

# Object Modeling and Animation with Smoke

Lin Shi

Yizhou Yu

Computer Science Department  
University of Illinois at Urbana-Champaign  
{yyz,linshi}@uiuc.edu, yizhouy@acm.org

## Abstract

This paper addresses the problem of controlling the density and dynamics of smoke so that the synthetic appearance of the smoke resembles a still or moving object. The target object is modeled as a volumetric density function. In order to match the target density with the smoke density, we propose an approach to design an imaginary force field including both long-range and short-range forces as well as a nonuniform biased diffusion equation driven by the target density. The long-range force field is induced by electric charges, and is responsible for long distance smoke transportation. The short-range force field is solved by an optimization procedure, and is responsible for local density adjustment. The synthetic force fields integrated with the diffusion equation can effectively achieve the desired results.

**Keywords:** Smoke Simulation, Controllable Animation, Euler Equations, Density Functions, Force Field Design, Optimization, Nonuniform Biased Diffusion

## 1 Introduction

Amorphous but elegantly moving matters, such as clouds, fog and smoke, usually give people plenty of space for imagination. We would be excited when a cloud in the sunset sky assumes the approximate shape of an animal or some other real object. It is an exhilarating event because it is rare for such amorphous matters to assume regular shapes. For the same reason, ghosts and deities are usually described to manifest themselves from smoke or clouds. Even a fairy tale has the following excerpt:

“... As Aladdin rubbed the lamp to try to get a better look, the lamp sprang magically to life! It was all Aladdin could do to hold onto the bucking lamp as it spewed colored smoke and magic sparkles! Like a volcano erupting, the lamp launched a long, blue stream upward. The blue smoke twisted and expanded as it rose toward the ceiling. Finally, it became an enormous, blue genie! ”

We would like to develop techniques for digitally reproducing these effects. Such techniques obviously have enormous applications in the entertainment industry. Our goal in this paper is to propose modeling and animation methods that produce physically plausible motion of smoke, which at the same time, assumes a regular shape which may be either static or moving, over a relatively long period of time. Needless to say, our goal is consistent with one of the general objectives of graphics research: the development of techniques that allow easy user-level control of the modeling and animation processes.

What is smoke in the first place? Smoke can be considered as a collection of light-scattering tiny particles floating in the air. The above problem is challenging since the smoke density in a fluid medium always tends to drift from a nonuniform distribution to a uniform one. In order to solve the proposed problem, we need to stop this process and meanwhile make the motion still resemble that of smoke.

## 1.1 Related Work

The modeling of smoke and other gaseous phenomena has received much attention from the computer graphics community over the last two decades. Early models focused on a particular phenomenon and animated the smoke's density directly without modeling its velocity [6; 11; 2]. Additional details was added using solid textures whose parameters were animated over time. Musgrave [10] proposed a procedural hypertexture [12] approach, to model dynamic fractal cloudlets which can be arranged to approximate regular shapes such as animals. Each cloudlet was animated separately. A common feature shared by these models is that they lack any dynamical feedback which is crucial to realistic animation.

A more natural way to model the motion of smoke is to simulate the equations of fluid dynamics directly. Recently, Foster and Metaxas [5] used relatively coarse grids to produce nice smoke motion in three-dimensions. Their simulations are only stable if the time step is small enough. To alleviate this problem and make the simulations more efficient, Stam introduced a model which is unconditionally stable and could be run at any speed [17]. This was achieved using a combination of a semi-Lagrangian advection scheme and implicit solvers. However, the simulations still suffered from too much numerical dissipation. Fedkiw *et. al.* [3] introduced vorticity confinement and a higher-order interpolation technique into visual simulation of smoke. As a result, the simulations can keep finer details on relatively coarse grids. Inviscid Euler equations instead of Navier-Stokes equations were used in their simulations.

Stam [16] applied a few types of vector fields to control the global trajectories of fluids. Foster and Metaxas [4] introduced a few embedded controllers that allow animators to specify and control a three dimensional fluid animation without knowledge of the underlying equations or the method used to solve them. They extensively used pressure controllers to achieve desirable results. However, the general objective of these attempts is not about making fluids look like regular still or moving objects. Witting [18] applied computational fluid dynamics to a traditional animation environment for theatrical effects.

Warping and morphing [7] are very useful tools for transforming images and objects. Warping does not have a clear target image or object while morphing usually produces transition between two objects. Sims [15] introduced a technique for successively warping images using vector fields. However, the results from the operation are not well controlled. Most morphing techniques focus on the smoothness of the planned transition instead of its physical plausibility.

## 1.2 Our Approach

To achieve realistic visual motion of smoke, we try to build our technique on those efficient approximate solution techniques [17; 3] for fluid mechanics. Since our goal is in general physically unrealizable, we only want to achieve physically plausible visual mo-

tion. A simple approach to solve our problem is to perform smoke simulation in the volume enclosed by the target object's boundary surface. However, there are a few problems with this hard boundary condition. Even though the boundary surface is not visualized during final rendering, it will still be obvious since the smoke stops or reflects back at the boundary. A smooth hard boundary for smoke is not a realistic phenomenon. The second problem is that the smoke will gradually fill up the volume and give rise to uniform density everywhere. Because of uniform density, the motion of smoke cannot be visually perceived any more.

