九州大学学術情報リポジトリ
Kyushu University Institutional Repository

A Simple Method for Morphing Smoke

Dobashi, Yoshinori
Division of Media and Network Technologies, Graduate School of Information Science and Technology, Hokkaido University : Associate Professor

Tani, Tsubasa
Hokkaido University

Sato, Syuhei
UEI Research

Yamamoto, Tsuyoshi
Division of Media and Network Technologies, Graduate School of Information Science and Technology, Hokkaido University : Professor

https://hdl.handle.net/2324/1546881

出版情報：MI lecture note series. 64, pp.54-60, 2015-09-18. 九州大学マス・フォア・インダストリ研究所
バージョン：
権利関係：

MATHEMATICAL PROGRESS IN EXPRESSIVE IMAGE SYNTHESIS 2015
Sep 25-27, 2015, Nishijin Plaza Kyushu University, Fukuoka

A Simple Method for Morphing Smoke

Yoshinori Dobashi*
Hokkaido University/JST CREST

Tsubasa Tani†
Hokkaido University

Syuhei Sato‡
UEI Research

Tsuyoshi Yamamoto§
Hokkaido University

Abstract Recently, computer graphics has been used in many applications, such as computer games and movies. In the field of computer graphics, physically based simulation of fluids is one of the most important research topics and it allows us to synthesize highly realistic images of fluid phenomena. However, one of the problems with the fluid simulation is the high computational cost, making its applicability limited. This paper proposes a method for efficiently synthesizing smoke animations by applying the idea of image morphing to the simulated density distributions of smoke. We precompute a set of dynamic density distributions by numerical fluid simulation and then create inbetween density distributions by deforming and blending the precomputed density distributions.

Keywords: Smoke, Fluid Simulation, Morphing

1 Introduction

Recently, we often see synthetic images created by computer graphics techniques in many applications, such as computer games and movies. Synthesizing realistic animations of fluid phenomena, such as smoke and water, has been an active research area in computer graphics and many methods have been proposed [2]. Most of the recent methods are based on the numerical analysis of the governing equations of fluid motion called Navier-Stokes equations. Since the numerical analysis of the equations is generally time-consuming, many acceleration methods have been proposed. However, creating realistic and high-resolution animation of fluids is still expensive. Moreover, the user has to adjust many parameters involved in those methods repeatedly until the desired visual effects are obtained. The expensive computational cost makes this adjustment process extremely tedious.

In this paper, we present a method for efficiently synthesizing realistic animations of smoke by using the idea of image morphing. We develop a method for morphing dynamic density distributions of smoke that are obtained by numerically solving the Navier-Stokes equations. In a preprocess, we simulate the smoke with different parameter settings by using the grid-based method [5]. Our method then synthesizes inbetween dynamic density distributions from the precomputed distributions of smoke. In order to achieve the morphing of smoke, we need the following three components:

- a method to determine corresponding points between different density distributions of smoke,

*doba@ime.ist.hokudai.ac.jp
†tani@ime.ist.hokudai.ac.jp
‡syuhei.sato@uei.co.jp
§yamamoto@ist.hokudai.ac.jp

- a deformation method for the smoke distribution, and
- a blending method between smoke densities.

For the corresponding points, we ask the user to interactively specify a representative path of the flow for each of the precomputed smoke density distributions. We call the representative path a flow curve. Next, the grid storing the smoke densities is deformed by using the flow curve. The densities are then linearly interpolated. We apply our method to synthesize dynamic smoke blown by wind and demonstrate the efficiency of the method.

The overview of the paper is as follows. In Section 2, we discuss some of the related work. Section 3 describes an overview of our system. Section 4 briefly explains the numerical fluid analysis that are used in our preprocess for creating dynamic density distributions of smoke. Our morphing method is then explained in Section 5. In Section 6, some experimental results are shown. Finally, Section 7 concludes the paper with discussion on limitations and future work of our method.

