**Noname manuscript No.**
(will be inserted by the editor)

# The Characteristic Map for Fast and Efficient VFX Fluid Simulations

**Jerry Tessendorf · Brandon Pelfrey**

the date of receipt and acceptance should be inserted later

**Abstract** The Method of Characteristics is examined as a tool for making fluid simulation more efficient and effective in VFX production. A mathematical framework for a Characteristic Map is shown to be a generalization of the previous methods called Gridless Advection and Semi-Lagrangian Mapping. We demonstrate that the Characteristic Map can be used to modify a fluid flow post-simulation, including injecting higher resolution motion, and precise flow control from blending Characteristic Maps.

**Keywords** Fluid Simulation · Characterist Function · Volume Rendering · Visual Effects

## 1 Introduction

The pressures of VFX production continually drive studios to find new and improved methods for fluid simulation that simplify the pipeline, improve and control visual quality, improve speed, reduce memory, and expand artistic stylization of the simulations. One successful approach has been to apply various post-simulation volumetric manipulations using the density and velocity fields, without further simulation. For example, Gridless advection (GA) and SEmi-Lagrangian MApping (SELMA) have been successfully applied in several shows [1]. These procedures generate a density field showing much more detail and visually-interesting features that the advected density coming directly from the simulation. We examine the natural extension of those procedures by incorporating the mathematical framework of the Characteristic Map (CM) function, and show that it acts as a unifying computational approach that can be applied to fluid simulations for several purposes:

1. For a desired level of detail, smoke simulations can be generated at lower spatial grid resolution, then re-advected via the CM.
2. New types of tradeoffs are made available. For example, the density field can be built entirely from procedural calculations, while still being advected through a velocity field. The CM can be entirely procedurally built from the velocity field, or sampled to a grid repeatedly for memory savings, while producing higher spatial detail than advection and gridding of the density directly.
3. Multiple unrelated simulations can be merged seamlessly with standard blending methods operating on the CM. This even includes densities and velocity fields that are procedurally generated without simulation.
4. Compressible flows have additional impacts on the density field, and this can be applied using the CM and a quenching factor shown in section 3.3.2. For finite-in-time stepping of the density and velocity fields, this factor can grow or collapse exponentially, spoiling the visual impact of the field. We show a way to mitigate this for VFX applications.

The goal of these post-simulation manipulations is to extend the utility of the fluid simulations without having to re-run simulations or start over with entirely new ones. We hope to make fluid simulations fit better into the standard practice of VFX production through tools that allow iterative adjustment, editing, and blend-

Jerry Tessendorf
School of Computing, Clemson University
Rhythm & Hues Studios
E-mail: jtessen@clemson.edu

Brandon Pelfrey
School of Computing, Clemson University
E-mail: bpelfre@clemson.edu

ing of simulation results in a style consistent with other areas of VFX.

The rest of this paper is devoted to two tasks: quantifying the CM and it's computation from the velocity field, and demonstrating the utility of the items above. We illustrate these points using a simple smoke simulation generated at low spatial resolution, large time steps, with a highly dissipative advection scheme. Not only is this simulation easy to generate on modest hardware, the much improved visual quality of the CM-based modifications emphasize the production benefits of the Characteristic Map.

Section 3 lays out the mathematical structure of the Characteristic Map, beginning with it's forerunners in production, gridless advection and Semi-Lagrangian Mapping. The evolving density field is derived for incompressible and compressible velocity fields. Applications of the CM to the tasks listed above are discussed in section 4, and illustrated with the simple smoke simulation.

## 2 Previous Work

The method of characteristics has enjoyed wide and varied adoption in PDE research, yet has seen little exploration in computer graphics. The popular semi-Lagrangian advection schemes which can be viewed as an application of the method of characteristics [2]. These methods have been attractive for their ease of implementation and unconditional stability. One of the main drawbacks of this method comes from examining its effect on the physical equations. It has been shown in [3] that the semi-Lagrangian method solves a perturbed equation which has a velocity-dependent diffusion term. This diffusion behavior has been problematic since it diffuses density details which we would like to sharpen for production scenarios. The characteristic mapping approach ameliorates this issue.

