Eurographics/ ACM SIGGRAPH Symposium on Computer Animation (2006)
M.-P. Cani, J. O'Brien (Editors)

Path-based Control of Smoke Simulations

Yootai Kim[1] and Raghu Machiraju[1] and David Thompson[2]

[1]The Ohio State University, Columbus, OH, USA
[2]Mississippi State University, Mississippi State, MS, USA

Abstract

In this paper, we propose a novel path-based control method for generating realistic smoke animations. Our method allows an animator to specify a 3D curve for the smoke to follow. Path control is then achieved using a linear (closed) feedback loop to match the velocity field obtained from a 3D flow simulation with a target velocity field. The target velocity field can be generated in a variety of ways and may include the small scale swirling motion characteristic of turbulent flows. We provide several examples of complex smoke paths to demonstrate the efficacy of our approach.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism Animation

1. Introduction

Fluid phenomena such as flowing water and rising smoke are common occurrences that are observed in everyday life. Consequently, realistic depictions of such motions are desirable in modern animation. Remarkable visual realism has been achieved in the animation of smoke, flame, and water by exploiting techniques from Computational Fluid Dynamics (CFD) [FSJ01, FF01, NFJ02, MCG03, SRF05, PK05, AN05, FOK05, HK05]. It is well known and appreciated that the primary impediment to employing CFD simulation for animation is control of the resulting flow field.

Flow control falls into two classes: forward and inverse control. In forward control, a fluid flow is obtained via user specification of indirect quantities such as a force field or a pressure field (most notably [FM97]). To obtain the desired effects, the animator must change the parameters of the simulation in a costly trial-and-error process. In an inverse control approach, control parameters are determined to drive a fluid simulation to achieve the animator's goals. The initial and final shapes of the fluid mass serve as inputs to the simulation. Examples of such an approach are described in Treuille et al. [TMPS03] and Fattal and Lischinski [FL04].

In some instances, however, an animator wishes to make a mass of fluid to follow a specific path in a physically realistic manner. Consider the example shown in Figure 1. In this case, it is desired that the bulk motion of the smoke follow each letter while simultaneously exhibiting realistic, small-scale swirling motion similar to a turbulent flow. Existing techniques would be hard pressed to deliver a similarly controllable animation in an efficient manner. Moreover, the motion that is suggested by our animation captures the essence of the phrase that was etched in the skies over the Emerald City!

This paper presents an intuitive, path-based strategy to control simulated fluid motion that is described in Figure 2. An animator directly specifies a path for the bulk fluid motion using a NURBS curve. Instead of matching shapes, we match the velocity field from the simulation against a procedurally generated target velocity field defined in the region surrounding the specified path. Our control strategy produces a physically realistic flow by driving a simulated velocity field toward a specified target velocity field. The target velocity field includes various desired effects but may not be physically realistic. The simulated velocity field represents a compromise that is physically realistic, because it satisfies the equations that govern the motion of an incompressible fluid, and incorporates the characteristics of the target field. We demonstrate the generality of our approach by generating target velocity fields that include effects such as

© The Eurographics Association 2006.

delivered by EUROGRAPHICS DIGITAL LIBRARY www.eg.org diglib.eg.org

Figure 1: *Path-based smoke simulation of Elphaba's message in the feature film* The Wizard of Oz. *The left image shows the specified fluid paths. Each letter is formed individually. Note that the prescribed paths include regions of high curvature as well as self intersections. The right image shows the final frame of the simulated smoke sequence.*

a bulk motion tangent to the path, a bulk motion that swirls around the path, and small-scale swirling motion.

The salient contributions of this paper include: (i) a new method to produce a realistic looking flow of smoke, including small-scale turbulent motion, along a user-specified path, (ii) a novel way to integrate a procedurally defined velocity field with physics-based simulation through the use of closed-loop linear control, and (iii) an efficient algorithm that incorporates simultaneous solution of the path control problem with the flow simulation.

The paper is structured as follows. In Section 2, we review related work. We then explain our path-based control formulation in Section 3. Some details of our implementation are provided in Section 4. In Section 5, we include results that demonstrate the effectiveness of our approach. Finally, we draw conclusions and discuss limitations and future work.

2. Related Work

In this section, we provide a focused context for our work by considering only literature relevant to fluid simulation control. Foster and Metaxas [FM97] proposed a forward-control mechanism called an *embedded controller* to guide the flow synthesis. Their approach allows an animator to introduce various effects by altering the boundary conditions and fluid properties as well as the pressure and velocity fields. In Schpok et al. [SDE05], users can interactively control a simulation by manipulating a set of automatically extracted simulation features such as vortices and uniform advection. Inverse-control mechanisms can be categorized as one of two types: open-loop and closed-loop (or feedback) control. Treuille et al. [TMPS03] used an open-loop approach to control a smoke simulation by specifying the smoke density at various keyframes. The adjoint method was introduced by McNamara et al. [MTPS04] to improve the computational and storage efficiency of the open-loop control algorithm. However, complex flow synthesis is still intractable given the computational expense of the required optimization.

The work of Fattal and Lischinski [FL04] employed a form of closed-loop control. Their method used a non-linear feedback force to steer the controlled fluid density to the target density. Hong and Kim [HK04] proposed the use of a geometric potential field derived from a given target shape to generate a control force. In Shi and Yu [SY05a, SY05b], control forces were applied to drive the simulated fluid to resemble both still or moving objects while allowing the fluid to retain essential properties of the flow field. All of these methods were specifically designed for fluid shape control, and there is not much work reported for fluid path control in simulations. In an approach similar to ours, Gates [Gat94] used splines to describe flow paths and vortical motion to simulate small scale turbulence. Therein, fluid flows are produced along a spline curve by integrating a directed flow primitive which satisfies the continuity equation. Our control strategy, instead, develops a real flow through guiding a simulated velocity toward a target or reference velocity. In Foster and Fedkiw [FF01], artificial velocity constraints are enforced along space curves to control the fluid flow. It was, however, not possible to create complex motion paths and include both large and small scale effects. Recently, Angelidis et al. [ANSN06] provides a path control for a 3D smoke simulation based on vortex methods.

A linear feedback control-based method for velocity tracking in 2D incompressible flows was developed by Gunzburger and Manservisi [GM00]. Linear feedback control has several advantages for graphics applications. First, it is fast and easy to implement. Second, it can be applied to any

© The Eurographics Association 2006.