In our proposed method, we define a volume density function for the target object and try to drive the smoke so that the density function of the smoke closely matches the target density function. The target density function is denoted as  $D(\mathbf{x}, t)$  where  $\mathbf{x}$  represents a point in a 2D or 3D space, and the extra variable  $t$  means the target density may vary at different times. Similarly, the density function of the simulated smoke is denoted as  $\rho(\mathbf{x}, t)$ . The problem becomes how to generate physically plausible  $\rho(\mathbf{x}, t)$  that approximately matches  $D(\mathbf{x}, t)$ .

Our solution to this problem makes use of carefully designed force fields and nonuniform biased diffusion, which will be elaborated later, to drive the smoke density into the desired form. The target density function does not have a hard boundary. It allows the smoke to move beyond the object, and possess a natural free-form boundary. Using volume density functions is also a more general approach to model objects than boundary representations.

When the target object has its own motion, the smoke should also follow its motion so that the object shape is approximated fairly well at each time step. Because of temporal coherence, the target object usually only moves a little from frame to frame. Therefore, the local gradient information of the evolving smoke density function can be used to keep pace with the object.

The contributions we have made in this paper include: i) methods, including an optimization procedure, to design desirable force fields, ii) a nonuniform biased diffusion process. As a result, the generated smoke simulations can resemble both the geometry and motion of regular 2D and 3D objects.

## 2 Smoke Equations

Since we consider smoke as a collection of particles immersed in a gas-like fluid such as the atmosphere, we need to simulate the motion of the gas first. The smoke particles largely follows the velocity of the gas while having a little bit diffusion going on. The diffusion part moves the smoke particles from higher density regions to lower density regions. We consider gases as inviscid, incompressible, constant density fluids whose velocity is governed by the incompressible Euler equations

$$\nabla \cdot \mathbf{u} = 0 \quad (1)$$

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

where Eq. (1) is for the conservation of mass and (2) is for the conservation of momentum. Since we are only concerned about the visual quality of smoke simulation instead of its numerical accuracy, the pressure gradient,  $\nabla p$ , is only used for the reinforcement of incompressibility. As long as there is some approach to obtain a divergenceless velocity field of the gas, this term can be dropped during simulation without obviously affecting the visual quality of the results. A Poisson equation for the pressure is solved in [3] to obtain a divergenceless velocity field.

The density of the smoke are assumed to be controlled by the following equation

$$\frac{\partial \rho}{\partial t} = -(\mathbf{u} \cdot \nabla) \rho + \nu_\rho \nabla^2 \rho + S_\rho \quad (3)$$

where the first term on the right hand side indicates the advection of this scalar field along the velocity field of the gas, the second term accounts for its own diffusion which is independent of the gas, and  $S_\rho$  is a source term with negative values for a sink. In this paper, we assume the temperature is a constant everywhere.

To solve the equations (1) and (2), we follow the basic steps outlined in [3]: first add advection and force to the velocity field; then project the velocity field by solving the pressure using the Poisson equation  $\nabla^2 p = \frac{1}{\Delta t} \nabla \cdot \mathbf{u}$  to enforce zero divergence. To transport smoke density, the three terms on the right hand side of Eq. (3) are addressed sequentially to produce an approximate solution: the advection term takes the velocity field of the gas at the current time step; the diffusion term then takes the density function produced by the advection term; sources are added at the end.

In order to control the density and motion of the smoke as we wish, there are three places in the above equations where we need to focus our attention: the external force field  $\mathbf{f}$  for the motion of the gas, the diffusion and source terms for the density of the smoke. The force field can influence the velocity field of the gas which in turn transports the smoke density to the desired locations, while diffusion directly adjusts the smoke density function. However, excessive amount of diffusion may result in unnatural smoke movement, therefore damage the visual quality of the simulation. The appropriate placement of smoke sources can alleviate the burden of smoke transportation. However, in our opinion, how many sources we need is an empirical question whose answer depends on the preference of human animators. In an extreme situation, there could be a source at every point, and the smoke density function can match the target density immediately. This is obviously not an interesting solution. The following sections address our automatic control schemes in terms of the first two aspects and leave the part on source placement to the user.

## 3 Force Field Design

It is extremely important for the velocity field of the gas to follow fluid equations in order to produce natural looking motion. However, there is no constraints on the external force field. The force field can change frequently and discontinuously without endangering the quality of the results. Basically, fluid equations absorb these discontinuities and integrate them into continuous velocity changes. The most important part of our technique is force field design which generates force fields that can effectively drag the smoke to approximate the desirable shapes. The designed force fields have two major components: a long-range force field which is responsible for moving the smoke to distant locations and raw adjustment, and a short-range force field which is responsible for more accurate density adjustment in local neighborhoods, especially at locations with a well-defined density gradient.

