COMPUTER GRAPHICS forum

DOI: 10.1111/j.1467-8659.2010.01719.x

COMPUTER GRAPHICS forum
Volume 00 (2010), number 0 pp. 1–11

A Smoke Visualization Model for Capturing Surface-Like Features

Jinho Park[1], Yeongho Seol[2], Frederic Cordier[3] and Junyong Noh[2]

[1]Department of Multimedia, Namseoul University, Korea
c2alpha@gmail.com
[2]Graduate School of Culture Technology, KAIST, Korea
{seolyeongho, junyongnoh}@kaist.ac.kr
[3]University of Haute-Alsace, France
fredcord@gmail.com

Abstract

Incense, candle smoke and cigarette smoke often exhibit smoke flows with a surface-like appearance. Although delving into well-known computational fluid dynamics may provide a solution to create such an appearance, we propose a much efficient alternative that combines a low-resolution fluid simulation with explicit geometry provided by NURBS surfaces. Among a wide spectrum of fluid simulation, our algorithm specifically tailors to reproduce the semi-transparent surface look and motion of the smoke. The main idea is that we follow the traces called streaklines created by the advected particles from a simulation and reconstruct NURBS surfaces passing through them. Then, we render the surfaces by applying an opacity map to each surface, where the opacity map is created by utilizing the smoke density and the characteristics of the surface contour. Augmenting the results from low-resolution simulations such a way requires a low computational cost and memory usage by design.

Keywords: smoke, fluid animation, surface, streakline, visualization

ACM CCS: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism-Animation; I.3.8 [Computer Graphics]: Applications

1. Introduction

The simulation and rendering of smoke are essential aspects in the creation of modern visual effects. Many recent movies have deployed various scenes containing computer-generated clouds, mists, fumes, and smokes. Painstaking reproduction of the motion and appearance of such fluid elements are of paramount importance as the realism should not be compromised when the CG and real video footage are composited together.

We attend to a crucial problem of visualizing small-scale smoke. Specifically, our work is motivated by the thin and plume-like appearance of the smoke shown in Figure 1(a). Such an appearance differs from the shape of a cotton bundle that results from explosions. The smoke exhibits thin semi-transparent surfaces and detailed plumes. This manifestation is often observed in incense, cigarette smoke and candle smoke. The plume-like phenomenon deserves careful attention for the successful reconstruction of its aesthetic motion and appearance. Visualization of such details of smoke is in great demand in applied fields such as the creation of commercial films (Figure 12).

A density field is a commonly used rendering medium for smoke animation. An Eulerian approach stores the smoke density in each grid cell. The density field is advected with the flow (The Navier–Stokes equations are a general model used to describe a fluid flow). The global illumination rendering scheme using a photon map by [FSJ01] produces results with a high level of realism. However, uniform grid-based approaches generally require a high-resolution domain for better simulation quality, which leads to large computational loads for rendering as well as simulation. The octree-based adaptive grid method by [LGF04] allows effective grid resolutions. Three refinement strategies were applied to handle

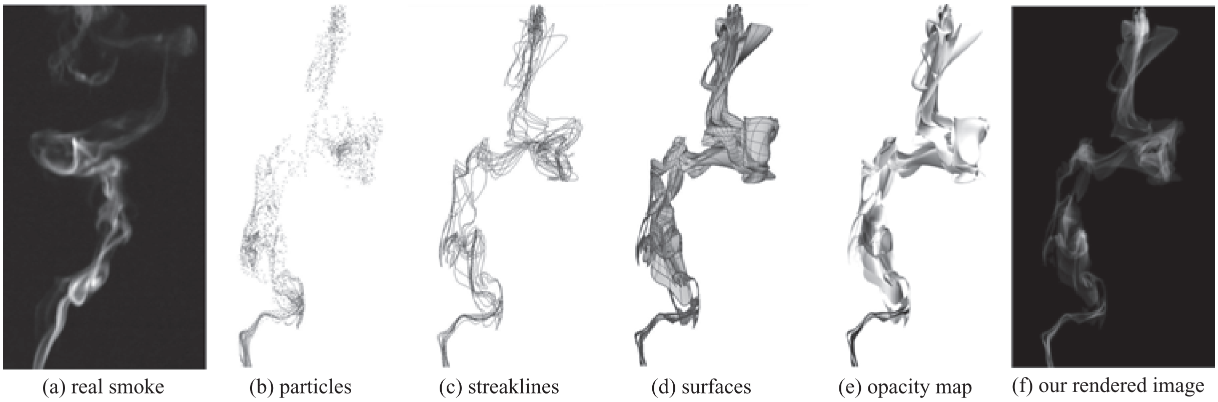
(a) real smoke (b) particles (c) streaklines (d) surfaces (e) opacity map (f) our rendered image

Figure 1: *The particles (b) emitted from a source are advected in the simulated flow to form the streaklines (c). The close streaklines are clustered together to create surfaces (d). Then, the smoke density field is utilized to generate an opacity map (e). The final rendered image (f) shows the surface-like features of smoke as in real smoke (a).*

smoke: refinement near objects, refinement near high vorticity and refinement in a band of density values ($0.1 < \rho < 0.3$ in their case). Although their method should be versatile enough to be applicable to any situations, in principle, it is unclear how well their strategies work for our goal of producing surface-like features in practice. Although density is intuitive and widely used medium for smoke rendering, its volumetric characteristics cause hazy appearance of smoke. It is therefore hard to represent the neat surface-like feature without tremendous computational loads.

Particles as a rendering medium are widely accepted by both researchers and designers. Many commercial software packages for visual effects such as [Hou] and [Rea] use Lagrangian methods in which particles serve the purpose of both simulation and rendering. Even in an Eulerian method, many techniques exploit massless particles passively advected with a time-varying velocity field [FOK05, FOA03, LF02]. The rendering primitive is a point or sphere whose centre coincides with the position of each particle. The rendering issues addressed in [AN05] are similar to those that motivated our work. Their ellipsoid smoke particle representation scheme reflects the surface-like characteristics of actual thin smoke. However, the quality of the results does not sufficiently describe smoke motion and appearance in detail. Although the number of required particles is not specified in [FOK05], it is often necessary to use them in large amounts for visually pleasing results. Although particle-based smoke rendering approaches achieve visual realism in general, disturbing smoke grains often seem inevitable. At least in theory, an infinite number of particles are required to completely fill a surface.