2 Related Work

Stam [17] addressed the stability problem in solving the Navier-Stokes equations and made the fluid simulation practical. After this work, many methods have been proposed for simulating various fluid phenomena. Readers can find details of those methods in [2]. One of the problems with the fluid simulation, however, is the expensive computational cost. Many methods have therefore been proposed to reduce the cost [9, 3, 1, 4, 8]. However, high-resolution simulation is still costly and tuning simulation parameters with repeated simulations is time-consuming. Some researchers use 2D fluid simulations for efficiently creating high-resolution animations of explosions [13] or fire [6][14]. These methods can reduce the computational cost significantly compared to high-resolution 3D fluid simulations. However, these methods are limited to the simulation of fluids that can be roughly approximated by a set of 2D simulations.

Recently, several methods have been developed to create high-resolution results from a low-resolution simulation. Nielson et al. [11, 10] proposed methods for simulating high-resolution fluids that resemble a reference low-resolution simulation result. Yuan et al. [20] proposed a method that regulates high-resolution fluid simulation with flow patterns extracted from a low-resolution fluid simulations. Although these methods can generate high-resolution results with desired fluid behavior, costly high-resolution simulations are required to produce final animations. By combining low-resolution fluid simulation with turbulent noise functions, high-resolution results can be synthesized without conducting any high-resolution fluid simulations [7, 16, 12]. These methods can add small scale details but the results are not very realistic when compared to the results obtained by fluid simulations.

Treuille et al. [18] and Wicke et al. [19] proposed data-driven methods for accelerating the fluid simulations. However, one problem with these methods is that the methods require significantly long precomputation time ranging from twenty to thirty hours.

3 Overview of Our Method

The input to our method is multiple dynamic density distributions of smoke that are simulated by numerically solving the NS equations. For the simulation of smoke, we use the method developed by Fedkiw et al [5]. The density distribution is then represented by using a grid and is stored at discrete time step $t(= 1, 2, \cdots)$. Our goal is then to synthesize inbetween density distributions by

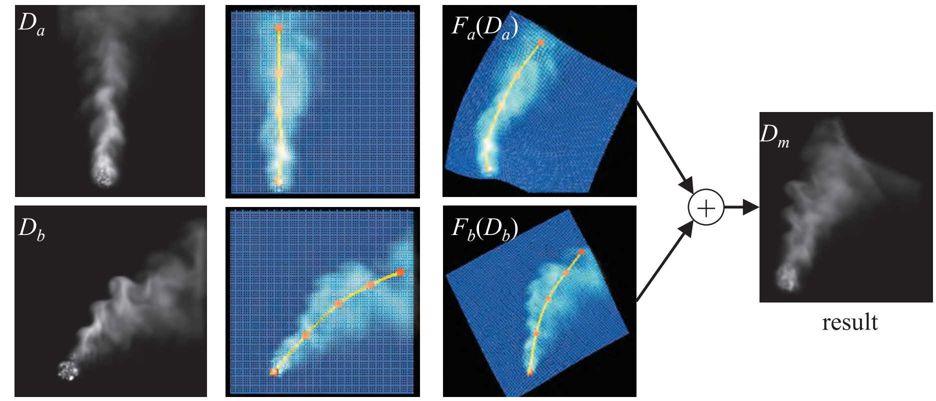


Figure 1: Overview of our method.

interpolating the precomputed dynamic density distributions. The simplest way to achieve this is to simply blend the precomputed density distributions but it does not work very well. It is obvious that the so-called ghosting artifact appears.

In order to address this problem, we borrow the idea from the image morphing; the precomputed density distributions are deformed first and then blended. Fig. 1 shows an overview of our method. We first ask the user to specify the center line of the smoke or *flow curve* for each of the precomputed dynamic density distributions. These curves allow us to make corresponding points between the precomputed density distributions. Next, an inbetween curve is generated by linearly interpolating the corresponding points of the user-specified flow curves. The grid storing the precomputed density distribution is deformed such that the flow curve coincides with the inbetween curve. For the deformation of the grid, we use the method developed by Schaefer et al [15]. The deformed density distributions are then blended together to create the final density distribution that are visualized by using a volume rendering technique.

4 Smoke Simulation

We assume the inviscid, incompressible flow for creating the input density distributions of smoke. The motion of the smoke is then calculated by solving the following Navier-Stokes equations.