The method of characteristics has been used in studying many hyperbolic problems, especially those where advection plays a larger role than other effects like diffusion, the so called ‘transport-dominated’ problems. In the field of fluid dynamics where the transport terms can be significant and are non-linear, the method has been used to accurately compute advection for two-phase flows [4] using a fractional time-stepping procedure similar. This splitting process, which is similar in spirit to the typical Stam-style solver framework, performs an error analysis showing that the method of characteristics leads to an error bounded by terms of $O(\Delta t^2)$.

Our application of the CM differs from these past applications, in that we use the CM after the simulation is complete to reconstruct a new version of the density field. The method preserves strong gradients in the density better than the simulations do, and so reveals more spatial structure than the density fields directly produced by the simulations.

The calculation of the CM bears resemblance to schemes for advecting textures [5]. There are significant differences however because the goals of CM and advected textures are quite different. One of the critical issues in advecting textures is a refreshment process to restore or maintain the qualities of the texture because advection would otherwise stretch and distort it beyond recognition. There is no such concern in our approach, because we are focused on methods to enhance and alter the fluid simulation for control and iteration purposes, without regard to the impact on textures. Texture advection is a separate concern which could be applied independent of our process.

## 3 Formulation of the Characteristic Map

Before introducing the CM in it's general form, it is worthwhile to look at two special cases that have been useful in VFX productions recently. Gridless advection and Semi-Lagrangian Mapping were introduced in [1] and [6]

### 3.1 Gridless Advection

*Gridless Advection* (GA) is a render-time technique for improving the visual clarity of densities which are undergoing advection. While traditional rendering of a coarse density grid would lead to a blurred visual appearance, GA enables the user to resolve finer details than would otherwise be possible. The advection source can be from a simulation, a suitable noise field (like vector-valued Perlin noise), or a combination of both. If the underlying velocity field is noisy, then GA has the effect of adding fine filaments and sheets to the surface of the density. When this process is repeated it is possible to produce fine-scale details that are finer than can be resolved in the ray march procedure of a volume renderer, and so the number of iterations are usually limited so that temporal aliasing (which visually appears as a shimmering effect) in the ray march is avoided. Also, in practical use, users find that applying more than four or five iterations of GA in a render adds more time and memory to a volume render than is desirable.

Implementing gridless advection only requires altering the density sample position during the ray march of a volume renderer. Given a position $\mathbf{x}$ that the ray marcher would sample for density, GA instead samples a density advected by the flow using semi-Lagrangian advection:

$$\rho_{GA}(\mathbf{x}) = \rho(\mathbf{x} - \mathbf{u}(\mathbf{x}, t)\Delta t) \tag{1}$$

It is also possible to perform multiple steps of GA via an iteration

$$\rho^{k+1}_{GA}(\mathbf{x}) = \rho^{k}_{GA}\left(\mathbf{x} - \mathbf{u}(\mathbf{x}, t^{k+1})\Delta t\right). \tag{2}$$

with $\rho^0$ being the initial density before any gridless advection. Each time the density is sampled by the volume renderer this iterative chain of steps is evaluated. For a few iterative steps the time cost is not severe and the visual benefit is great. The cost of evaluating gridless advection is linear (space and memory) in the number GA iterations since each frame requires evaluating a chain of advections and storing each velocity field in memory. GA is impractical when doing large numbers of iterations.

## 3.2 SEmi-Lagrangian MApping (SELMA)

*SELMA* is a practical compromise of GA for applications of a large number of iterations. It allows simulations to be retargeted to different initial densities and enables a trade-off between spatial detail in advection and memory/time requirements while keeping a visually crisp density field. In SELMA, the CM is sampled to a finite resolution grid which is updated by advecting the mapping function over time. At the end the density is evaluated at the mapping function.