Our goal is to propose a smoke synthesis method to reproduce its thin surface-like features efficiently. We believe that our method can be very useful in the field of visual effects or commercial films where visual correctness is more appreciated than physical accuracy. Conventional approaches are not well-suited for small-scale smoke animation, as there are no parameters that directly relate to the *surfaceness* of the simulation results. A very efficient visualization model is proposed here that is specifically designed to create the semi-transparent surface look of the smoke. Briefly, the procedure of our method is described here. Initially, massless particles are emitted from user-specified source positions. Unlike common particle-based rendering approaches, the number of emitted particles is typically very small, for example, 20 per frame in the example shown in Figure 1(b). The main idea involves following traces known as streaklines that are created by the advected particles and reconstructing NURBS surfaces passing through them. Then, we render the surfaces by applying an opacity map to each surface, where the opacity map is created by utilizing the smoke density and the characteristics of the surface contour. The density value interpolated from the positions on the surfaces plays the role of the opacity value at the position.

Our contributions include the following distinctive features: The characteristics of the small-scale smoke are modelled in particular. Our model efficiently reflects the surface-like feature of smoke without relying on a high grid resolution, a large number of particles, or new sophisticated advection algorithms, which may be more appropriate for large-scale phenomena such as an explosion. A simple but very resourceful smoke visualization method is presented. This method achieves visually pleasing results without incurring a high computational cost and memory consumption. We propose to employ a novel surface reconstruction technique using NURBS. We believe that utilization of NURBS surfaces can provide an efficient solution to some of the fluid phenomena that may require complex solutions from the CFD point of view.

2. Related Work

2.1. Smoke animation

For the last decade, there have been great efforts in the production of realistic smoke animation to fulfil the needs of visual effects. Since the landmark research work [Sta99] based on the computational fluid dynamics (CFD) was introduced to the graphics community, improving the numerical accuracy and the computational efficiency has been important topics in fluid simulation [FSJ01, LGF04, FOK05, MCPN08, KTJG08]. Although these efforts attempt to improve the quality of simulation by tackling the CFD *per se*, various control techniques have been presented to generate smoke animations corresponding to user inputs. These include target-driven forces [FL04], a user-specified path [KMT06] and feature extraction [SDE05]. Many research efforts have been made to synthesize gaseous phenomena similar to smoke, explosions [FOA03], fire [NFJ02, LF02] and clouds [DKY*00].

2.2. Smoke rendering

Our work is closely related to the visualization of unsteady flows. Recent research [Max05] offers coverage of the visualization field that is more comprehensive in that it reports progress made in the visualization of molecules, scalar fields and vector fields.

2.2.1. *Volume rendering*

Based on bidirectional Monte Carlo tracing and photon maps, Jensen and Christensen [JC98] presented an efficient global illumination method for scenes with participating media. In [Sta99], the gases are volume rendered using a 3D hardware texture map. Fedkiw *et al.* [FSJ01] implemented both the hardware-based renderer by [Sta99] and the global illumination renderer by [JC98]. Dobashi *et al.* [DKY*00] renders clouds images in a common hardware-accelerated rendering method using OpenGL. In [NFJ02], the hot gaseous products, smoke and soot are rendered using a blackbody radiation model. In [SDE05], the resulting simulations are rendered interactively by the hardware-accelerated implementation of advected textures. Staubli *et al.* [SSP*05] presented a real-time volume rendering method for technical (unrealistic) visualization of transient smoke propagation. For real-time performance, Zhou *et al.* [ZHG*07] proposed a method to achieve the shadowing and scattering effects of inhomogeneous media.

2.2.2. *Particle rendering*

Feldman *et al.* [FOA03] renders the flame arising from an explosion by drawing the fuel and soot particles directly. In [BH05], smoke particles are rendered directly to an accumulation buffer accounting for camera, fluid motion and particle ages. Feldman *et al.* [FOK05] renders massless particles passively advected in the flow. In a structural flame model for a production environment by [LF02], a B-Spline curve represents the spine of a flame where the particles are sampled from the curve evolved with a wind field.

2.2.3. *Flow visualization*

Becker *et al.* [BML95] presents a visualization method based on particle tracking for describing time-varying flows. Texture-based flow visualization methods compute a texture that is used to generate a dense representation of the flow. It uses the filtering of texture values according to the underlying flow [LHD*04]. In the method known as image-based flow visualization (IBFV) originally proposed by [vW04], flow is visualized as moving textures with line integral convolution and spot noise. Telea and Wijk [TvW03] extended IBFV to 3D steady flow visualization problems. Geometric flow visualization methods utilize geometric objects such as streamlines, streaklines and pathlines [PVH*02]. In [WTS*07], generalized streaklines visualize the particles that are ejected from the wall. Although the method in [Hul92] depicts flows using stream surfaces constructed by the polygonal tiling of adjacent pairs of streamlines, it is focused only on steady flow where all the time derivatives of the flow field vanish. For real-time applications of the flame, Bridault and his colleagues [BLLR06, BLRR07] model flame shapes using NURBS surfaces. Although their approaches are similar to ours, smoke is more challenging because of complex behaviour and diffusive nature over time.

There are recent work [BB09, vFWTS08] that utilize surfaces for smoke visualization, which is similar in motivation to ours. In particular, Funck *et al.* [vFWTS08] reproduces thin smoke with semi-transparent streak surfaces very effectively. However, their approach relies on a simplified model. The smoke source is assumed to have a line-like structure such as a burning stick. The connectivity of streaklines is pre-determined by the user. The surface opacity that represents smoke density is determined by heuristic parameters while underlying flows are computed by the direct numerical Navier–Stokes simulation. Although Funck *et al.*'s method demonstrates its effectiveness in the scientific visualization field, it is not clear how well cigarette or candle smoke that this paper specifically tackle can be reproduced by their method. We analyse Funck *et al.*'s work in detail in 'Discussion'.

3. Method

3.1. Smoke simulation method

Our smoke animation utilizes the existing physics-based fluid simulation techniques. A time-varying velocity field is computed by numerically solving the Navier–Stokes equations.