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \cdot \nabla)\mathbf{u} - \frac{1}{\rho}\nabla p + \mathbf{f}, \qquad (1)$$

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

where $\mathbf{u}$ is the velocity of the fluid, ρ is the density, p is the fluid pressure, and $\mathbf{f}$ represents any external forces such as gravity and wind. We use the GPU to accelerate the computation to solve the above equations numerically. We also use the vorticity confinement method [5] in order to add turbulent motions.

For simulating smoke, we simply advect the smoke densities along with the velocity field obtained by solving the above equations. The smoke density D is therefore computed by using the following

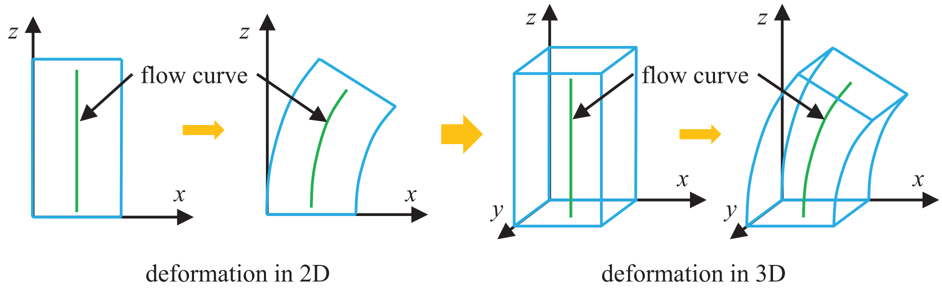


Figure 2: Deformation of grids. The deformation in 2D (left) is applied to the grid in 3D (right).

equation.

$$\frac{\partial D}{\partial t} = -(\mathbf{u} \cdot \nabla)D + D_s, \tag{3}$$

where D_s is the density of the smoke source.

5 Morphing Method

This section describes the method for morphing the precomputed density distributions. There are two types of morphing in our method: spatial and temporal morphing. The spatial morphing deforms the grids so that the shapes of the precomputed smoke roughly match with each other. The temporal morphing is used to make the speed of the smoke motions become the same.

5.1 Spatial Morphing

Let us explain our method by assuming that we have two input density distributions D_a and D_b as shown in Fig. 1. Our system first asks the user to specify the flow curves by drawing spline curves on the screen. The system then generate a set of sample points on the curve. The inbetween flow curve is then computed by interpolating the corresponding sample points linearly according to the user-specified blending factor α. When p_i and q_i are the sample points on the two user-specified flow curves, sample point r_i on the inbetween flow curve is represented by:

$$r_i = (1 - \alpha)p_i + \alpha q_i \tag{4}$$

We then deform the two grids storing the precomputed density distributions so that p_i and q_i coincide with r_i by using the method proposed by Schaefer et al [15]. As shown in Fig. 2, our system first computes the two-dimensional deformation by using p_i, q_i, and r_i. Then, this deformation is applied to every cross section of the grid perpendicular to the view direction. The density distribution of the morphed smoke, D_m, at time step $t(= 1, 2, \cdots)$ is obtained by using the following equation.

$$D_m(t) = (1 - \alpha)F_a(D_a(t)) + \alpha F_b(D_b(t)), \tag{5}$$

where F_a and F_b represent the deformation operators applied to D_a and D_b, respectively. That is, $F_a(D_a)$ and $F_b(D_b)$ represent the deformed density distributions.

5.2 Temporal Morphing

By using the method described in the previous subsection, the shapes of the deformed smoke distributions become similar. However, when the speed of the smoke motion is different from that of the other smoke, the ghosting artifact will appear. That is, we perceive two different overlapping smoke distributions moving at different speeds. We address this problem by artificially modifying the speed of the smoke motion in the following way.