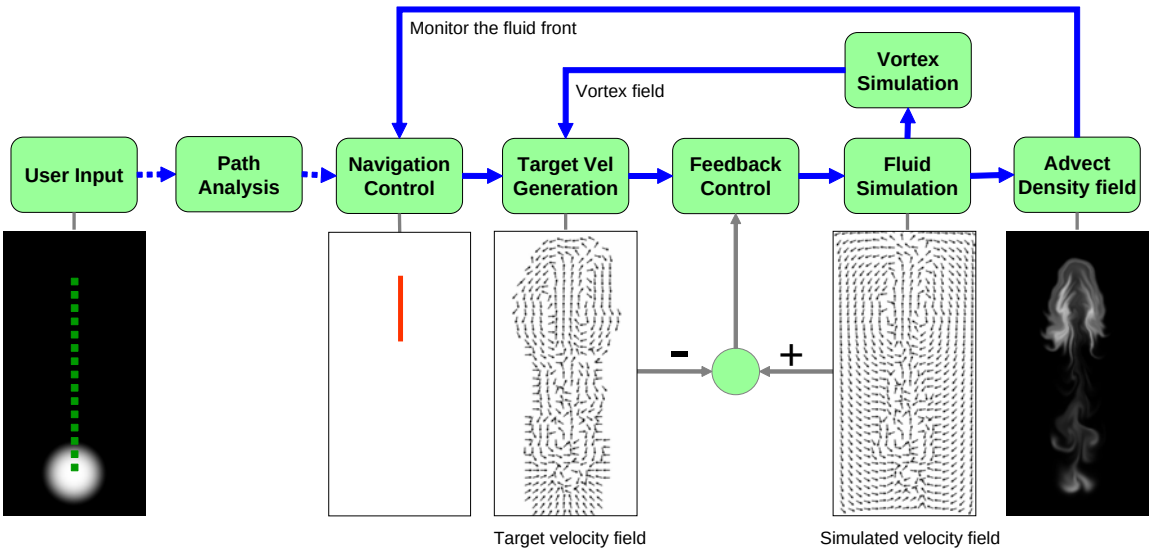


Figure 2: *Block diagram of our path-based smoke control algorithm: First, in a preprocessing stage, the user-specified smoke flow path (dotted green line in the left image) is segmented to avoid self intersections and regions with high curvature are identified. Then, the navigation control module determines the current path segment (solid orange line in the second image from the left) based on the location of the fluid front. A target velocity field is then generated for the domain using a local tangent field, a twisting Rankine field, a global normal field, and a velocity field from an unsteady vortex particle simulation to model swirling effects. Next, the feedback control loop employs a fluid simulation to match the motion of the fluid. Once the target field is matched, a density field is advected using the simulated velocity field to visualize the flow. The process is then repeated for the next time step.*

numerical scheme. Third, linear feedback control produces qualitatively good results for graphics application.

3. Path-based Control of Fluid Simulations

In this section, we describe our path-based smoke simulation strategy. We first present our method in algorithmic fashion. Next, we describe the components that are needed for the controlled synthesis of smoke animations. Our method requires the specification of a 3D flow path and a region surrounding the path, a target velocity field, and parameters for simulation and rendering. A 3D rectilinear grid of adequate resolution is also required for the flow simulation. The output from the simulation is a material density field that is rendered with suitable techniques (see Section 4).

3.1. Linear Feedback Control

The Navier-Stokes equations for an incompressible fluid with constant viscosity are given by:

$$\mathbf{u}_t + \mathbf{u}\cdot\nabla\mathbf{u} - \nu\nabla^2\mathbf{u} + \nabla p = \mathbf{f}_l, \quad (1)$$

$$\nabla\cdot\mathbf{u} = 0, \quad (2)$$

where $\mathbf{u}$ is the fluid velocity, p is the pressure, $\mathbf{f}_l$ is an applied control force, and ν is the kinematic viscosity. The subscript t indicates a time derivative, and ∇ and ∇^2 are the gradient and Laplace operators respectively. The constant fluid density is set to unity for simplicity. It should be noted that for the smoke simulation, we set ν to zero. Further, as discussed in Section 3.2.4, we employ a vortex particle method to produce swirling motion and thereby enhance the realism of the animation.

The resulting velocity matching problem requires the determination of a control force, $\mathbf{f}_l$ in Equation 1, that forces the solution $\mathbf{u}$ of the governing equations to be close to a specified target flow $\mathbf{U}$. The target flow, $\mathbf{U}$, can be a created in a variety of ways. We describe our methods in Section 3.2.

We now formalize the appropriate linear feedback control problem following Gunzburger and Manservisi [GM00]. The goal is to have a controlled solution $\mathbf{u}$ that satisfies the following conditions:

$$\lim_{t\to\infty} \|\mathbf{u}-\mathbf{U}\| \to 0 \quad (3)$$

$$\frac{d}{dt}\|\mathbf{u}-\mathbf{U}\|^2 \leq 0, \quad (4)$$

with the L_2 norm given by $\|\mathbf{u}\| = \left(\int_\Omega \|\mathbf{u}\|^2 d\mathbf{x}\right)^{1/2}$ with Ω being the spatial domain. Equation 3 enforces velocity

© The Eurographics Association 2006.

matching while Equation 4 states the necessary condition for stability of the controlled flow. If both conditions are satisfied, the actual flow velocity $\mathbf{u}$ is close to the target flow velocity $\mathbf{U}$ and remains so over time.

We realize closed-loop control for velocity matching by augmenting the governing flow equations, i.e., Equations 1 and 2, with a linear feedback control law:

$$\mathbf{f}_l = \mathbf{F}_b - \gamma(\mathbf{u} - \mathbf{U}), \tag{5}$$

where γ is a positive constant and $\mathbf{F}_b$ measures the body force generated by the target flow:

$$\mathbf{F}_b = \mathbf{U}_t + \mathbf{U} \cdot \nabla \mathbf{U} - \nu \nabla^2 \mathbf{U}. \tag{6}$$

Equation 5 uses the velocity difference $(\mathbf{u} - \mathbf{U})$ as a control error and strives to enforce the equality $\mathbf{u} = \mathbf{U}$ in the computational domain. Equation 6 is obtained by replacing $\mathbf{u}$ on the left side of Equation 1 with $\mathbf{U}$ and neglecting the pressure gradient term as no attempt is made to match a pressure field [GM00]. Intuitively, the first term on the right side of Equation 5 is the approximation to the force required to yield $\mathbf{u} = \mathbf{U}$ given by Equation 6. The second term can be interpreted as the stabilizing or damping force required to drive the controlled velocity field $\mathbf{u}$ toward the target velocity field $\mathbf{U}$. In contrast, other efforts [FL04, HK04] employ a control force that attempts to impose a target shape on the fluid.