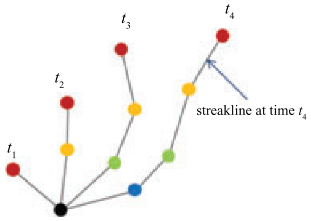

Figure 2: *A point source (black dot) temporally emits the coloured particles shown here as red, orange, green, and blue. At time* t_2, *for example, the only red and orange particles exist in the domain. The particles are advected by the time-varying velocity field. At a given time, a streakline is defined as a line passing through the particles starting from the same source. The connection follows the birth order of the particles.*

Several earlier researches [Sta99, FSJ01, KLLR05] outline the details of the Navier–Stokes equations and discuss methods of solving the equations numerically. The smoke density field is evolved in the simulation as in [FSJ01] and the massless particles from the source are advected passively within the velocity field. Our simulation is based on the fractional step method by [Sta99]. The vorticity confinement and monotonic cubic interpolation method by [FSJ01] are incorporated to model the small-scale rolling features. The Back and Forth Error Compensation Correction (BFECC) techniques by [KLLR05] is used for advecting both the density field and the particles. This method provides second-order accuracy in both space and time. In our implementation of the smoke simulation, open boundary condition was applied to remove an influence by walls, in which the boundary is regarded as the air. Under closed boundary condition, the smoke flow typically formed an excessive mushroom-like structure at its head due to the closed walls.

3.2. Streakline

We pay attention to the surface-like features observed in real smoke. It is difficult to reconstruct a surface representing smoke directly by fitting scattered particles. Moreover, a naive surface fitting method cannot guarantee temporal coherency. Instead, the particle movements are traced. Particles emitted from a source move according to the fluid flow to form a streakline. *Streaklines are the loci of fluid particles that have passed continuously through a particular spatial point in the past* (Figure 2) [Fab95]. Conceptually, we represent what we see from real smoke as a collection of streaklines and interpolation between them. Streaklines cannot self-intersect or intersect with each other, because the underlying velocity field is divergence-free.

It is trivial to construct a streakline from the emitted particles. At a user-specified source position, a particle is emitted with an initial velocity at a predetermined time interval. Here, a source ID is assigned to each source. All of the particles from a source are then inherited with the same source ID. As particles are created over time, the particles with the same source ID are saved sequentially. They can be connected either as a piecewise linear curve or as a more precise higher order curve. At this point, however, it is not necessary to generate exact fitting curves, as the particles are used to reconstruct a surface later. The data of the particles, such as the position and density (Rendering Section), are stored at every time step. Because the particles are advected with a time-varying continuous velocity field, the streaklines satisfy temporal coherency.

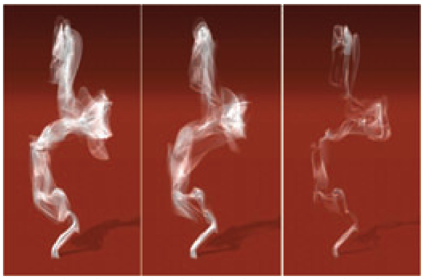
(a) no clustering (b) naive clustering (c) our clustering

Figure 3: *(a) One surface formed by all streaklines (b) manually selected four surfaces used (c) four surfaces created by our method.*

The number and positions of emitting sources affect the shape of the reconstructed surfaces. The user specifies the location and the number of sources. In our grid-based approach, a cell is designated as a source cell by assigning the user-specified velocity and density condition. One source cell can contain multiple point sources. In our cases, 10–20 point sources are placed randomly in a source cell. The initial velocity of a particle from the source is computed by interpolation between enclosing cell faces. Similarly, the density at a source is determined by neighbour cell densities. The emission period is another parameter affecting the surface shape and lies at the trade-off between computing resources and the resulting level of quality. We shoot a particle at a source every simulation time step. Such configuration works well with all our experiment results.

3.3. Surface reconstruction

Streaklines depict the overall motion of unsteady flows. We now turn to the explicit reconstruction of a surface to represent actual appearance of the smoke. Our method involves reconstructing a NURBS surface from each cluster of streaklines.

The quality of the final rendered result depends on how to generate the surfaces from the streaklines. Figure 3 shows three test cases of surface clustering. A surface formed by all the streaklines (no clustering) typically generates too opaque and thick appearance (a). The resulting images look unnatural even with a high sampling rate for opacity map (Rendering Section). Although a naive manual clustering might produce a better appearance than no clustering, the quality of the results are unpredictable depending on the user skill and time spent (b). To minimize manual intervention and produce a consistent look of semi-transparent smoke appearance, we use a carefully designed clustering algorithm (c).

A user partitions the point sources into a set of candidate groups and assigns a candidate group ID. Point sources located far from each other are likely to belong to different candidate groups. Moreover, close points can be assigned to different groups if desired by the user. The resulting surfaces are then created using the point sources within the same candidate group ID.

We observe that close streaklines in spatio-temporal space are likely to belong to a surface. Unlike [vFWTS08], our method does not predetermine the connectivity of the streaklines. The close smoke particles in an early stage cannot be assumed close in space throughout the simulation period, since the underlying flow is turbulent by nature. Each point source is associated with a time varying streakline. The distance h between two streaklines from the sources S_α and S_β, respectively, is defined by the average of Euclidean distances between the corresponding particles over simulation time, and is formulated as

$$h(\alpha, \beta) = \sum_{t=0}^{t_e} \left[\sum_{i=1}^{N(t)} |\mathbf{p}(\alpha, t, i) - \mathbf{p}(\beta, t, i)|^2 \right] \Big/ N(t).$$

Here, t_e is the end time of simulation and $\mathbf{p}(\alpha, t, i)$ is the position at time t of the ith particle starting from source S_α. $N(t)$ is the number of emitted particles from a source until time t. For example, $N(t) = t$ if one particle is emitted every time step. The number of streaklines that constitute a surface is unrestricted. All pairs from a candidate group are compared, and the same group ID is assigned to sources S_α and S_β if $h(\alpha, \beta)$ is less than the threshold h_0. A user may adjust h_0 whose default value is determined by the average value of hs. The number of reconstructed surfaces and participating streaklines varies with h_0 as can be seen from Figure 4. Although the overall structures remain similar, different h_0s produce different appearances. A high h_0 tends to include more streaklines resulting in more complicated overlapping surfaces. In contrast, a low h_0 tends to include fewer streaklines resulting in simpler surfaces. Note that depending on the choice of h_0, only a subset of streaklines can be used for surface reconstruction.