### 3.1 Long Range Forces

When we compare the target density function with the smoke density function, the smoke density may be higher than the target density at some regions while lower at the others. These regions may be quite distant from each other. Therefore, the excessive amount of smoke in some of the regions can not be redistributed to the desired locations in one time step. We adopt a heuristic approach to move the smoke step by step. At each step, external forces are imposed at the set of locations, denoted as  $L_e$ , with excessive amount of smoke. The direction and magnitude of these forces depend on the distribution of the set of locations, denoted as  $L_l$ , with too little smoke. This kind of long range effects can be effectively modeled using induced fields, such as electric fields and gravity fields. We choose to place a negative point charge at each location that belongs to  $L_l$ . Thus, each location in  $L_e$  receives an electric force towards each location in  $L_l$ . The accumulated force at each location in  $L_e$

can be formulated as

$$\mathbf{f}_i^L = \sum_j \frac{D_j - \rho_j}{r_{ij}^\alpha} \mathbf{r}_{ij} \quad (4)$$

where  $r_{ij} = \|\mathbf{r}_{ij}\|$ ;  $\alpha = 2$  for real electric fields, but can be adjusted to a different value. The inverse power function indicates that nearby charges have higher priority to receive smoke. If a singular point is undesirable, an exponential function with a negative exponent can be used. However, for a discrete simulation grid,  $r_{ij}$  would not approach zero.

### 3.2 Short Range Forces

The long-range forces are good at redistributing the amount of smoke across distant regions. For faster convergence rate, there is additional local information we can exploit. For example, suppose the smoke density is lower than the target density at a certain point. The local gradient direction of the smoke density function informs us that if we move the smoke along its negative gradient direction, the smoke density would be increased at next time step.

For previously mentioned reasons, we can drop the  $\nabla p$  term in Eq. (2) as long as we can still preserve incompressibility of the fluid. Thus, the velocity field for the next time step becomes

$$\mathbf{u}_{t+1} = \mathbf{u}_t - \Delta t (\mathbf{u}_t \cdot \nabla) \mathbf{u}_t + \mathbf{f}_{t+1} \Delta t \quad (5)$$

where  $\mathbf{u}_t$  and  $\mathbf{f}_t$  represent the velocity and force at time step  $t$ , respectively. Let us focus on the advection term and ignore the diffusion and source terms in Eq. (3) for now since we solve these terms sequentially. By looking one time step ahead and plugging Eq. (5) into Eq. (3), we obtain the actual amount of density change

$$\Delta \rho_t = \rho_{t+1} - \rho_t \quad (6)$$

$$= -\Delta t \nabla \rho_t \cdot (\mathbf{u}_t - \Delta t (\mathbf{u}_t \cdot \nabla) \mathbf{u}_t + \mathbf{f}_{t+1} \Delta t). \quad (7)$$

The apparent amount of density shift we would like to achieve is given by  $\Delta \rho_t^a = D(\mathbf{x}, t) - \rho(\mathbf{x}, t)$ . That means the designed force field should make  $\Delta \rho_t$  have the same sign as  $\rho_t^a$ ,

$$\Delta \rho_t = \lambda \Delta \rho_t^a \quad (8)$$

where  $\lambda$  should be a constant between 0 and 1 to make the simulation stable. This objective can be achieved by adding an incremental short-range force field,  $\mathbf{f}_{t+1}^S$ , on top of the long-range force field,  $\mathbf{f}_{t+1}^L$ , so that the combined force field,  $\mathbf{f}_{t+1} = \mathbf{f}_{t+1}^L + \mathbf{f}_{t+1}^S$ , satisfies Eq. (8). Thus, the short-range force field becomes the variable that needs to be solved.

If we further assume that  $\mathbf{f}_{t+1}$  must be collinear with  $\nabla \rho_t$ ,  $\mathbf{f}_{t+1}^S$  can be solved immediately at every point except those with  $\nabla \rho_t = 0$  using Eq. (8). Nevertheless, we need to consider the zero divergence requirement and other additional requirements such as the requirement that the force field should be spatially smooth. Considering all these factors, the short range force field at a time step can be solved by the following global optimization problem