Calculation of the SELMA map follows from recognizing that the iteration formula for GA can be written as an iteration of a map $\mathbf{X}^k(\mathbf{x})$

$$\rho^k(\mathbf{x}) = \rho^0(\mathbf{X}^k(\mathbf{x})) \tag{3}$$

$$\mathbf{X}^k(\mathbf{x}) = \mathbf{X}^{k-1}\left(\mathbf{x} - \mathbf{u}(\mathbf{x}, t^{k+1})\Delta t\right) \tag{4}$$

with the initial conditions $\mathbf{X}^0(\mathbf{x}) = \mathbf{x}$. This rephrasing of GA separates the iteration process from the density field. SELMA is the process of sampling the GA map onto a grid at each iteration. The resolution and extent of this map grid need not be the same as the velocity data, and it can be useful to use a higher resolution than the velocity field to capture much of the detail generated by GA. Hence at each iteration the mapping data consists if a gridded vector field which eliminates further need of the velocity data.

Although SELMA is a Semi-Lagrangian advection of a map, it can be applied to velocity fields generated from any type of advection scheme. Figure **??** shows the impact of using SELMA by comparing the density field produced directly from the simulation with the one produced from applying the SELMA-computed CM to the initial density field. This comparison is made for a fluid simulation using Semi-Lagrangian advection in 1 (a) and (c), and for a BFECC fluid simulation in 1 (b) and (d). Regardless of the type of underlying fluid simulation, SELMA enhances interesting visible detail in the density field.

Whereas the cost of using GA is linear with the number of iterations, SELMA has constant cost in time and memory, because the map is always kept on a grid with constant time evaluation and no velocity fields need be retained in memory. In practice using a grid in SELMA is a highly effective compromise between the extremely high but costly detail in multiple iterations of GA and the low resolution but fast sampling of a gridded density field.

## 3.3 Characteristic Map

The *Characteristic Map* (CM) $\mathbf{X}(\mathbf{x}, t, s)$ defines a mapping from points in space at a time $s$ to their advected positions at a later time $t$. The mapping is completely defined by an initial state and a time evolution equation which advects the mapping function through a velocity field $\mathbf{u}(\mathbf{x}, t)$. Phrasing the advection as a form of mapping gives greater control and flexibility when compared to traditional advection schemes. This scheme decouples the density evolution from any particular initial density field and allows the injection of more flow detail after velocity simulation is completed.

For both compressible and incompressible flows, the CM is the same quantity. But compressible flows require an additional quenching factor on the advected density related to the non-zero divergence of the velocity field. We first present the CM for incompressible flow.

### *3.3.1 Incompressible Flow*

The continuity equation describes the evolution of a density field $\rho(\mathbf{x}, t)$ over time as a function of initial conditions $\rho(\mathbf{x}, 0) = \rho_0(\mathbf{x})$, velocity field $\mathbf{u}(\mathbf{x}, t)$, and a possibly time-varying source (or sink) $S(\mathbf{x}, t)$ of density.

The continuity equation for incompressible flow,

$$\frac{\partial \rho}{\partial t} + \mathbf{u} \cdot \nabla \rho = S(\mathbf{x}, t) \tag{5}$$

has a solution in terms of the CM, which is essentially a conversion from the continuity equation to an evolution equation for the CM. The solution is composed by following a density through characteristic lines in the fluid flow:

$$\rho(\mathbf{x}, t) = \rho_0(\mathbf{X}(\mathbf{x}, t, 0)) \tag{6}$$

$$+ \int_0^t ds \;\; S(\mathbf{X}(\mathbf{x}, t, s), s) \tag{7}$$

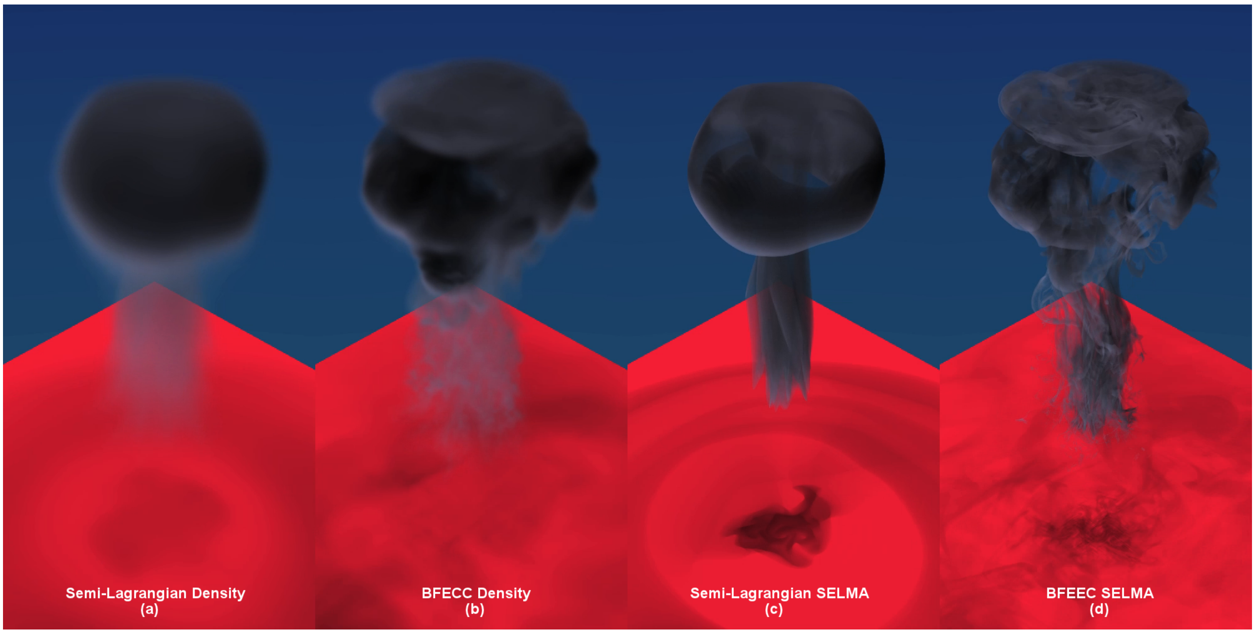

**Fig. 1** Comparison of the density field directly from a fluid simulation and from applying the SELMA-computed CM to the initial density field. (a) Density directly output from a Semi-Lagrangian fluid simulation; (b) Density directly output form a BFECC fluid simulation; (c) Density from SELMA-mapped Semi-Lagrangian fluid simulation in (a); (d) Density from SELMA-mapped Semi-Lagrangian fluid simulation in (b).

where the CM $\mathbf{X}$ satisfies the evolution equation

$$\frac{\partial \mathbf{X}(\mathbf{x},t,t')}{\partial t} + \mathbf{u}(\mathbf{x},t) \cdot \nabla \mathbf{X}(\mathbf{x},t,t') = 0. \tag{8}$$

The CM has the initial condition and "pre-condition" $\mathbf{X}(\mathbf{x},t',t') = \mathbf{x}$ for $t \le t'$, which is the identity mapping. The evolution equation has an equivalent integral form

$$\mathbf{X}(\mathbf{x},t,t') = \mathbf{x} - \int_{t'}^{t} ds\ \mathbf{u}(\mathbf{X}(\mathbf{x},s,t'),s) \tag{9}$$

which lends itself to numerical implementation. In a numerical implementation over finite-in-time steps $\Delta t$, the CM evolution equation 9 is a recursion relation, and so a straightforward algorithm for its dynamics is given by advecting the CM through the flow field

$$\mathbf{X}(\mathbf{x},t+\Delta t,t') = \mathbf{X}(\mathbf{x}-\mathbf{u}(\mathbf{x},t)\Delta t,t,t') \tag{10}$$

This form is a Semi-Lagrangian scheme for advection. Other schemes, for example BFECC, could also be used for this dynamics. For this paper, we use Semi-Lagrangian.

### *3.3.2 Compressible Flow*

Compressible flow modifies the continuity equation with an additional term proportional to the divergence of the velocity field.

$$\frac{\partial \rho}{\partial t} + \mathbf{u} \cdot \nabla \rho + (\nabla \cdot \mathbf{u})\rho = S(\mathbf{x},t) \tag{11}$$