The shape of a reconstructed surface can differ depending on how the points across the streaklines are connected. One strategy involves finding the configuration with the minimum length (Figure 5). Computing the minimum length is a travelling salesman problem classified as NP-hard. Here, a greedy algorithm is utilized to solve this problem. First, a pair of point sources S_α and S_β is chosen, whose distance $h(\alpha, \beta)$ is the minimum among the point sources in a group G. Then, a point source S_γ is determined such that $h(\gamma, \alpha)$ or $h(\gamma, \beta)$ is the minimum among $h(\xi, \alpha)$ and $h(\xi, \beta)$, $\xi \in G$, $\xi \neq \alpha, \beta$.S_γ is then added to the current connection to form a new configuration. Repeating this process produces an efficient (although suboptimal) solution.

We build a NURBS surface passing through the streaklines with the same group ID (Figure 6). NURBS is effective in handling the smooth smoke flow and allows easy editing (see 'Discussion'). Although the sub-division modelling may also provide a smooth surface, it would require a base polygon

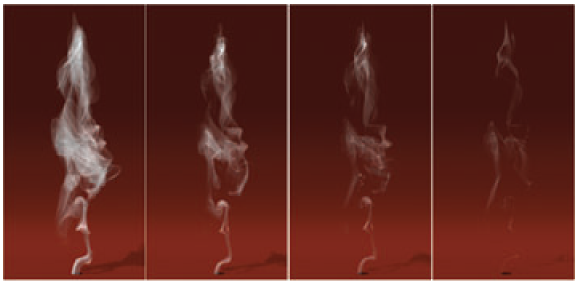
(a) h_0= 1.5*average 1 surface 20 streaklines
(b) h_0= 1.0*average 4 surfaces 16 streaklines
(c) h_0= 0.7*average 6 surfaces 14 streaklines
(d) h_0= 0.25*average 2 surfaces 4 streaklines

Figure 4: *Parameter tuning of threshold h_0.*

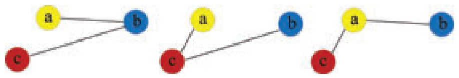

Figure 5: *Schematic alignment of streaklines from three sources, S_a, S_b and S_c where $h(a, c) < h(a, b) < h(b, c)$. Three cases are possible depending on the connections. The last case with the minimum length $h(a, c) + h(a, b)$ is used here.*

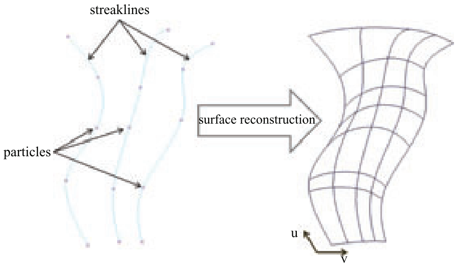

Figure 6: *Particles from a point source form a time-varying streakline (left). Streaklines by several point sources define a NURBS surface whose knot points are the particles of the streaklines (right).*

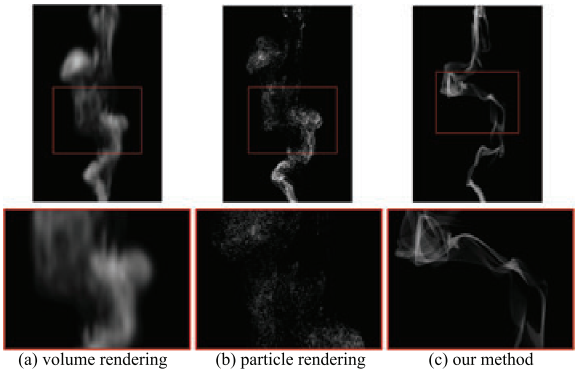

Figure 7: *Camera close-up.*

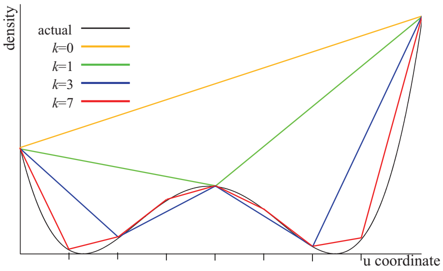

Figure 8: *Density map along the u coordinate is obtained by linearly interpolating sample points with sampling rate k. The map converges to the actual density field as k increases.*

mesh to be explicitly reconstructed from the particles to be used with our method. Clustered streaklines naturally imply an intrinsic parametrization for NURBS surface representation. Here, each particle in the streaklines serves as a knot. The u coordinate of the NURBS is the particle position in the streaklines. The v coordinate represents the position of each particle from the sources at a given time. Point-based rendering methods [PZvBG00, RL00, FCOS05, ABCO*03] may be useful to represent the embedded surface implied by a given point set. However, the algorithms focus on the unstructured point set acquired from scanning devices. In contrast, our method exploits the structure already established by streaklines. Moreover, partaking particles in our method are not dense enough to be suitable for point-based methods.

In contrast to the conventional approaches that utilize particles or volume rendering, we make use of the reconstructed NURBS surfaces for smoke visualization. As a rendering primitive, NURBS surface inherits the advantages of the vector graphics in opposition to bitmap images. When close up, a bitmap image inevitably reveals crude artefacts while the vector graphics is intrinsically scalable. Unlike particle-based or volume rendering, a vectorized representation of NURBS surface does not undergo quality degradation with camera close-up (Figure 7). This consideration was important for our choice of NURBS surface as one of the applications of our method is to replace the professional photographs with simulation results (Figure 12).

3.4. Rendering

We generate an opacity map to exhibit the semi-transparency of the reconstructed surfaces. The opacity at a particle position represents the smoke density. One way of computing the density is to count the number of particles nearby, which does not work in our case as only a small number of particles are employed. We utilize a density field instead, which can be concurrently computed in the simulation stage. The density field is advected by a velocity field describing a fluid flow. BFECC is used for the advection of the density field as well. The monotonic cubic interpolation scheme by [FSJ01] is adopted to reduce the numerical dissipation. A density value is assigned to a particle using the same interpolation scheme.