Gunzburger and Manservisi further showed that the augmented flow equations, Equations 1, 2, and 5, have a solution for an admissible target velocity $\mathbf{U}$ for $\gamma > M$ where M is a positive constant proportional to the maximum speed of the target velocity field. $\mathbf{U}$ is a admissible target velocity if it is a divergence free vector field, i.e., $\nabla \cdot \mathbf{U} = 0$. Under these conditions, the error $\|\mathbf{u} - \mathbf{U}\|$ converges to zero exponentially with the passage of time. Although the error decreases to zero for all values of γ, a more rapid decrease is obtained for larger values of γ.

It should be noted that, in general, our target velocity field (described in Section 3.2.4) does not satisfy the divergence free condition. In essence, computing a nontrivial velocity field that satisfies the divergence free condition *and* matches a specific flow path is equivalent to the problem we are trying to solve. Therefore, we have elected to generate the target velocity field using a flexible approach that simplifies the specification of the path but may produce velocity fields with a nonzero divergence. Since the computed velocity field satisfies the divergence free condition, it is not possible to obtain an exact match between the computed field and the target field. It should be noted that our intent is not to prescribe the velocity field but to guide it. In practice, we have observed that our controlled velocity fields do converge and that they capture the characteristics of the specified target fields in a physically realistic manner.

3.2. Target Velocity Field Generation

We now describe a tool that allows an animator to create, in an intuitive fashion, a target velocity field that includes a bulk fluid motion along the specified path, a bulk swirling motion around the path, and small-scale swirling effects. In this section, we refer to Figure 3 which shows a depiction of the process that constructs a target velocity field $\mathbf{U}$ for the example shown in Figure 2.

3.2.1. Bulk Flow Path Specification

Our approach simplifies the specification of the bulk flow path by using a NURBS curve to define the path:

$$\mathbf{x} = \mathbf{C}(u),\ u \in [0,1], \tag{7}$$

where u is a parametric coordinate. The user specifies the width of the flow path w which then defines a region of influence, Ω_w, in the vicinity of the path segment $\mathbf{C}_s(u_s)$ as shown in the image on the far left in Figure 3. The region of influence is determined using the distance function $d(\mathbf{x})$ to the curve $\mathbf{C}_s(u_s)$ and is defined as:

$$\Omega_w = \{\mathbf{x} | d(\mathbf{x}) \leq R_w\}, \tag{8}$$

where the path radius is defined as $R_w = 0.5w$. For reference, we define a path segment as:

$$\mathbf{C}_s(u_s) = \mathbf{C}(u),\ u \in [u_b, u_e],\ u_b < u_e,\ u_b, u_e \in [0,1] \tag{9}$$

where u_b and u_e are the beginning and ending parametric coordinates of a path segment, respectively.

3.2.2. Path Analysis

As shown in Figure 2, path analysis is performed before the onset of the path control simulation. The generated target velocity field may not be sufficiently smooth around points of high curvature and may produce jerky fluid motion. We locate points of high curvature using the following test; if the radius of the osculating circle is smaller than a given threshold, this point is marked as one with high curvature. Then, the local maximum of a high curvature region is identified as a critical point. Each critical point now serves as end point of a segment so that no segment has a critical point in its interior as shown in Figure 4.

3.2.3. Navigation Control

We now describe how our approach can handle a self-intersecting flow path for which a unique target velocity field cannot be defined easily. For each time step, we determine the current curve segment and define a target velocity field for that segment. The segment size is predetermined by the user so that each segment has no self intersections. Hence, a curve segment can be defined as an interval $S = [u_1, u_2]$ with fields $lo[S] = u_1$ (the low endpoint) and $hi[S] = u_2$ (the high endpoint). The interval S satisfies the following two properties: $lo[S] < hi[S]$ and $lo[S], hi[S] \in [0,1]$. Then, a segment

© The Eurographics Association 2006.

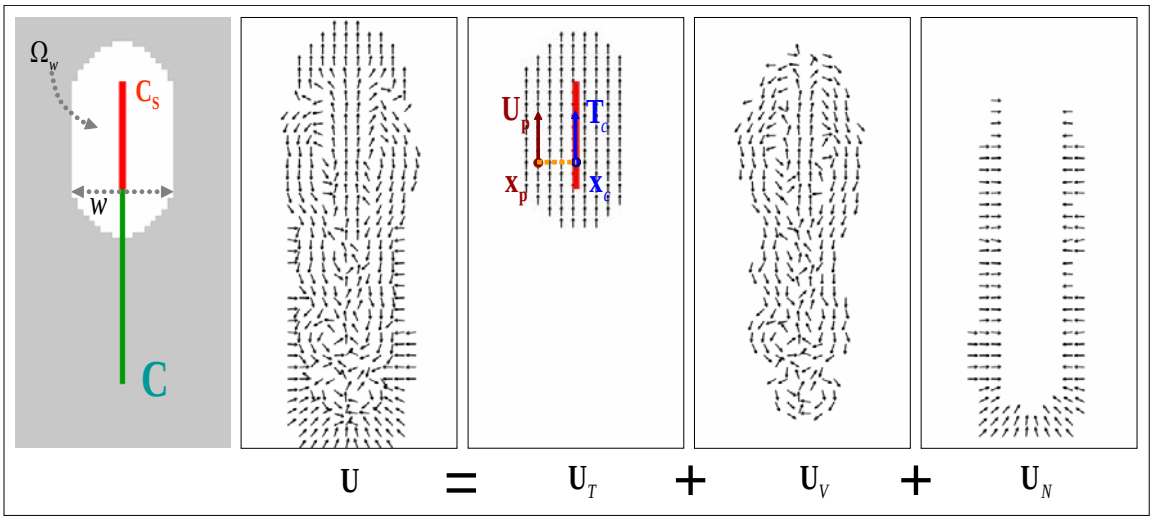


Figure 3: *Construction of target velocity field:* $\mathbf{C}(u)$ *is the input path and* $\mathbf{C}_s(u)$ *is a path segment,* Ω_w *is the region of influence, and w is the region width. The target velocity is comprised of a steady flow that is locally tangent to the path segment, an unsteady, vortex induced flow, and an unsteady normal flow to constrain the flow.*