This additional term compensates for compression and expansion effects in the flow field.The divergence term does not stop us from writing an exact solution for this problem as well (see Appendix). The divergence accumulates into the exponential form

$$T(\mathbf{x},t,t') = \exp\left\{-\int_{t'}^{t} ds\ Q(\mathbf{X}(\mathbf{x},s,t'),s)\right\} \tag{12}$$

where $Q(\mathbf{x},t) = \nabla \cdot \mathbf{u}(\mathbf{x},t)$. The explicit solution can then be written in the closed form

$$\rho(\mathbf{x},t) = T(\mathbf{x},t,0)\ \rho_0(\mathbf{X}(\mathbf{x},t,0)) \tag{13}$$

$$+ \int_0^t ds\ T(\mathbf{x},t,s)\ S(\mathbf{X}(\mathbf{x},t,s),s) \tag{14}$$

For a numerical implementation with finite-in-time steps $\Delta t$, we can evolve the state of the CM for a compressible flow field exactly as for incompressible flow, and also update $T$ using the divergence $Q$. Initializing a field $T(\mathbf{x},t,t')$ to be initially 1 everywhere, we first advect the mapping $\mathbf{X}$ in the same way as before, and also update $T$ by

$$T(\mathbf{x},t+\Delta t,t') = T(\mathbf{x},t,t')\ \exp\{-\Delta t\ Q(\mathbf{X}(\mathbf{x},t,t'),t)\} \tag{15}$$

The factor $T$ illustrates the numerical difficulty in handling the unstable nature of compressibility. Large divergence can cause an exponential growth or collapse in density. One way to combat this is to adjust the simulation time steps so that $|\Delta t\ \nabla \cdot \mathbf{u}| < 1$. This can force extremely small time steps and increase the difficulty of creating the density field. Alternatively, a production-friendly approach that avoids the problem is to clamp this exponential term

$$T_{\text{clamp}}(\mathbf{x},t,t') = \text{clamp}\left(T(\mathbf{x},t,t'), T_{min}, T_{max}\right) \quad (16)$$

This clamping function limits extreme fluctuations in density due to high flow divergence.

## 4 Applications

In this section we examine strategies for achieving three different purposes using the CM. Some purposes have more than one possible strategy that can be mixed together. Also, the strategies can be combined to achieve multiple purposes simultaneously.

### 4.1 Capturing high resolution

Generating density fields with details smaller than the grid resolution is the primary focus of GA and SELMA. The discussions in section 3 show that GA and SELMA are two numerical implementations of the CM in the special case of no sources of density and advecting only the initial density field through the CM that maps from the simulation initial time to the current moment.

To contrast the appearance and benefits of these two implementations, figure 2 shows the application of each on a frame of a simple buoyant gas simulation. Of particular note is that GA enhances detail with only a few iterations by clarifying the edges of density boundaries, and visually enhancing filaments and two-dimensional sheets, without altering the bulk distribution of the density. In this sense, GA acts to "sweeten" the density field at render time.

SELMA more strongly alters the density distribution, but the gross pattern of the low resolution density distribution remains. The filaments and sheets are further enhanced, the edges are sharper still. Because of its greater time and memory efficiency, SELMA has been applied through the entire 50 frames of velocity field that lead to this frame.

Had we applied GA for the full 50 frames, the amount of fine detail would have been more than could be rendered in a rational amount of time. For example, figure 3 shows frame 60 of a different gas simulation, rendered as the gridded simulation density and as GA advected density through the full 60 iterations. Render time of the gridded density is two minutes, while the GA advected density took 100 hours to render, partially because the ray march step is 10 times finer than the grid cell size in order to capture the finer detail. Yet even after 100 hours of fine ray marching, there are clear aliasing patterns in the GA advected image that indicate the density has features finer than the ray march could capture. This is why GA is an excellent sweetening tool, but impractical for advocating over many frames.

### 4.2 Injecting noise in the flow

The approach of injecting high resolution noise into a flow can be accomplished within the context of the CM approach. The velocity applied during the advection is the sum of the simulation velocity and a vector-valued noise function. The iterative CM construction is then

$$\mathbf{X}(\mathbf{x},t+\mathit{\Delta}t,t') \;=\; \mathbf{X}(\mathbf{x}-\mathbf{u}_{injected}(\mathbf{x},t)\mathit{\Delta}t,t,t'), \quad (17)$$

where the injected velocity is the sum of the simulation velocity and noise:

$$\mathbf{u}_{injected}(\mathbf{x},t) \;=\; \mathbf{u}(\mathbf{x},t) \;+\; \mathbf{u}_{noise}(\mathbf{x},t) \quad (18)$$

Figure 4 shows the effect of this injection, for a noise velocity composed of vector-valued perlin noise.