Density values sampled only at particle positions are often insufficient to form a complete density map of the corresponding NURBS surface. The density map constructed by interpolation using the small number of samples would not conform to the actual density field generated by simulation. A region without smoke (zero density value) may end up having a nonzero opacity value caused by the interpolation. Such an artefact becomes apparent when the particles travel in a wide range. To ensure the coherence, we perform supersampling on the surface. The particles on the streaklines have uniform coordinates in the UV plane. A user specifies the sampling rate to indicate how many points are needed between the particles in UV coordinates. Figure 8 shows an example with various sampling rates. The example is provided in one dimension for easy understanding. The density map without sampling ($k = 0$) is far from the actual density in most regions. As the sampling rate increases, the interpolated density map approaches to the actual density field.

For elevated realism, another important factor to consider is the contour of the surface. The contour has higher opacity than the inner region in real smoke, as shown in Figure 1(a). The contour is accentuated by the angle between the viewing direction $\mathbf{v}$ and surface normal $\mathbf{n}$. Combining these two quantities, the density d and the angle, the proposed opacity map p is formulated as $p = d \times \mathrm{e}^{-|\mathbf{n}\cdot\mathbf{v}|}$. In the experiments, the exponential function $\mathrm{e}^{-|\mathbf{n}\cdot\mathbf{v}|}$ produces better results than the simple attenuation function $1 - |\mathbf{n} \cdot \mathbf{v}|$.

4. Results

We compare the result synthesized by our method with real smoke (Figure 1). The objective of the comparison is to see whether our method can reproduce the characteristics

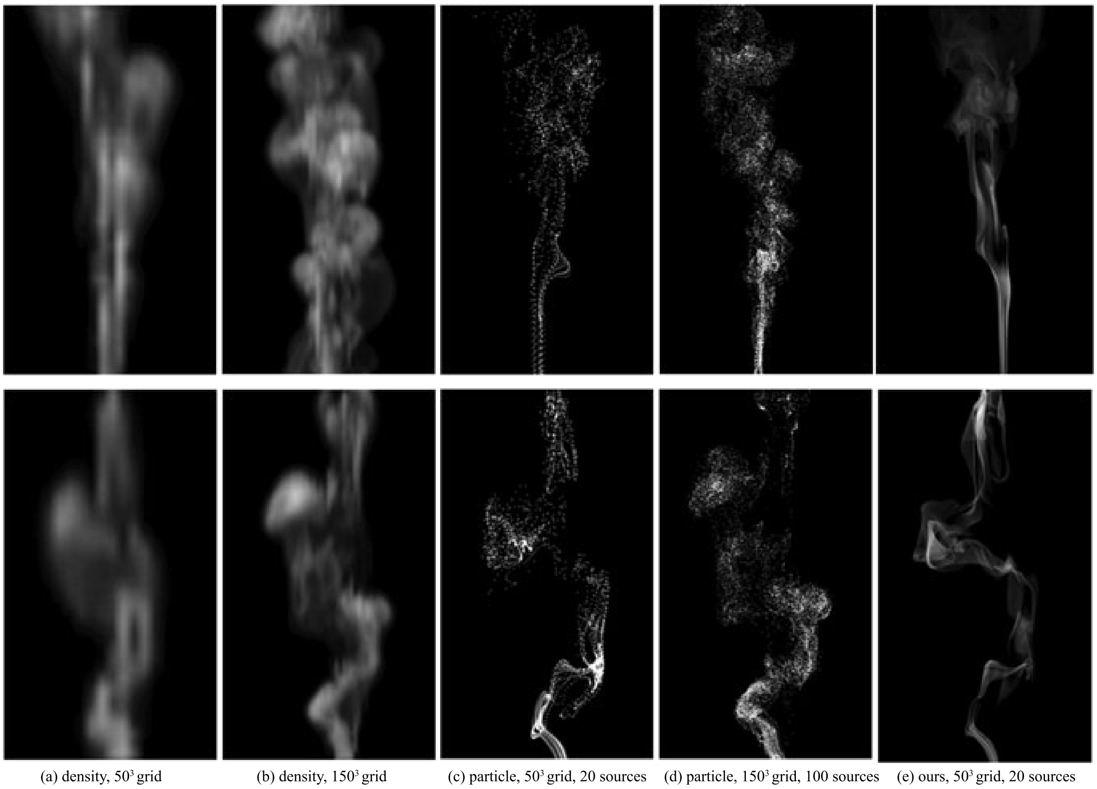
(a) density, 50³ grid
(b) density, 150³ grid
(c) particle, 50³ grid, 20 sources
(d) particle, 150³ grid, 100 sources
(e) ours, 50³ grid, 20 sources

Figure 9: *Comparison with previous methods (a) density field with 50^3 grid (b) density field with 150^3 grid (c) particle-based with 50^3 grid and 20 point sources (d) particle-based with 150^3 grid and 100 point sources (e) our method with 50^3 grid and 20 point sources.*

seen from a real scene. Similar to the real smoke motion and appearance, the synthesized image apparently shows a surface-like aspect, a conspicuous contour and a soft transition. Figure 1 shows the result generated from a low-grid resolution of 50^3. Twenty point sources were used and total of five surfaces were reconstructed with 11 streaklines. Despite the low-resolution simulation, artefacts such as jaggedness or excessive diffusion are not noticeable.

The superiority of our smoke visualization model compared to the previous approaches is demonstrated in Figure 9. Each row represents a different type of smoke animation. Smoke particles are advected passively by the velocity field computed using the Navier–Stokes equations (Section Smoke simulation method). The adjustment of the initial velocity condition, user specified wind and gravity affect the underlying velocity field. Simulations in the first, third and the last columns were performed with the same domain resolution of 50^3. The first column shows the result visualized via a volume rendering utilizing the density field, as in [FSJ01]. Clearly, low-resolution simulation with the previous methods is not capable of producing high-quality smoke visualization. The third column shows the rendering of the particles projected onto the image plane. At every time step, 20 particles were emitted into the domain. Our model used 20 point sources and generated five and four surfaces for the upper and lower row, respectively (the last column). As can be seen from the image, our method is superior to other conventional approaches in the low-resolution domain.