sequence is defined as:

$$\begin{aligned} S_0 &= [0,\ \triangle u] \\ S_{i+1} &= \big[hi[S_i],\ hi[S_i]+\triangle u\big], \end{aligned} \tag{10}$$

where the subscript denotes a segment index and $\triangle u$ is a segment size. In addition, we make each path segment overlap the previous segment to make the transition smooth. Hence, we get the following recursion:

$$S_{i+1} = \big[hi[S_i]-\tau\triangle u,\ hi[S_i]+\triangle u\big], \tag{11}$$

where τ is a trailing factor. The trailing factor determines the length of the overlap between the current and previous curve segment. When a critical point exists in segment S_i, it is forced to be a new end-point of the segment S_i. Values in the range [0.005 - 0.05] and [1.5 - 3] are used for $\triangle u$ and τ, respectively.

We determine when to consider a new segment by tracking the fluid front. We monitor the material density of smoke around the neighborhood of the leading front of the current path segment S_i. We switch to the next path segment S_{i+1} when the material density becomes larger than a given threshold. Figure 4 shows a sequence of segments advancing past a critical point.

3.2.4. Target Velocity Field

The target field $\mathbf{U}$ is obtained by a superposition of four velocity components:

$$\mathbf{U} = g_t\mathbf{U}_T(\mathbf{x}) + g_r\mathbf{U}_R(\mathbf{x}) + g_n\mathbf{U}_N(\mathbf{x},t) + g_v\mathbf{U}_V(\mathbf{x},t), \tag{12}$$

where $\mathbf{U}_T(\mathbf{x})$ is a steady tangent field specified for the local path segment $\mathbf{C}_s(u_s)$, $\mathbf{U}_R(\mathbf{x})$ is a steady Rankine field, $\mathbf{U}_N(\mathbf{x},t)$ is an unsteady normal field that depends on the local material density at the boundary of Ω_w, $\mathbf{U}_V(\mathbf{x},t)$ is an unsteady discrete vortex field, and g_t, g_r, g_n, and g_v are gain parameters for the tangent, Rankine, normal, and vortex field respectively. Figure 3 illustrates the target field consisting of a tangent, vortex, and normal field in a 2D simulation. Note that the Rankine field can be depicted only in 3D space and therefore it is not included in Figure 3. The tangent field accounts for a bulk motion and forces the fluid to follow the specified path $\mathbf{C}(u)$. The Rankine field produces a bulk swirling motion around the specified flow path and the normal field constrains the fluid to the region of influence Ω_w. Finally, the vortex field imposes small-scale effects that model the swirling motions that are characteristic of smoke. Different target velocity fields can be created by changing the values of gain parameters. Other procedurally generated fields can be included to produce additional effects. We now describe each velocity component in detail.

Tangent Field: $\mathbf{U}_T(\mathbf{x})$ describes a steady, local velocity field that is defined in the region of influence Ω_w and is parallel to the path segment $\mathbf{C}_s(u_s)$. Since this component of the velocity field is steady, the path segment $\mathbf{C}_s(u_s)$ can be treated as a streamline. A streamline is a curve such that at every point on the curve the tangent is in the direction of the velocity of the flow. The velocity field $\mathbf{U}_T(\mathbf{x})$ for a curve $\mathbf{C}(u)$ is thus written as:

$$\mathbf{U}_T(\mathbf{x}) = M(\mathbf{x})\mathbf{T}(\mathbf{x}), \quad \mathbf{x}\in\Omega_w \tag{13}$$

where the speed function $M(\mathbf{x})$ specifies the magnitude of the velocity at position $\mathbf{x}$ and $\mathbf{T}(\mathbf{x})$ is the unit tangent vector derived from the NURBS curve $\mathbf{C}(u)$. The tangent

© The Eurographics Association 2006.

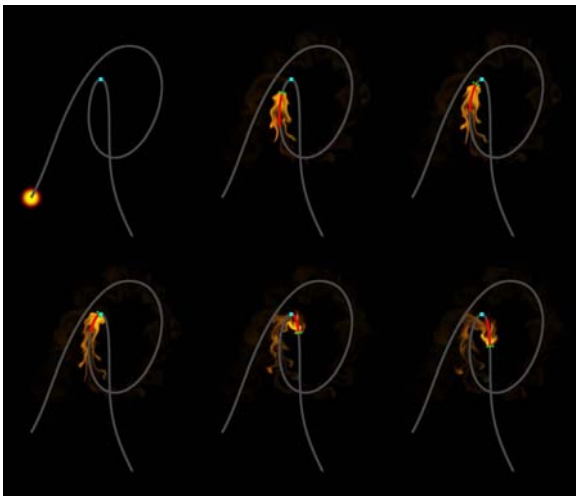


Figure 4: *Path analysis and navigation control: The cyan point in the top left image shows a critical point. The yellow ball shows the path width. The red curve is a path segment, and the green points at the front end of the segment are material density-monitoring points. We can see the segment advance along the path once the fluid front encounters the monitoring points. Renderings from frames 0, 114, 116, 119, 125, and 128 are depicted from top to bottom, left to right.*

vector at $\mathbf{x}_c$, which corresponds to the parametric position u_c, is given by:

$$\mathbf{T}(\mathbf{x}_c) = \left(\frac{d\mathbf{C}(u)/du}{\|d\mathbf{C}(u)/du\|} \right)_{u=u_c}. \tag{14}$$

The velocity vector at the point $\mathbf{x}_p$ in Ω_w can be computed by finding the point on the curve ($\mathbf{x}_c = \mathbf{C}(u_c)$) closest to $\mathbf{x}_p$ and using the tangent at $\mathbf{x}_c$. The following equation is used for the speed function:

$$M(\mathbf{x}) = H\left(1 - \frac{d(\mathbf{x})}{R_w}\right), \tag{15}$$

where $d(\mathbf{x})$ is a distance function to the path segment $\mathbf{C}_s(u_s)$, R_w is the path radius, and the smooth-step function H is:

$$H(s) = \begin{cases} -2s^3 + 3s^2 & \text{if} \quad 0 \le s \le 1 \\ 0 & \text{otherwise.} \end{cases} \tag{16}$$

.