$$\begin{aligned} \min_{\mathbf{f}_{t+1}^S} \quad & c_1 \sum_{\mathbf{x}} (\Delta \rho_t(\mathbf{x}) - \lambda \Delta \rho_t^a(\mathbf{x}))^2 + c_2 \sum_{\mathbf{x}} (DIV(\mathbf{x}))^2 \\ & - c_3 \sum_{\mathbf{x}} \left( \frac{\mathbf{f}_{t+1}(\mathbf{x})}{\|\mathbf{f}_{t+1}(\mathbf{x})\|} \cdot \frac{\nabla \rho_t(\mathbf{x})}{\|\nabla \rho_t(\mathbf{x})\|} \right)^2 \\ & - c_4 \sum_{\mathbf{x}} \sum_{\mathbf{y} \in N(\mathbf{x})} \cos(\theta_{t+1}(\mathbf{y}) - \theta_{t+1}(\mathbf{x})) \\ & + c_5 \sum_{\mathbf{x}} \sum_{\mathbf{y} \in N(\mathbf{x})} (\|\mathbf{f}_{t+1}^S(\mathbf{y})\| - \|\mathbf{f}_{t+1}^S(\mathbf{x})\|)^2 \quad (9) \end{aligned}$$

where  $c_1, c_2, c_3$  and  $c_4$  are positive weighting coefficients, and everything is expressed as a function of  $\mathbf{x}$  which represents a pixel for 2D images or a voxel for a 3D grid. The third term in Eq. (9) tries to keep the directions of the combined forces close to the gradient field of the smoke density.  $\theta_{t+1}(\mathbf{x})$  represents the orientation of the short range force at  $\mathbf{x}$ , and  $N(\mathbf{x})$  stands for a local neighborhood of  $\mathbf{x}$ .  $DIV(\mathbf{x})$  represents the divergence function of some direction field. Intuitively, it should be  $\nabla \cdot \mathbf{u}_{t+1}(\mathbf{x})$ . However, in practice, we find that better results can be achieved by applying the long range force field first followed by the projection step to enforce zero divergence, then applying a divergenceless short range force field. If this sequence of operation is followed,  $DIV(\mathbf{x})$  should be  $\nabla \cdot \mathbf{f}_{t+1}^S(\mathbf{x})$ .

### 3.3 Forces after Convergence

One of the most important features of smoke is that it moves all the time. For the same reason, to make a density function looks like smoke, we need to keep it alive. Once the smoke density function approximately matches the target density function, we still need to drive it with forces. Obviously, it is undesirable to move the smoke density away from the target density again. To maintain the close match between the two densities, at every point, we need to move the smoke along a direction with minimal derivative which happens to be a direction perpendicular to the local gradient of the target density. For a 2D target density function  $D(x, y)$ , the direction

$$\begin{pmatrix} D_y \\ -D_x \end{pmatrix}$$

is perpendicular to the gradient. For a 3D target density function  $D(x, y, z)$ , the direction

$$\begin{pmatrix} D_y - D_z \\ D_z - D_x \\ D_x - D_y \end{pmatrix}$$

is also perpendicular to the gradient. As long as these density functions are twice differentiable, it can be easily verified that the above mentioned direction fields are always divergenceless; therefore, they preserve mass and can be directly applied as force fields to the fluid to keep it alive.

## 4 Biased Diffusion

Force fields alone are not enough to enforce high-resolution details in the target density function although they can effectively constrain the smoke density in a local region. The velocities at nearby points are correlated because of the zero divergence requirement on the velocity field. To adjust smoke density at high-resolution details, one possibility is to run a diffusion process backwards. However, it is well known that running diffusion backwards is unstable and amplifies noise generating ripples.

For the sake of stability, we define a generalized diffusion process by using a nonuniform bias function as its equilibrium state, which means the generalized diffusion process stops when the smoke density function becomes exactly the same as the bias function. It is quite simple to achieve this generalized diffusion by applying conventional diffusion to the difference field between the smoke density and the bias function. In this paper, the target density serves as the bias function.

Obviously the more biased diffusion imposed the better convergence rate it will achieve. But the problem is that excessive amount of diffusion may result in unnatural motion since real smoke does not have much diffusion. Fortunately, in practice, the result still looks quite good with little amount of biased diffusion. To further improve the visual quality of controlled smoke simulation, we can also vary the diffusion coefficient from time to time. Thus, the

modified version of Eq. (3) becomes

$$\frac{\partial \rho}{\partial t} = -(\mathbf{u} \cdot \nabla) \rho + \nu_\rho(t) \nabla^2 (\rho - D) + S_\rho \quad (10)$$

where  $\nu(t)$  is the varying diffusion coefficient and  $D$  is the target density function.

## 5 Spatially Varying Control

It is not only hard, but also unnecessary to expect the smoke density matching the target density everywhere. The advection of smoke is indispensable for high visual quality which would be lost if severe restrictions on density were imposed. On the other hand, an object is usually characterized by a few features. We would be able to recognize the object when these features become observable. Therefore, it would be desirable to have spatially varying control of the smoke density. Throughout all the implementation, we are mostly interested in "feature points" where the magnitude of the gradient is relatively large. This is by no means the best feature detector. Other more sophisticated filters may also be applied [1].