Our goal is to create surface-like features efficiently. We investigated whether conventional volume or particle rendering could be employed for this purpose. To have a greatly elaborated velocity field, the second column in Figure 9 shows the result exploited with a much higher grid resolution of 150^3 than the first column whereas other parameters remained the same. The increase of three times resolution for each axis requires about 27 times more memory and computation time. The fourth column has an increased domain resolution as does the second column and contains 100 point sources. The images in the fourth column show the smoke details. However, as apparent from the resulting images, a naive increase of parameters such as a grid resolution or the number of particles does not effectively reproduce the surface-like features which are apparent in the fifth column. In Figure 9, we cropped unnecessary parts of the images.

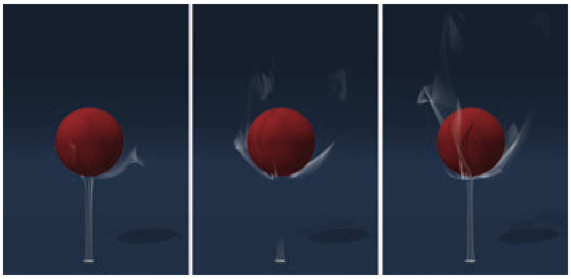

Figure 10: *Rising smoke collides with a static sphere (left panel). Source emits smoke again after a short interval (middle panel). The particles interact with the sphere (right panel).*

The discrepancy between the interpolated and the actual densities are expected to be large when the particles travel wide distances. To avoid visual artefacts from the distant particles, we exploited a supersampling method. Figure 10 demonstrates the effectiveness of our method. The source emitted smoke continuously except for a short interval. A sphere is inserted to the experiment to create various effects. No-slip condition was applied for the boundary condition of the sphere. Four source cells contained thirty randomly distributed point sources. The simulation was 260 frames long. The density value of 1.0 for source cells was set at frame 1–100 and 161–260. From the 101st to 160th frame, the zero density value was assigned to the source cells and particle emission was turned off. After colliding with the sphere, the smoke was stretched and the particles were separated in wide distances. Note that the gap between the particles around the interval is also wide. We set the sampling rate to be 4 and 2 in u- and v-direction, respectively. In Figure 10, the middle image clearly demonstrates that our surface-based method can represent the area with no density properly. In addition, three images show that the structure defined by the surfaces is well preserved even after the collision is happened.

Figure 11 shows the performance of our method when it is applied to realistic image synthesis. We first recorded a birthday party scene with a digital video camera, and then the original smoke motions were replaced by the result of our method. To simulate a wind blow, a velocity generated by a wind force is assigned to the source cells as the initial condition. We used five candidate groups for five candles. Each candle has one or two surfaces. The synthesized smoke exhibits convincing appearance. This example demonstrates that our method can be applicable to diverse image synthesis fields.

Professional photographers often arrange a well-conditioned studio with a careful light setup to capture the beauty of small-scale smoke flows. In an effort to mimic the actual photos with our method, we produced some of the images as shown in Figure 12. Ten to twenty point sources and three to five surfaces were used in the smoke scenes. The aesthetics of the smoke flows are very well represented.

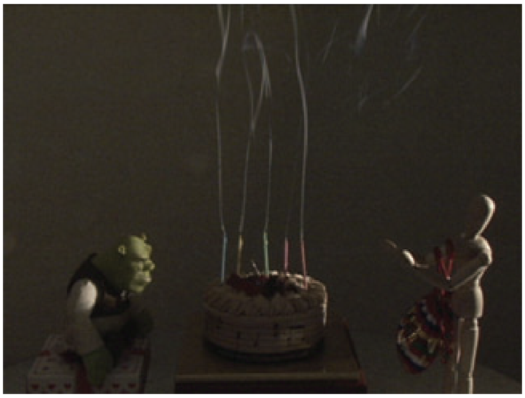

Figure 11: *A birthday party.*

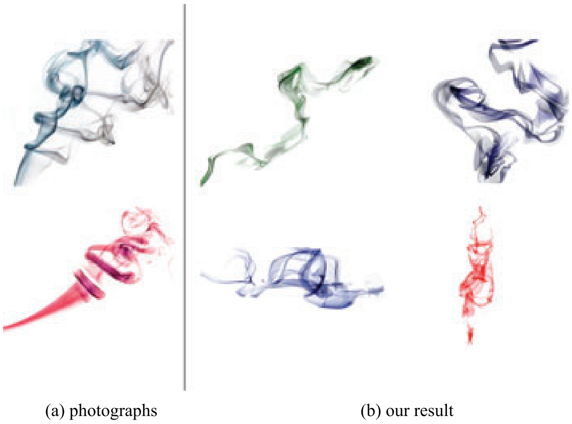

Figure 12: *Professionally photographed images vs. various results of our method.*

Our method was implemented as a MAYA plug-in and was rendered in MAYA on Intel Xeon node (2.8 GHz CPU and 4 GB RAM). Particles were rendered in the MAYA hardware renderer. Mental ray in MAYA was used for rendering the density field and the surfaces of our method. We rendered an image at every simulation time step. A volume shader in Mental ray was modified to handle the time-varying density field. The volume rendering of the density field required 12–38 s per frame depending on the domain resolution. The particle rendering took less than 1 s per frame. Given simulation data of 300 frames, we achieved the reconstructed surfaces within 1 min. Rendering of the reconstructed surfaces took about 4–5 s per frame depending on the number of the surfaces.

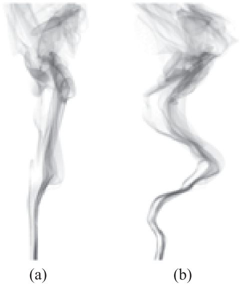

Figure 13: *Smoke surface editing (a) before and (b) after.*

5. Discussion

Our rendering primitives are NURBS surfaces. We acknowledge that NURBS surfaces are heavily used in geometry modelling for its easy editing capability. The same idea can be easily applied to our fluid simulation framework. Instead of repeating the smoke simulation over and over to achieve a desired result, a user may want to simply manipulate the control points of NURBS surfaces. For example, it is possible to generate a warped smoke animation from a tranquil smoke flow as shown in Figure 13. Although such warping is not accompanied with the physical constraints such as the enforcement of *divergence free*-ness, it can be applied to animation scenes where the user requirements are more important than the physical accuracy. We achieved the preliminary result with a manipulation of control points defined in the NURBS surfaces. However, such editing may require an elaborate mechanism that faithfully cater to various user requirements while ensuring temporal coherency. We leave this topic as a future work.