Rankine Field: The Rankine velocity field is employed to produce a bulk rotational motion by treating the specified path as a vortex core and imposing the velocity field associated with a Rankine vortex [Lug83] in the plane whose normal is oriented along the path. Such motion is often observed in nature, in tornadoes and in drains, and can enhance visual realism. This is an example of the type of procedurally generated velocity fields that can be included in the target field. The mathematical definition of the Rankine vortex is:

$$M_\theta(r) = \begin{cases} r/R & \text{if} \quad r \le R \\ R/r & \text{if} \quad r > R, \end{cases} \tag{17}$$

where M_θ is a azimuthal velocity, and R is the radius of the vortex core. Hence, the Rankine field is given as:

$$\mathbf{U}_R(\mathbf{x}) = M_\theta\big(d(\mathbf{x})\big) \frac{\mathbf{V}_\theta(\mathbf{x})}{\|\mathbf{V}_\theta(\mathbf{x})\|}, \tag{18}$$

where $\mathbf{V}_\theta = \mathbf{U}_T(\mathbf{x}) \times (\mathbf{x} - \mathbf{x}_c)$, and $\mathbf{x}_c$ is the point on the curve closest to $\mathbf{x}$ (see Figure 3).

Normal Field: A normal velocity field $\mathbf{U}_N(\mathbf{x},t)$ is employed to constrain the smoke motion to occur inside the specified region of influence Ω_w. We employ the following potential function for this velocity field:

$$\mathbf{U}_N(\mathbf{x},t) = \rho(\mathbf{x},t) H\left(\frac{d_g(\mathbf{x})}{R} - 1\right) \frac{\mathbf{N}(\mathbf{x})}{\|\mathbf{N}(\mathbf{x})\|}, \tag{19}$$

where $d_g(\mathbf{x})$ is a global distance function to the input curve $\mathbf{C}(u)$, $\rho(\mathbf{x},t)$ is the material density (see Section 4), R is the normal field radius, H is a smooth-step function (Equation 16), and $\mathbf{N}(\mathbf{x}) = -\nabla d(\mathbf{x})$. This imposed velocity has the effect of pushing the fluid toward the path and only minimally affects the flow inside Ω_w.

Vortex Field: $\mathbf{U}_V(\mathbf{x},t)$ is added to the flow to induce an unsteady small-scale swirling motion that is an essential feature in the flow of smoke. If only a tangent field is used, the resulting animation looks placid and unrealistic. We employ a vortex particle method to generate a small-scale rotational field by solving the vorticity transport equation [CK00]:

$$\vec{\omega}_t + (\mathbf{u} \cdot \nabla)\vec{\omega} - (\vec{\omega} \cdot \nabla)\mathbf{u} - \nu\nabla^2\vec{\omega} = \nabla \times \mathbf{f}, \tag{20}$$

where we define the vorticity vector as $\vec{\omega} = \nabla \times \mathbf{u}$.
In the vortex particle method, the vorticity field $\vec{\omega}$ is approximated by a distribution of vortex particles with compact support. We denote the position occupied by a vortex particle by the subscript p. Let $\mathbf{x}_p = (x_p, y_p, z_p)$ be the position of a particle with a vorticity value $\vec{\omega}_p$. The analytical vorticity $\widetilde{\vec{\omega}_p}$ is defined as:

$$\widetilde{\vec{\omega}_p}(\mathbf{x}) = \vec{\omega}_p \phi(\mathbf{x} - \mathbf{x}_p) \tag{21}$$

where $\phi(x)$ is a linear kernel function (tent) with a kernel radius of R_k. To solve Equation 20 for the vorticity, the position of a vortex particle is updated using the particle trajectory equation:

$$\frac{d\mathbf{x}_p}{dt} = \mathbf{u}. \tag{22}$$

The vorticity of each particle is updated using the vorticity transport equation (Equation 20). Following the approach of Selle et al. [SRF05], the velocity field $\mathbf{u}$ required in Equation 20 is obtained from the solution of Equation 1. ν is set to zero and the external force term is ignored for this application. The vorticity value is clipped to unity so

© The Eurographics Association 2006.

that the vortex stretching term affects the flow only the direction of the vorticity vector. Finally, the vortex velocity field $\mathbf{U}_V(\mathbf{x},t)$ is obtained using:

$$\mathbf{U}_V(\mathbf{x},t) = \sum_p \mathbf{N}_p \times \widetilde{\omega}_p, \tag{23}$$

$$\mathbf{N}_p(\mathbf{x}) = \frac{(\mathbf{x}_p - \mathbf{x})}{\|\mathbf{x}_p - \mathbf{x}\|}. \tag{24}$$

4. Implementation

The methods described in the previous section facilitate the construction of a path-based fluid animation system that is easy to control and produces realistic-looking animations of smoke. However given the computational requirements, efficient and robust components are needed to realize the system. We now briefly describe some of the required components.

Distance Computation: A fast algorithm is required to compute the distance function needed to construct the target velocity field (Section 3.2). We use the fast sweeping algorithm in Zhao [Zha04] that combines upwind differencing with Gauss-Seidel iterations of alternating sweeping directions to solve the following Eikonal equation:

$$|\nabla d(\mathbf{x})| = 1, \qquad d(\mathbf{x}) = 0,\ \mathbf{x} \in \mathbf{C}. \tag{25}$$

Fluid Solver: A fast and stable fluid solver is essential to obtain the flow solutions. We chose the stable fluids method [Sta99]. This robust and efficient approach is widely used in fluid animation.

Visualization: Lastly, we need to produce a material density field in order to visualize the flow. A substance defined by a scalar field is carried and dispersed by the fluid velocity field $\mathbf{u}$ using the following advection-diffusion equation:

$$\rho_t = -\mathbf{u}\cdot\nabla\rho + \alpha\nabla^2\rho - \beta\rho + \kappa S, \tag{26}$$

where ρ is a material density field defined on our computational grid, S is a material source field, α is a diffusion constant, β is a dissipation constant, and κ is a source gain [Sta99]. The values employed for these variables depend on the desired fluid effects.