Spatially varying control can be carried out during force field optimization as well as biased diffusion. The cost function for short-range force field can be modified to

$$\begin{aligned} & c_1 \sum_{\mathbf{x}} w(\mathbf{x}) (\Delta \rho_t(\mathbf{x}) - \lambda \Delta \rho_t^a(\mathbf{x}))^2 + c_2 \sum_{\mathbf{x}} (DIV(\mathbf{x}))^2 \\ & - c_3 \sum_{\mathbf{x}} w(\mathbf{x}) \left( \frac{\mathbf{f}_{t+1}(\mathbf{x})}{\|\mathbf{f}_{t+1}(\mathbf{x})\|} \cdot \frac{\nabla \rho_t(\mathbf{x})}{\|\nabla \rho_t(\mathbf{x})\|} \right)^2 \\ & - c_4 \sum_{\mathbf{x}} \sum_{\mathbf{y} \in N(\mathbf{x})} \cos(\theta_{t+1}(\mathbf{y}) - \theta_{t+1}(\mathbf{x})) \\ & + c_5 \sum_{\mathbf{x}} \sum_{\mathbf{y} \in N(\mathbf{x})} (\|\mathbf{f}_{t+1}^S(\mathbf{y})\| - \|\mathbf{f}_{t+1}^S(\mathbf{x})\|)^2 \end{aligned} \quad (11)$$

where an extra weighting function  $w(\mathbf{x})$  is inserted. The weighting function has larger values at feature points. Since there are multiple terms in the cost function to account for continuity, divergence as well as density matching. The short-range force field is in general overdetermined. By having spatially varying weights, we can let the density matching term play a major role in determining short-range forces at feature points while the forces at other places are determined mostly by the zero divergence and continuity terms.

The same principle can be applied during biased diffusion. That is, the diffusion process should be stronger and the convergence of density should be faster at feature points to make the target object recognizable. To achieve this, the diffusion coefficient needs to be spatially varying, too. This is similar to "anisotropic diffusion" introduced in [13]. The difference is that the diffusion at feature points is tuned to be weaker than usual in [13] while we set the diffusion coefficients close to zero besides at feature points to preserve the vivid dynamic appearance of smoke. Thus, Eq. (10) can be modified again to incorporate this kind of nonuniform diffusion.

$$\frac{\partial \rho}{\partial t} = -(\mathbf{u} \cdot \nabla) \rho + \nu_\rho(t) \nabla \cdot (c(\mathbf{x}, t) \nabla (\rho - D)) + S_\rho \quad (12)$$

It is proven in [13] that a differential equation of the form  $\frac{\partial I}{\partial t} = \nu(t) \nabla \cdot (c(\mathbf{x}, t) \nabla I)$  is stable provided that both  $\nu(t)$  and  $c(\mathbf{x}, t)$  are positively valued and differentiable. Therefore, our biased diffusion can also be made stable by defining  $I = \rho - D$  and  $\frac{\partial D}{\partial t} = 0$  during each iteration. Since we alternate advection and biased diffusion during each iteration, further investigation is needed to figure out the conditions under which this whole process remains stable for

an arbitrary period of time. In practice, we have not been able to observe any unstable situations although inappropriate parameters may cause failure in convergence.

## 6 Implementation

Both long-range and short-range force fields are computed on the fly at each time step. We use the conjugate gradient algorithm [14] to minimize the global cost function for solving short-range force fields. Nonuniform diffusion for the smoke density function is also performed at each time step.

Our basic implementation for solving the fluid equations follows the one introduced in [3]: dice the work space into discretized voxels, and define scalar quantities at the center of the voxels and vectors on the voxel surfaces. The velocity at any point inside a voxel can be obtained by linearly interpolating each component of the velocity vector separately. From the computed force fields, we obtain a new velocity field, and therefore new density distribution of the smoke. And the final step at each time step is to impose the diffusion operation. The implementation of scalar diffusion follows the method in [17].

We also implemented both open and closed boundary conditions for 2D and 3D grids. The closed boundary condition sets the velocity perpendicular to the boundary to be zero while the open boundary condition assumes the velocity fields inside and outside the grid are continuous at the boundary. One of the advantages of the open boundary condition is that the projection step would result in less damage to our designed force fields.

Advection is done using the semi-Lagrangian method, which needs to compute a trace matrix. Pixels are traced back to its original position in the current velocity field. And the new value of the quantity (scalar field, vector field) is assigned the value at the position traced to. But once it is out of the boundary, it is assigned a value consistent with the chosen boundary condition.

The projection step involves solving the Poisson equation, which results in a linear equation with a sparse matrix. Conjugate gradient method is the natural way of solving this problem. LAPACK [9] does a good job on this.

## 7 Results

We have successfully tested our method in a variety of examples, including static images, moving objects in video sequences, as well as synthetic 2D and 3D objects. We use a 200x200 grid for 2D simulations, and a 80x80x80 grid for 3D simulations. We have run most of the simulations on a Pentium 4 1.7GHz processor. The 2D examples take 5 seconds/frame including force field design, fluid simulation, and smoke advection and diffusion. The 3D examples take 90 seconds/frame.