During the precomputation process, we record an average speed of the motion at the center position of the smoke distribution. The center position is specified by the user. In creating the final animation using our morphing method, the speed of the smoke motion is adjusted so that the recorded average speed becomes the same. For example, when we have two precomputed density distributions and their recorded average speeds are v_a and v_b, respectively, we use the following equations instead of Eq. 5 to obtain the final density distribution at time step t:

$$
\begin{aligned}
D_m(t) &= (1-\alpha)F_a(D_a(t_a)) + \alpha F_b(D_b(t_b)), && (6)\\
t_a &= t_a - 1 + v/v_a, && (7)\\
t_b &= t_b - 1 + v/v_b, && (8)\\
v &= (1-\alpha)v_a + \alpha v_b. && (9)
\end{aligned}
$$

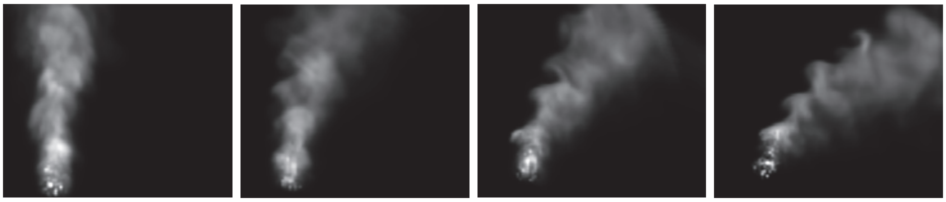
Note that t_a and t_b are no longer integer values and we compute $D_a(t_a)$ and $D_b(t_b)$ by linearly interpolating the temporally neighboring precomputed density distributions.

6 Results

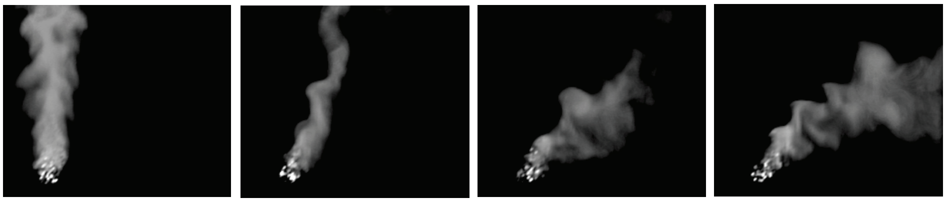
This section shows results obtained by using the proposed method. We used a desktop PC with an Intel Core i7-3770 (CPU) and an NVIDIA GeForce GTX 660 (GPU) to create the following examples. The GPU was used for the precomputation only and the morphing process was computed on the CPU.

Fig. 3 shows an experimental result created to demonstrate the validity of our method. We precomputed two dynamic density distributions with and without applying wind forces. The number of grid points is $100 \times 50 \times 100$. The number of time steps of the simulation was 180 and it took 327 seconds for the precomputation. We then compare the smoke animation created by using our morphing method (Fig. 3(a)) with the animation created by solving the NS equations by gradually increasing the wind speed (Fig. 3(b)). Although the small-scale details are different between these animations, overall motions are the same as shown in Fig. 3. Figs. 3(a) and (b) took 40 and 174 seconds, respectively. Although our method requires a long precomputation process, the computational cost for creating the smoke animation is approximately four times faster than that by solving the NS equations. In addition, once the precomputation has finished, smoke animations with different wind speeds can be efficiently generated by using our method.

In Fig. 4, we apply our method to the simulation of smoke rising from a steam locomotive. The grid size for the smoke simulation for the precomputation is $200 \times 100 \times 200$. The precomputation and the morphing process took 80 and 12 minutes, respectively. Our method can successfully synthesize realistic and high resolution smoke animation efficiently.



(a) smoke simulated by using our morphing method



(b) smoke simulated by solving NS equations

Figure 3: Comparison of smoke animations created by using our method and by solving NS equations.

Figure 4: Examples of smoke rising from a steam locomotive.

7 Conclusions

We have proposed the morphing method for smoke simulations. In the method, multiple dynamic density distributions of smoke are precomputed by numerical analysis of NS equations with different parameter settings. The smoke animations are efficiently generated by deforming and blending the precomputed density distributions. We demonstrated the validity and the efficiency of the proposed method by several examples. Our method can successfully generate realistic animations of smoke.