5. Results

Our simulation system has been implemented using C++ and OpenGL. We use the NURBS++ library [Lav02] to manipulate NURBS curves and the PBRT library [PH04] to render the 3D material density volume data. The computations were performed on a Windows PC with a Intel Xeon 2.0 GHZ processor and 1 GB of RAM. For most of our results, the damping coefficient employed in our control strategy, γ, is set to a value of 50 (Section 3.1). The tangent gain, g_t is set to 1, and The normal gain, g_n, and the rankine gain, g_r are set to zero unless noted otherwise (Section 3.2.4). As shown in Figure 7, vortex particles are populated to form a vortex ring along the path in every time step to induce turbulent motion. The vortex ring consists of 30 vortex particles randomly distributed at a constant radius within Ω_w. The vorticity vector for each particle is consistently oriented along the local tangent to the ring. In addition, a material density field is added around the path segment to enhance the visibility with a source gain $\kappa = 1$ (Section 4). Animations for our smoke flow synthesis examples are provided as supplementary video material.

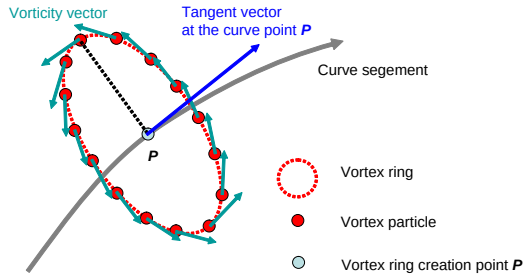
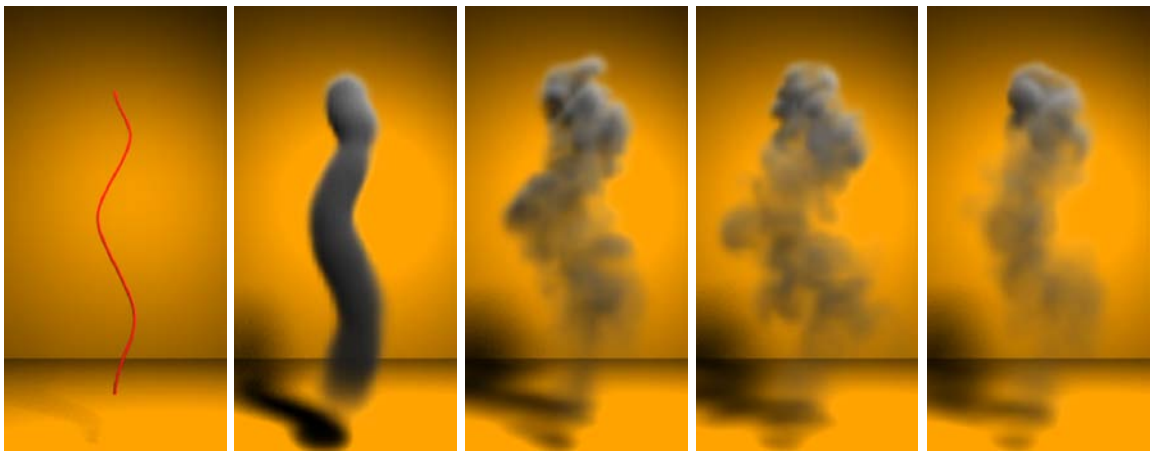


Figure 7: *Vortex ring generation: The consistently oriented vortex particles create a realistic "rolling" motion that is characteristic of smoke flow.*

Figure 5 illustrates the effect of each target field component. The leftmost image shows the input path. The rest of the images are the last frames of four different simulations to show the effect of a tangent, vortex, Rankine, and normal field, respectively. Figure 5(b) employs only a tangent field and the results are similar to those obtained by dragging a source through the field. A vortex field is added to the tangent field in Figure 5(c), and the result shows rich and realistic motions. Figure 5(d) uses a Rankine field, the tangent and vortex field. The inertia of the particles induced by the Rankine field causes some spreading that is not present in the vortex image. Finally, a normal field is introduced in Figure 5(e). There is reduced spreading in comparison to the vortex image and the Rankine image. These static images do not convey the range of motion present in the animations.

Figure 6 shows a 3D path control example for a knot-shaped curve. The first two rows show the effects of using different g_v values, and the third row shows a simulation using the Rankine field. The first row uses $g_v = 0.1$ and the second row $g_v = 0.2$. The smoke not only follows the path of the knot faithfully, but also exhibits realistic swirling motions. The second row shows more turbulent behavior than the first one. Note the increased dissipation of the smoke in the second row. In addition, it should be noted that shape-based methods may produce a better knot shape in the final frame; however, they would be hard-pressed to replicate the path-adhering sequence shown in this example. Additionally, many intermediate shapes must be prepared as

© The Eurographics Association 2006.



(a) input path (b) tangent field (c) vortex field (d) rankine field (e) normal field

Figure 5: *Effects of target velocity components: (b)* $g_t = 2.5$ *and* $g_v, g_r, g_n = 0$*, (c)* $g_t = 1$*,* $g_v = 0.05$*, and* $g_r, g_n = 0$*, (d)* $g_t = 1$*,* $g_v = 0.05$*,* $g_r = 1$*, and* $g_n = 0$*, (e)* $g_t = 1$*,* $g_v = 0.05$*,* $g_r = 1$*, and* $g_n = 100$