### 4.3 Blending flows

The CM is a natural mechanism for blending flows together. The reason is analogous to how surface displacements are used in surface rendering. Multiple surface displacements can be blended through addition, subtraction, alpha-blending, etc. The quantity $\mathbf{X}-\mathbf{x}$ is a form of 3D displacement of the flow volume that is conceptually very similar to the surface displacement, and so can be blended in like fashion. For example, if two flows produce two CMs $\mathbf{X}_1$ and $\mathbf{X}_2$, an alpha blend would be

$$\mathbf{X}_{blend} = (1-\alpha)\mathbf{X}_1 + \alpha\mathbf{X}_2 \quad (19)$$

By animating the $\alpha$ and/or varying it over the volume, a particular flow effect can be introduced and completely removed in precisely controlled ways. Figure 5 shows and example of the blending. On the left is the SELMA advected density for a low resolution flow, i.e. $\mathbf{X}_1$ is the SELMA-generated CM of the low resolution simulation. In the center, at each frame the density is advected by both the low resolution simulation and a perlin-noise-based velocity field, i.e. $\mathbf{X}_2$ is the SELMA-generated CM of the combination of the simulation velocity and perlin-noise velocity. On the right, the two CMs are blended and then applied to the density field.

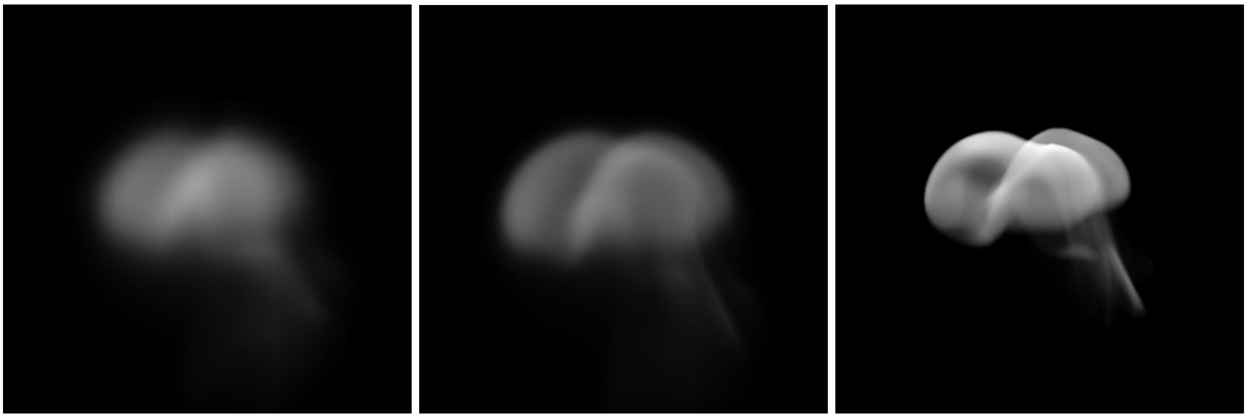

**Fig. 2** Impact of GA and SELMA on a low resolution buoyant gas simulation. Left: gridded density directly from the simulation; Middle: gridded density from the simulation with three iterations of Gridless Advection; Right: the initial density field mapped to the same frame as the others with SELMA. The simulation grid is 32X128X32, and the SELMA grid is 128X512x128. This is frame 50 of a 120 frame simulation, with no subframe stepping. Volume rendering conditions are identical for the three images.