Because our method represents smoke only with surfaces, it is inefficient for the modelling of volume effects. For instance, after certain period of time, the smoke loses its surface-like structure and starts to show the volumetric appearance. Then our model alone often fails to accurately visualize the entire simulation. Seamless fusion between the surface representation and density field representation might be necessary to account for both effects. We believe that this can be another good candidate for future research.

For the realistic rendering of participating medium, one should model the physical phenomena such as scattering, absorption and emission of light. Currently, we do not consider such interaction of smoke particles with light. Although, the experiments show that our method produces reasonably good resulting images, we plan to investigate more complicated light interaction model in the future.

Surface reconstruction procedure in our method starts after the entire simulation is completed. Even with the same initial condition, the reconstructed surfaces may vary depending on the total length of simulation due to the clustering. Such characteristics prevent our method from being used for interactive applications. However, it may not be a big limitation considering that rendering stage in high-quality fluid animation is commonly separate from the simulation stage.

Our method samples points between particles in a uniform manner. Sampling rate is determined by a user-specified parameter regardless of the particle distribution. However, it may be more efficient to use an adaptive sampling method where a coarser region takes a higher sampling rate. We believe that the arc length parametrization of NURBS surface may be helpful for the adaptive sampling.

Source positions and the emission frequency in our method are fixed for simulation. Many sources and frequent emissions are required for describing the complex flow such as strong vortices. Recent adaptive schemes [KGJ09, BFTW09] can be useful to efficiently capture the complex flow.

The reconstructed surfaces may self-intersect or intersect with each other in a complex flow. We did not put much efforts to untangle the surfaces thinking that the tangled surfaces may still produce reasonably good results as the real surfaces formed by the smoke look very complicated. However, this issue requires further investigation.

To compare our method with Funck *et al.*'s [vFWTS08], we implemented their method and applied the same velocity field used in our examples. Instead of our NURBS surfaces, the polygonal surfaces were generated from the scattered particles. While our method determined the surface opacity using a simulated density field, Funck *et al.*'s method required approximated density and additional parameters such as regularity, curvature and the ages of polygons.

Our experiments revealed two drawbacks in their opacity assigning scheme when it is applied to a complex flow. First, their method does not maintain temporal coherency properly. Although the underlying velocity field is temporally continuous, unpleasant popping of polygons was unavoidably apparent (see Supporting Information). They provided the shape opacity parameter to hide irregular polygons (equation (6) in [vFWTS08]). This parameter only works for the calm flows used in their examples. Our velocity fields include complex flows producing sudden changes of polygon regularity. Secondly, they assigned low opacity to high mean curvature region. Such configuration tends to produce planar smoke shapes instead of complex ones. Our method excels in this aspect as shown in Figures 14(a) and (c).

To verify the effectiveness of our clustering scheme, we applied our opacity map to Funck *et al.*'s polygonal geometry instead of opacity computed by their method. The result (Figure 14(b)) shows similar appearance to our no clustering example in Figure 3(a). Although we cluster close surfaces in spatio-temporal space and enhance the visual realism, their method has no such clustering procedure and produces the unnatural and thick appearance when the simulated density data was used.

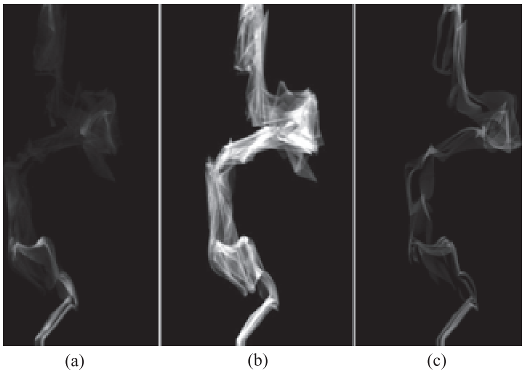

Figure 14: *(a) Funck et al.'s method* [vFWTS08] *loses much detail as it assigns low opacity to high curvature regions. (b) The method inevitably exhibits thick appearance when our opacity map with simulated density was applied. (c) Our method shows realistic appearance of complex smoke.*

6. Conclusion

We present a novel smoke visualization model for capturing the surface-like features. By utilizing streaklines that are created by particle traces, we reconstruct the time-varying NURBS surfaces that effectively describe smoke motion and appearance. Our model is simple to implement, very efficient in its representation of semi-transparent smoke features, and innovative in lessening the computational loads. This model works with any higher order simulation mechanism.

Acknowledgments

This work was supported by the Korea Research Foundation Grant funded by the Korean Government[KRF-2008-357-D00225] and IT R&D program of KEIT (KI001732-2009-03, Software Development for Digital Creature). The authors also thank the anonymous reviewers for their considerable help in preparing the work for publication.

References

[ABCO*03] Alexa M., Behr J., Cohen-Or D., Fleishman S., Levin D., Silva C. T.: Computing and rendering point set surfaces. *IEEE Transactions on Visualization and Computer Graphics 9* (2003), 3–15.

[AN05] Angelidis A., Neyret F.: Simulation of smoke based on vortex filament primitives. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2005), pp. 87–96.

[BB09] Brochu T., Bridson R.: Animating smoke as a surface. In *Proceedings of the Symposium on Computer Animation 2009, Poster Session* (2009).

[BFTW09] Buerger K., Ferstl F., Theisel H., Westermann R.: Interactive streak surface visualization on the gpu. *IEEE Transactions on Visualization and Computer Graphics 15* (2009), 1259–1266.

[BH05] Batty C., Houston B.: Visual simulation of wispy smoke. In *Proceedings of the ACM SIGGRAPH '05 Sketches* (2005).

[BLLR06] Bridault-Louchez F., Leblond M., Rousselle F.: Enhanced illumination of reconstructed dynamic environments using a real-time flame model. *AFRIGRAPH'06* (2006), 31–40.

[BLRR07] Bridault F., Lebond M., Rousselle F., Renaud C.: Real-time rendering and animation of plentiful flames. In *Proceedings of the 3rd Eurographics Workshop on Natural Phenomena* (2007), pp. 31–38.