In all experiments, the designed force field can effectively constrain the smoke and generate low-resolution structures of the target objects. The diffusion coefficient is set to be four orders of magnitude smaller than the long-range and short-range forces to make diffusion barely noticeable. The diffusion process is so weak that it alone cannot make the smoke converge on the target density distribution. However, this tiny amount of diffusion can help carve out the sharp features of the objects, and therefore, improve the visual quality of the results significantly. The locations of the smoke sources are interactively determined. Smoke sources are often placed close to the target object and move with it to have sufficient chances to converge. Usually there is only one source corresponding to each object.

### 7.1 Real Images and Video Sequences

For objects in real images and video sequences, their intensity distributions are considered as the target density distribution. The objects are segmented out from the background interactively along

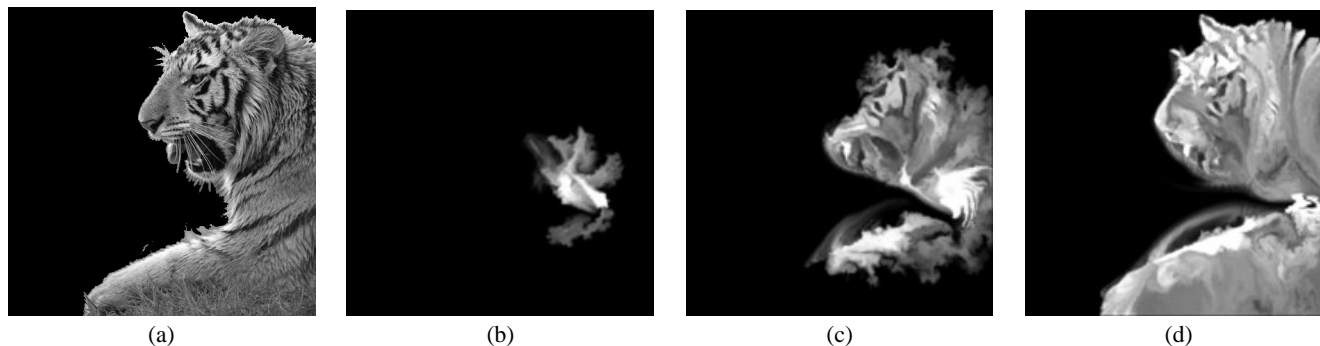


Figure 1: (a) A static image of a tiger; (b)-(d) intermediate images from a controlled smoke simulation with the image in (a) as the target density function.

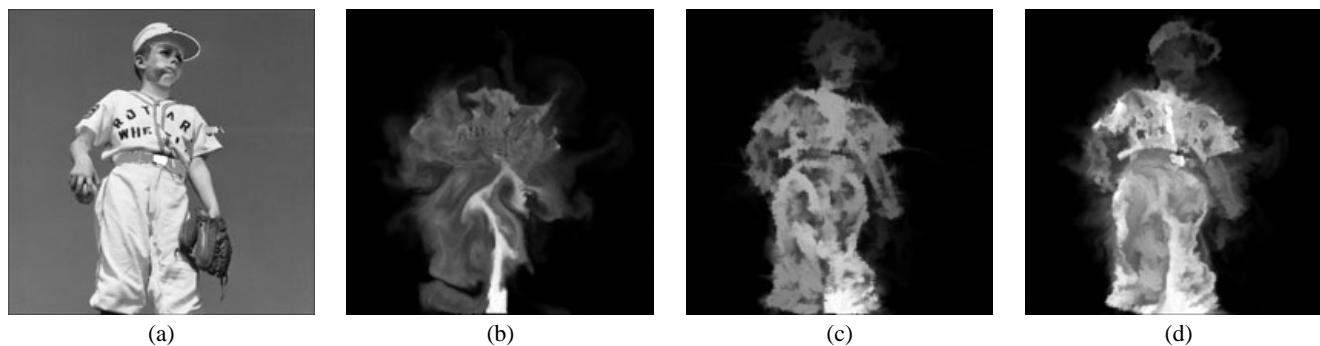


Figure 2: (a) A static image of a boy as the target density; (b) an image from a smoke simulation with diffusion only; (c) an image from a simulation with forces only but without diffusion; (d) an image from a simulation with all the forces as well as diffusion. The last image gives the best result.

with edge detection results. After simulation, the smoke density distribution is directly converted to the output image intensities. Fig. 1 shows the result for a static tiger image. The simulated smoke is kept alive all the time although the target object is static. Fig. 2 compares the results from three synthetic smoke sequences generated over a human figure in an image. The sequence with diffusion only fails to converge to the target density because the diffusion coefficient is set to be around  $10^{-4}$ . The sequence with forces only fails to produce fine details. The best result is produced when all the forces as well as the biased diffusion is applied. Fig. 3 shows the result for a running horse in a video segment. Note that the smoke not only approximates the shape, but also follows the motion of the horse because of the short-range forces.