There are several issues to be addressed in the future. In our current method, the flow curves are specified by the user. We would like to develop a method that can automatically determine the flow curves. The corresponding points between the precomputed smoke density distributions are determined by using the flow curves only. It would be better to determine the corresponding points so that the differences between the density distributions are minimized. Accelerating our morphing process using the GPU is also an interesting and important topic for real time applications such as computer games.

References

[1] C. Batty, F. Bertails, and R. Bridson. A fast variational framework for accurate solid-fluid coupling. *ACM Transaction on Graphics*, 26(3):100, 2007.

[2] Robert Bridson. *Fluid Simulation for Computer Graphics*. AK Peters, 2008.

[3] Keenan Crane, Ignacio Llamas, and Sarah Tariq. *Real Time Simulation and Rendering of 3D Fluids*, chapter 30. Addison-Wesley, 2007.

[4] Y. Dobashi, Y. Matsuda, T. Yamamoto, and T. Nishita. A fast simulation method using overlapping grids for interactions between smoke and rigid objects. *Computer Graphics Forum*, 23(3):539–546, 2008.

[5] R. Fedkiw, J. Stam, and H. W. Jensen. Visual simulation of smoke. In *Proc. SIGGRAPH 2001*, pages 15–22, 2001.

[6] Christopher Horvath and Willi Geiger. Directable, high-resolution simulation of fire on the gpu. *ACM Transaction on Graphics*, 28(3):Article 41, 2009.

[7] Theodore Kim, Nils Thürey, Doug James, and Markus Gross. Wavelet turbulence for fluid simulation. *ACM Transaction on Graphics*, 27(3):Article 50, 2008.

[8] Michael Lentine, Wen Zheng, and Ronald Fedkiw. A novel algorithm for incompressible flow using only a coarse grid projection. *ACM Transaction on Graphics*, 29(4):Article 114, 2010.

[9] F. Losasso, F. Gibou, and R. Fedkiw. Simulating water and smoke with an octree data structure. *ACM Transaction on Graphics*, 23(3):457–462, 2004.

[10] Michael B. Nielsen and Brian B. Christensen. Improved variational guiding of smoke animations. *Computer Graphics Forum*, 29(2):705–712, 2010.

[11] Michael B. Nielsen, Brian B. Christensen, Nafees Bin Zafar, Doug Roble, and Ken Museth. Guiding of smoke animations through variational coupling of simulations at different resolutions. In *Proc. ACM SIGGRAPH/Eurographics Symposium on Computer Animation 2009*, pages 217–226, 2009.

[12] Tobias Pfaff, Nils Thürey, Jonathan Cohen, Sarah Tariq, and Markus Gross. Scalable fluid simulation using anisotropic turbulence particles. *ACM Transaction on Graphics*, 29(6):Article 174, 2010.

[13] Nick Rasmussen, Duc Quang Nguyen, Willi Geiger, and Ronald Fedkiw. Smoke simulation for large scale phenomena. *ACM Transactions on Graphics*, 22(3):703–707, aug 2003.

[14] Syuhei Sato, Yoshinori Dobashi, and Tsuyoshi Yamamoto. A data-driven approach for synthesizing high-resolution animation of fire. In *Proc. Digital Production Symposium 2012*, pages 37–42, 2012.

[15] S. Schaefer, T. McPhail, and J. warren. Image deformation using moving least squares. *ACM Transaction on Graphics*, 25(3):533–540, 2009.

[16] H. Schechter and R. Bridson. Evolving sub-grid turbulence for smoke animation. In *Proc. the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 1–7, 2008.

[17] Jos Stam. Stable fluids. In *Proceedings of ACM SIGGRAPH 1999, Annual Conference Series*, pages 121–128, aug 1999.

[18] Adrien Treuille, Andrew Lewis, and Zoran Popovič. Model reduction for real-time fluids. *ACM Transaction on Graphics*, 25(3):826–834, 2006.

[19] Martin Wicke, Matt Stanton, and Adrien Treuille. Modular bases for fluid dynamics. *ACM Transaction on Graphics*, 28(3):Article 39, 2009.

[20] Zhi Yuan, Fan Chen, and Ye Zhao. Pattern-guided smoke animation with lagrangian coherent structure. *ACM Transaction on Graphics*, 30(6):Article 136, 2011.