[BML95] Becker B. G., Max N. L., Lane D. A.: Unsteady flow volumes. In *Proceedings of the Sixth IEEE Visualization 1995(VIS '95)* (1995), 329–337.

[DKY*00] Dobashi Y., Kaneda K., Yamashita H., Okita T., Nishita T.: A simple, efficient method for realistic animation of clouds. In *Proceedings of the ACM SIGGRAPH'00* (2000), pp. 19–28.

[Fab95] Faber T. E.: *Fluid Dynamics for Physicists*. Cambridge University Press, UK, 1995.

[FCOS05] Fleishman S., Cohen-Or D., Silva C. T.: Robust moving least-squares fitting with sharp features. In *Proceedings of the ACM SIGGRAPH '05* (2005), pp. 544–552.

[FL04] Fattal R., Lischinski D.: Target-driven smoke animation. In *Proceedings of the ACM SIGGRAPH '04* (2004), pp. 441–448.

[FOA03] Feldman B. E., O'brien J. F., Arikan O.: Animating suspended particle explosions. In *Proceedings of the ACM SIGGRAPH '03* (2003), pp. 708–715.

[FOK05] Feldman B. E., O'brien J. F., Klingner B. M.: Animating gases with hybrid meshes. In *Proceedings of the ACM SIGGRAPH '05* (2005), pp. 904–909.

[FSJ01] Fedkiw R., Stam J., Jensen H. W.: Visual simulation of smoke. In *Proceedings of the ACM SIGGRAPH '01* (2001), pp. 15–22.

[Hou] Houdini: *Side Effects Software*.

[Hul92] Hultquist J. P. M.: Constructing stream surfaces in steady 3D vector fields. *Proceedings of the IEEE Visualization'92* (1992), pp. 171–179.

[JC98] Jensen H. W., Christensen P. H.: Efficient simulation of light transport in scenes with participating media using photon maps. In *Proceedings of the ACM SIGGRAPH'98* (1998), pp. 311–320.

[KGJ09] KRISHNAN H., GARTH C., JOY K.: Time and streak surfaces for flow visualization in large time-varying data sets. *IEEE Transactions on Visualization and Computer Graphics 15* (2009), 1267–1274.

[KLLR05] KIM B., LIU Y., LLAMAS I., ROSSIGNAC J.: FlowFixer: Using BFECC for fluid simulation. In *Eurographics Workshop on Natural Phenomena* (2005).

[KMT06] KIM Y., MACHIRAJU R., THOMPSON D.: Path-based control of smoke simulations. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2006), pp. 33–42.

[KTJG08] KIM T., THUREY N., JAMES D., GROSS M.: Wavelet turbulence for fluid simulation. In *Proceedings of the ACM SIGGRAPH '08* (2008).

[LF02] LAMORLETTE A., FOSTER N.: Structural modeling of flames for a production environment. In *Proceedings of the ACM SIGGRAPH '02* (2002), pp. 729–735.

[LGF04] LOSASSO F., GIBOU F., FEDKIW R.: Simulating water and smoke with an octree data structure. In *Proceedings of the ACM SIGGRAPH '04* (2004), pp. 457–462.

[LHD*04] LARAMEE R. S., HAUSER H., DOLEISCH H., VROLIJK B., POST F. H., WEISKOPF D.: The state of the art in flow visualization: Dense and texture-based techniques. *Computer Graphics Forum 23*, 2 (2004), 203–221.

[Max05] MAX N.: Progress in scientific visualization. *The Visual Computer 21*, 12 (2005), 979–984.

[MCPN08] MOLEMAKER J., COHEN J. M., PATEL S., NOH J.: Low viscosity flow simulations for animation. In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2008).

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

[PVH*02] POST F. H., VROLIJK B., HAUSER H., LARAMEE R. S., DOLEISCH H.: Feature extraction and visualization of flow fields. *Eurographics '02 State-of-the-Art Reports* (2002), pp. 69–100.

[PZvBG00] PFISTER H., ZWICKER M., VAN BAAR J., GROSS M.: Surfels- surface elements as rendering primitives. In *Proceedings of the ACM SIGGRAPH '00* (2000), pp. 335–342.

[Rea] Realflow: *Next Limit Tech.*

[RL00] RUSINKIEWICZ S., LEVOY M.: QSplat: A multiresolution point rendering system for large meshes. In *Proceedings of the ACM SIGGRAPH '00* (2000), pp. 343–352.

[SDE05] SCHPOK J., DWYER W., EBERT D. S.: Modeling and animating gases with simulation features. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2005), pp. 97–105.

[SSP*05] STAUBLI O., SIGG C., PEIKERT R., GUBLER D., GROSS M.: Volume rendering of smoke propagation CFD data. In *Proceedings of the IEEE Visualization '05* (2005), pp. 335–341.

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

[TvW03] TELEA A., VAN WIJK J. J.: 3D IBFV: Hardware-accelerated 3D flow visualization. In *Proceedings of the IEEE Visualization '03* (2003), pp. 233–240.

[vFWTS08] VON FUNCK W., WEINKAUF T., THEISEL H., SEIDEL H. P.: Smoke surfaces: An interactive flow visualization technique inspired by real-world flow experiments. *IEEE Transactions on Visualization and Computer Graphics 14* (2008), 1396–1403.

[vW04] VAN WIJK J. J.: Image based flow visualization. In *Proceedings of the ACM SIGGRAPH '02* (2004), pp. 99–106.

[WTS*07] WIEBEL A., TRICOCHE X., SCHNEIDER D., JAENICKE H., SCHEUERMANN G.: Generalized streak lines: Analysis and visualization of boundary induced vortices. *IEEE Transactions on Visualization and Computer Graphics 13* (2007), 1735–1742.

[ZHG*07] ZHOU K., HOU Q., GONG M., SNYDER J., GUO B., SHUM H. Y.: Fogshop: Real-time design and rendering of inhomogeneous, single-scattering media. In *Proceedings of the 15th Pacific Conference on Computer Graphics and Applications (PG'07)* (2007), pp. 116–125.

Supporting Information

Additional Supporting Information may be found in the online version of this article.

Please note: Wiley-Blackwell is not responsible for the content or functionality of any supporting materials supplied by the authors. Any queries (other than missing material) should be directed to the corresponding author for the article.