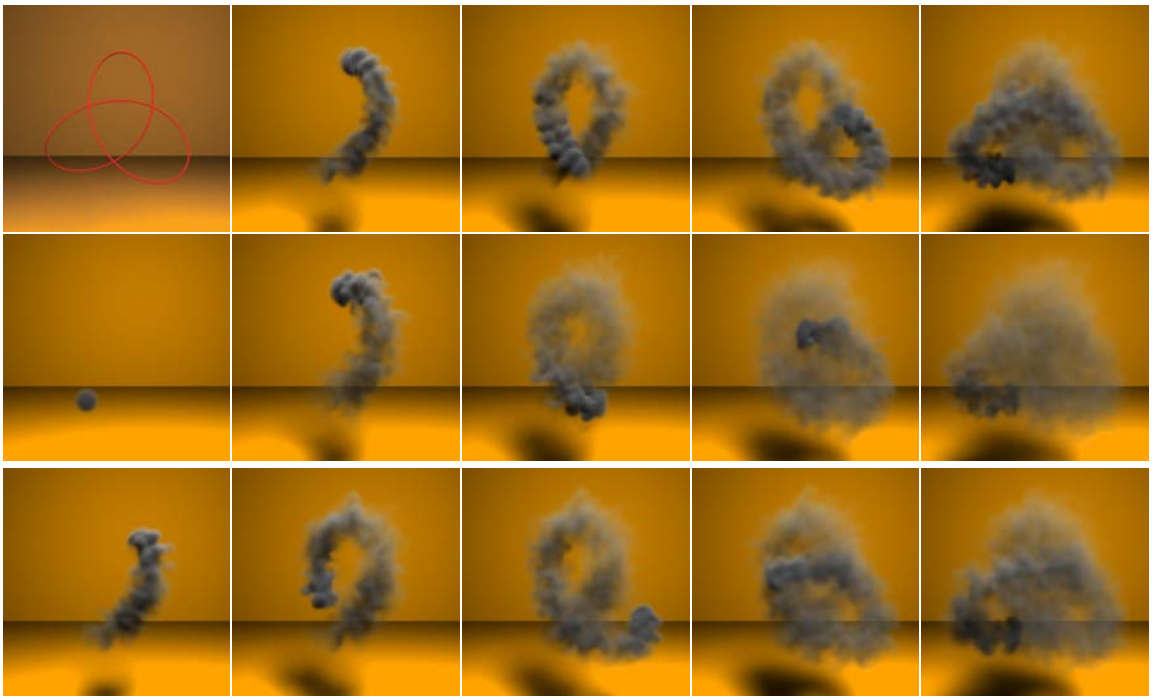


Figure 6: *Knot example: The flow produced appears realistic and illustrates the complexity of the flows that can be generated. Different vortex gain values are used in the first and second row. A Rankine field is applied in the third row. The leftmost images in the first and second rows are the input path and the initial smoke density, respectively.*

© The Eurographics Association 2006.

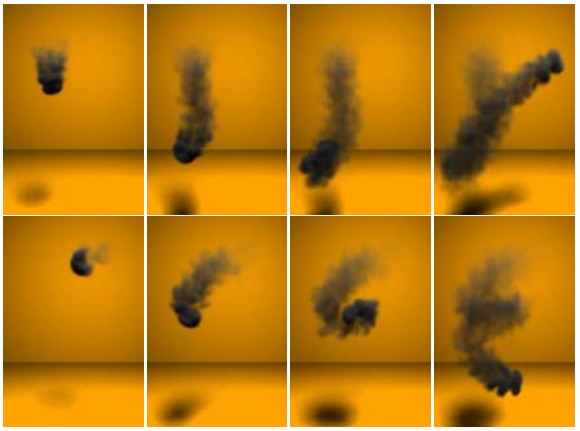


Figure 9: *Simulations of the letters "Y" and "E" in Figure 1*

keyframes or guiding objects. Moreover, the animation may not show natural path following motion. The third row shows the motion resulting from the application of the Rankine velocity field to the target velocity definition with a gain g_r of 1.0. The accompanying animation demonstrates how the target velocity field can be modified to include various effects and maintain physical realism. The simulation grid size is $70 \times 70 \times 35$, and the simulation speed is 0.6 fps on average.

The flow through the eyes of several needles, shown in Figure 8, is another example that demonstrates the power and utility of our path control. The specified path for the smoke is superimposed on the smoke animation to illustrate how well the smoke follows the path. It should be noted that these images may lead one to believe that a source density is being dragged through the specified path. However, none of the depicted effects can be realized unless all the components of the proposed framework are employed. The value of 0.05 is used for g_v. The computational grid size employed for this example is $105 \times 60 \times 105$, and the simulation speed is 0.1 fps. Figure 1 shows the reproduction of the scene from the feature film *The Wizard of Oz*. Each letter is separately simulated in a $60 \times 87 \times 30$ grid using a g_v value of 0.1. Later, the material density from each simulation is combined and rendered in one scene. Note that our method handles in a robust manner the motion of smoke around points of high curvature and/or corners in the letters, "T", "H", and "Y", and self-intersections in the letters "R" and "E". Figure 9 shows simulations for the letter "Y" and "E", respectively.

6. Conclusion

In this paper, we presented a new path-based control method for realistic smoke animations. The flow path is specified as a 3D NURBS curve. A target velocity field is generated to form streamlines along the path including small-scale swirling features. The controlled velocity field is adjusted to match the target velocity using the linear feedback control mechanism which is simple and fast. Finally, we demonstrated the efficacy of our methods through several instances of 3D flow synthesis for complex fluid paths.

Our method has several shortcomings. First, it is not possible to precisely control the mean speed of the flow because of the effects of the vortex field and the segment navigation. Second, the segment size must be determined manually. In practice, this is not a serious deficiency since it is not difficult to set the size unless the input curve is highly complex. Lastly, there is no mathematical proof for convergence of the linear feedback control for a non-admissible target velocity field.

For future work, we plan to improve the navigation control strategy to produce smoother transitions at points of high curvature and self-intersection. Another interesting problem is to include a static or moving object in the field while maintaining path control. Finally, the use of other procedural models to generate additional effects will be investigated.

Acknowledgments

We would like to thank the Advanced Computing Center for the Arts and Design (ACCAD) for allowing us to use their superb facilities.

References

[AN05] ANGELIDIS A., NEYRET F.: Simulation of smoke based on vortex filament primitives. In *SCA'05: Proc of the Symposium on Computer animation* (2005), pp. 87–96.

[ANSN06] ANGELIDIS A., NEYRET F., SINGH K., NOWROUZEZAHRAI D.: A controllable, fast and stable basis for vortex based smoke simulation. In *SCA'06: Proc of the Symposium on Computer animation* (2006).

[CK00] COTTET G.-H., KOUMOUTSAKOS P. D.: *Vortex Methods: Theory and Practice*. Cambridge University Press, 2000.

[FF01] FOSTER N., FEDKIW R.: Practical animation of liquids. In *ACM Trans. Graph. (Proc of SIGGRAPH'01)* (2001), pp. 23–30.

[FL04] FATTAL R., LISCHINSKI D.: Target-driven smoke animation. *ACM Trans. Graph. 23*, 3 (2004), 441–448.