## 7.2 Synthetic 2D and 3D Objects

Synthetic 2D or 3D objects are first discretized over the image plane or the voxel grid to produce the target density distribution. Smoother results can be obtained by antialiasing during discretization. All the rest of the pixels or voxels are assumed to have zero density. For 3D objects, the output images are rendered by ray-tracing the smoke volume density distribution [8]. Fig. 4 shows the successful result for a 3D static bunny model. Fig. 5 shows the result for a number of moving characters. Each of the characters has its own smoke source. The simulated smoke roughly keeps pace with the characters while moving around their neighborhoods to produce dynamic appearance.

The smoke animations for all these examples can be found in the accompanying video tape.

## 8 Conclusions

This paper presented a solution to the problem of controlling the density and dynamics of smoke so that the synthetic appearance of the smoke resembles a still or moving object. Specifically, we presented methods to design an imaginary force field consisting of various components as well as a nonuniform biased diffusion process driven by the target density. The synthetic force field integrated with the diffusion process can effectively achieve the desired results.

In future, we would like to enhance our techniques for image feature detection and dynamic feature tracking for moving objects in a video footage [1]. The target object may have very fine details in a local region while being largely featureless in some other regions. Using a uniform grid for smoke simulation is inefficient for these kind of objects. We would also like to investigate the possibility of adaptive multiresolution simulation.

## References

- [1] A. Bovik. *Image and Video Processing*. Academic Press, 2000.
- [2] D.S. Ebert, K. Musgrave, D. Peachy, K. Perlin, and S. Worley. *Texturing and Modeling: A Procedural Approach (second edition)*. AP Professional, 1998.
- [3] R. Fedkiw, J. Stam, and H.W. Jensen. Visual simulation of smoke. In *SIGGRAPH 01 Conference Proceedings*, pages 15–22, 2001.
- [4] N. Foster and D. Metaxas. Controlling fluid animation. In *Proceedings of Computer Graphics International*, pages 178–188, 1997.
- [5] N. Foster and D. Metaxas. Modeling the motion of a hot, turbulent gas. In *Proceedings of SIGGRAPH*, pages 181–188, 1997.
- [6] G.Y. Gardner. Visual simulation of clouds. In *Proc. of SIGGRAPH'85*, pages 297–304, 1985.
- [7] J. Gomes, T. Beier, B. Costa, L. Darsa, and L. Velho. Warping and morphing of graphical objects. SIGGRAPH 97 course notes.

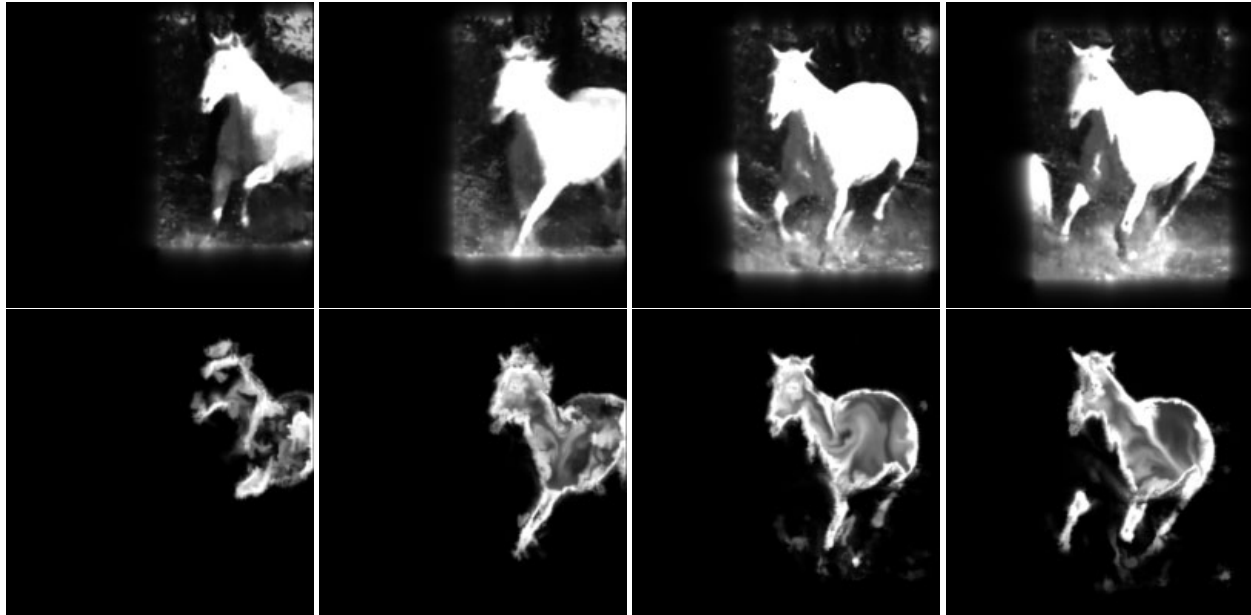


Figure 3: Top row: images from a real horse running sequence which serves as an evolving target density function; Bottom row: images from a controlled smoke simulation approximating both the shape and motion of the running horse.

