph. 2*, 2 (Apr.), 91–108.

REYNOLDS, C. W. 1987. Flocks, herds and schools: A distributed behavioral model. *SIGGRAPH Comput. Graph. 21*, 4 (July 1987), 25–34.

ROTH, M. AND GURITZ, R. 1995. Visualization of volcanic ash clouds. *IEEE Comput. Graph. Appl. 15*, 3 (May), 34–39.

SCHIAVONE, G. A., SURESHCHANDRAN, S., AND HARDIS, K. C. 1997. Terrain database interoperability issues in training with distributed interactive simulation. *ACM Trans. Model. Comput. Simul. 7*, 3, 332–367.

SIMS, K. 1990. Particle animation and rendering using data parallel computation. *SIGGRAPH Comput. Graph. 24*, 4 (Aug. 1990), 405–413.

STAM, J. AND FIUME, E. 1993. Turbulent wind fields for gaseous phenomena. In *Proceedings of the ACM Conference on Computer Graphics* (SIGGRAPH '93, Anaheim, CA, Aug. 1–6), M. C. Whitton, Ed. ACM Press, New York, NY, 369–376.

TANNEHILL, J. C. AND PLETCHER, R. H. 1984. *Computational Fluid Mechanics and Heat Transfer*. Taylor & Francis/Hemisphere Publishing, Bristol, PA.

WEJCHERT, J. AND HAUMANN, D. 1991. Animation aerodynamics. *SIGGRAPH Comput. Graph. 25*, 4 (July 1991), 19–22.

WILLIAMS, R. R. AND DAVIS, R. E. 1988. *Battlefield Dust Environment Symposium III (Call No.: UG683 B39s 1988)*. U.S. Army Engineer Waterways Experiment Station, Vicksburg, Miss.

Received: August 1997; revised: April 1999; accepted: June 1999