[FM97] FOSTER N., METAXAS D.: Controlling fluid animation. In *Proc of CGI'97* (1997), pp. 178–188.

[FOK05] FELDMAN B. E., O'BRIEN J. F., KLINGNER B. M.: Animating gases with hybrid meshes. *ACM Trans. Graph. 24*, 3 (2005), 904–909. ACM Press.

[FSJ01] FEDKIW R., STAM J., JENSEN H. W.: Visual simulation of smoke. In *ACM Trans. Graph. (Proc of SIGGRAPH'01)* (2001), pp. 15–22.

© The Eurographics Association 2006.

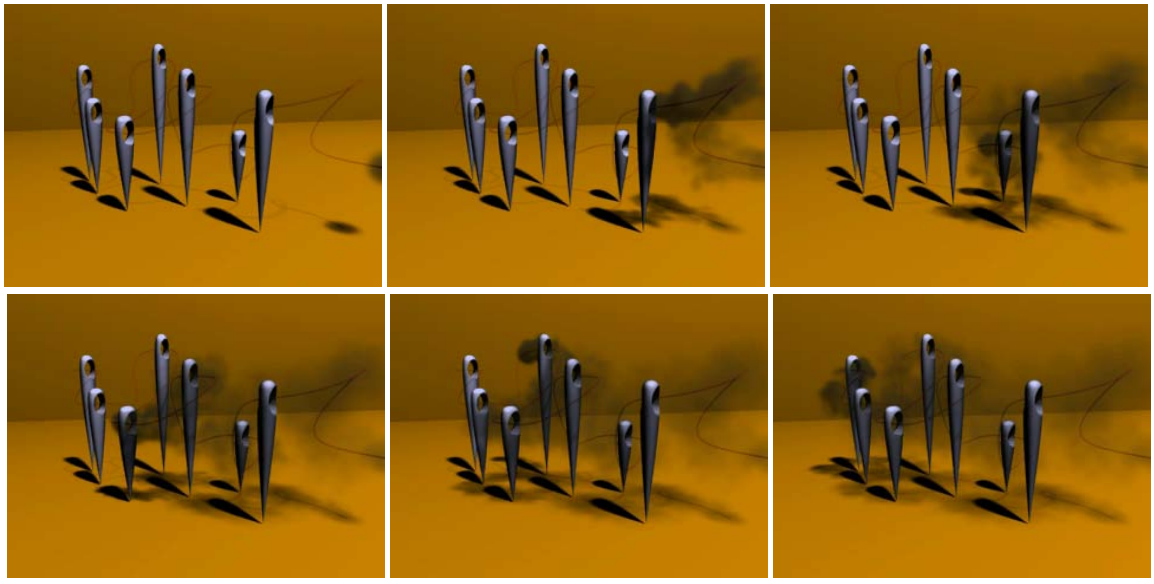


Figure 8: *Needle example: The path followed by the smoke illustrates that a complex smoke flow can be generated using our techniques. The smoke in this example does not interact with the needles.*

[Gat94] GATES W.: Interactive flow field modeling for the design and control of fluid motion in computer animation. *Master Thesis, UBC* (1994). http://www.cs.ubc.ca/labs/imager/th/pdf/Gates1994.pdf.

[GM00] GUNZBURGER M. D., MANSERVISI S.: Analysis and approximation for linear feedback control for tracking the velocity in navier-stokes flows. *Comput. Methods Appl. Mech. Eng. 189*, 3 (2000), 803–823.

[HK04] HONG J.-M., KIM C.-H.: Controlling fluid animation with geometric potential. *Computer Animation and Virtual Worlds 15*, 3-4 (2004), 147–157.

[HK05] HONG J.-M., KIM C.-H.: Discontinuous fluids. *ACM Trans. Graph. 24*, 3 (2005), 915–920.

[Lav02] LAVOIE P.: *The NURBS++ package*. 2002. http://libnurbs.sourceforge.net.

[Lug83] LUGT H. J.: *Vortex Flow in Nature and Technology*. John Willey and Sons, 1983.

[MCG03] MÜLLER M., CHARYPAR D., GROSS M.: Particle-based fluid simulation for interactive applications. In *SCA'03: Proc of the Symposium on Computer animation* (2003), pp. 154–159.

[MTPS04] MCNAMARA A., TREUILLE A., POPOVIĆ Z., STAM J.: Fluid control using the adjoint method. *ACM Trans. Graph. 23*, 3 (2004), 449–456.

[NFJ02] NGUYEN D. Q., FEDKIW R., JENSEN H. W.: Physically based modeling and animation of fire. In *ACM Trans. Graph. (Proc of SIGGRAPH'02)* (2002), pp. 721–728.

[PH04] PHARR M., HUMPHREYS G.: *Physically based rendering : from theory to implementation*. Morgan Kaufmann, Boston, 2004. http://pbrt.org.

[PK05] PARK S. I., KIM M. J.: Vortex fluid for gaseous phenomena. In *SCA'05: Proc of the Symposium on Computer animation* (2005), pp. 261–270.

[SDE05] SCHPOK J., DWYER W., EBERT D. S.: Modeling and animating gases with simulation features. In *SCA'05: Proc of the Symposium on Computer animation* (2005), pp. 97–105.

[SRF05] SELLE A., RASMUSSEN N., FEDKIW R.: A vortex particle method for smoke, water and explosions. *ACM Trans. Graph. 24*, 3 (2005), 910–914.

[Sta99] STAM J.: Stable fluids. In *ACM Trans. Graph. (Proc of SIGGRAPH'99)* (1999), pp. 121–128.

[SY05a] SHI L., YU Y.: Controllable smoke animation with guiding objects. *ACM Trans. Graph. 24*, 1 (2005), 140–164.

[SY05b] SHI L., YU Y.: Taming liquids for rapidly changing targets. In *SCA'05: Proc of the Symposium on Computer animation* (2005), pp. 229–236.

[TMPS03] TREUILLE A., MCNAMARA A., POPOVIĆ Z., STAM J.: Keyframe control of smoke simulations. *ACM Trans. Graph. 22*, 3 (2003), 716–723.

[Zha04] ZHAO H.: Fast sweeping method for eikonal equations. *Mathematics of Computation 74*, 3 (2004), 603–627.

© The Eurographics Association 2006.