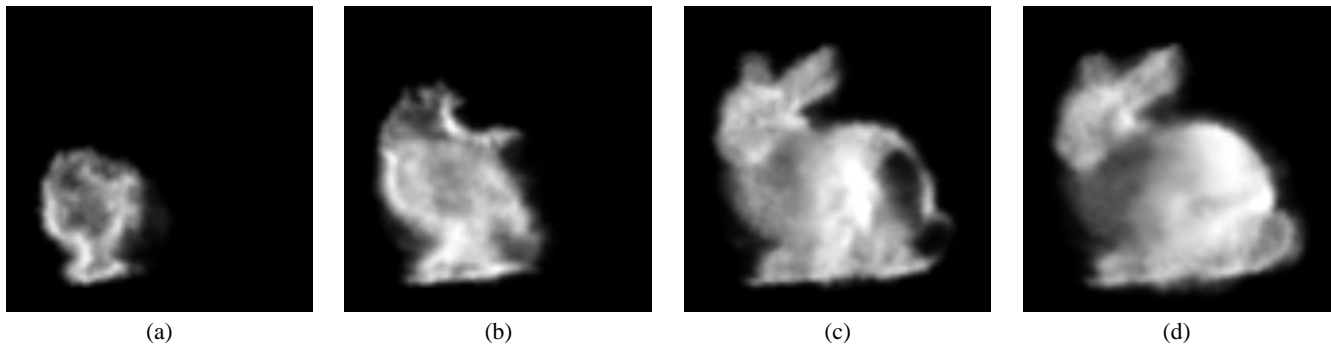


Figure 4: Intermediate images from a controlled smoke simulation whose target density function is a synthetic 3D bunny model.

- [8] J.T. Kajiya and B.P. von Herzen. Ray tracing volume densities. *Computer Graphics (SIGGRAPH 84 Proceedings)*, 18(3), 1984.
- [9] Lapack (linear algebra package). [www.netlib.org](http://www.netlib.org).
- [10] F.K. Musgrave. Great balls of fire. Online document, [www.wizardnet.com/musgrave/balls\\_of\\_fire.ps](http://www.wizardnet.com/musgrave/balls_of_fire.ps).
- [11] K. Perlin. An image synthesizer. In *Proc. of SIGGRAPH'85*, pages 287–296, 1985.
- [12] K. Perlin and E.M. Hoffert. Hypertexture. In *Proc. of SIGGRAPH'89*, pages 253–262, 1989.
- [13] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. Patt. Anal. Mach. Intell.*, 12(7):629–639, 1990.
- [14] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling. *Numerical Recipes in C*. Cambridge Univ. Press, New York, 1988.
- [15] K. Sims. Choreographed image flow. *Journal of Visualization and Computer Animation*, 3:31–43, 1992.
- [16] J. Stam. *Multi-Scale Stochastic Modeling of Complex Natural Phenomena*. PhD thesis, University of Toronto, 1995.
- [17] J. Stam. Stable fluids. In *SIGGRAPH 99 Conference Proceedings*, pages 121–128, 1999.
- [18] P. Witting. Computational fluid dynamics in a traditional animation environment. In *SIGGRAPH 99 Conference Proceedings*, pages 129–136, 1999.

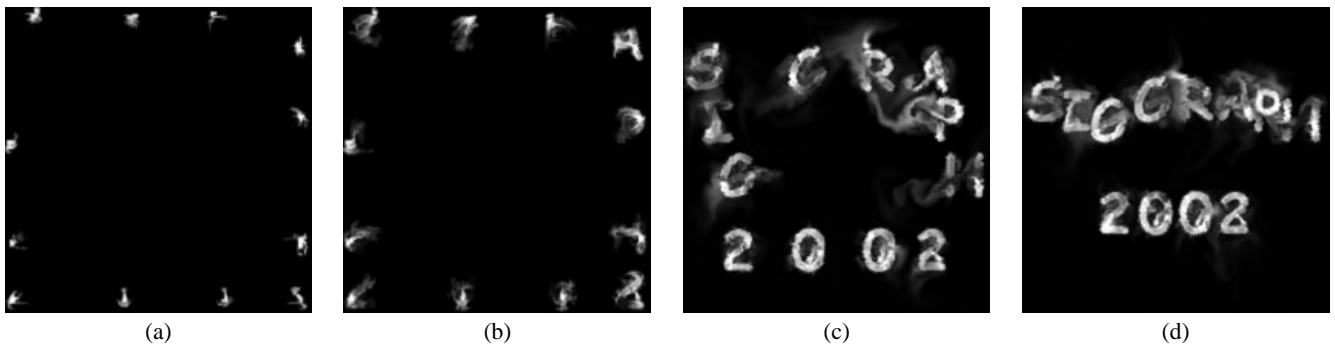


Figure 5: Intermediate images from a controlled smoke simulation whose target density function is a set of moving characters.