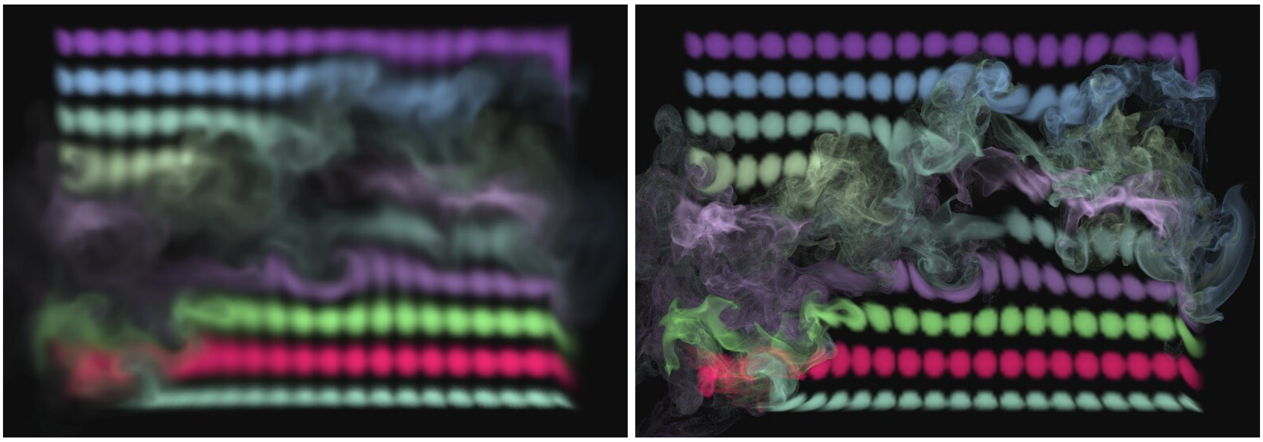

**Fig. 3** Demonstration of the extreme detail achievable with Gridless Advection. Left: Render of the gridded simulation density. Ray march step is equal to the cell size, render time is two minutes. Right: Render of the density from gridlessly advocating all of the frames. Ray march step is one tenth of the cell size, render time is 100 hours, too long for practical application.

## References

1. M. Wrenninge, N.B. Zafar, J. Clifford, G. Graham, D. Penney, J. Kontkanen, J. Tessendorf, A. Clinton, in *Volumetric Method in Visual Effects Course Notes, ACM SIGGRAPH 2010* (2010)
2. M. Lentine, J.T. Grétarsson, R. Fedkiw, "An unconditionally stable fully conservative semi-Lagrangian method", J. Comput. Phys. **230**, 2857 (2011)
3. R. Bridson, R. Fedkiw, M. Muller-Fischer, in *Fluid Simulation Course Notes* (ACM, New York, NY, USA, 2006), SIGGRAPH '06, pp. 1–87
4. J. Douglas, Jr., C.S. Huang, F. Pereira, The modified method of characteristics with adjusted advection for an immiscible displacement problem. Tech. rep., Dept. of Mathematics, Purdue University (1998)
5. F. Neyret. Advected textures (2003). URL http://www-evasion.imag.fr/Publications/2003/Ney03
6. S. Hasegawa, J. Iversen, J. Tessendorf, in "I Love It When A Cloud Comes Together*", Proceedings, ACM SIGGRAPH 2010* (2010)

## A Derivation of the Density Evolution Solution

For the incompressible evolution equaton, the solution has the form

$$\rho(\mathbf{x},t) = \rho^0(\mathbf{X}(\mathbf{x},t,0)) + \int_0^t dt'\ S(\mathbf{X}(\mathbf{x},t,t'),t')\ .$$

Substituting this into the evolution equation for incompressible flow, the vector field $\mathbf{X}$ must satisfy the evolution:

$$\frac{\partial \mathbf{X}(\mathbf{x},t,t')}{\partial t} \ +\ \mathbf{u}(\mathbf{x},t)\cdot\nabla\mathbf{X}(\mathbf{x},t,t') \ =\ 0$$

For the homogeneous (no sources) compressible case, dividing by the density gives the equation

$$\frac{1}{\rho}\left(\frac{\partial \rho}{\partial t} + \mathbf{u}\cdot\nabla\rho + (\nabla\cdot\mathbf{u})\rho\right) \ =\ 0$$

Rephrased in terms of the log of the density the equation is

$$\frac{\partial \ln\rho}{\partial t} + \mathbf{u}\cdot\nabla\ln\rho + (\nabla\cdot\mathbf{u}) = 0$$

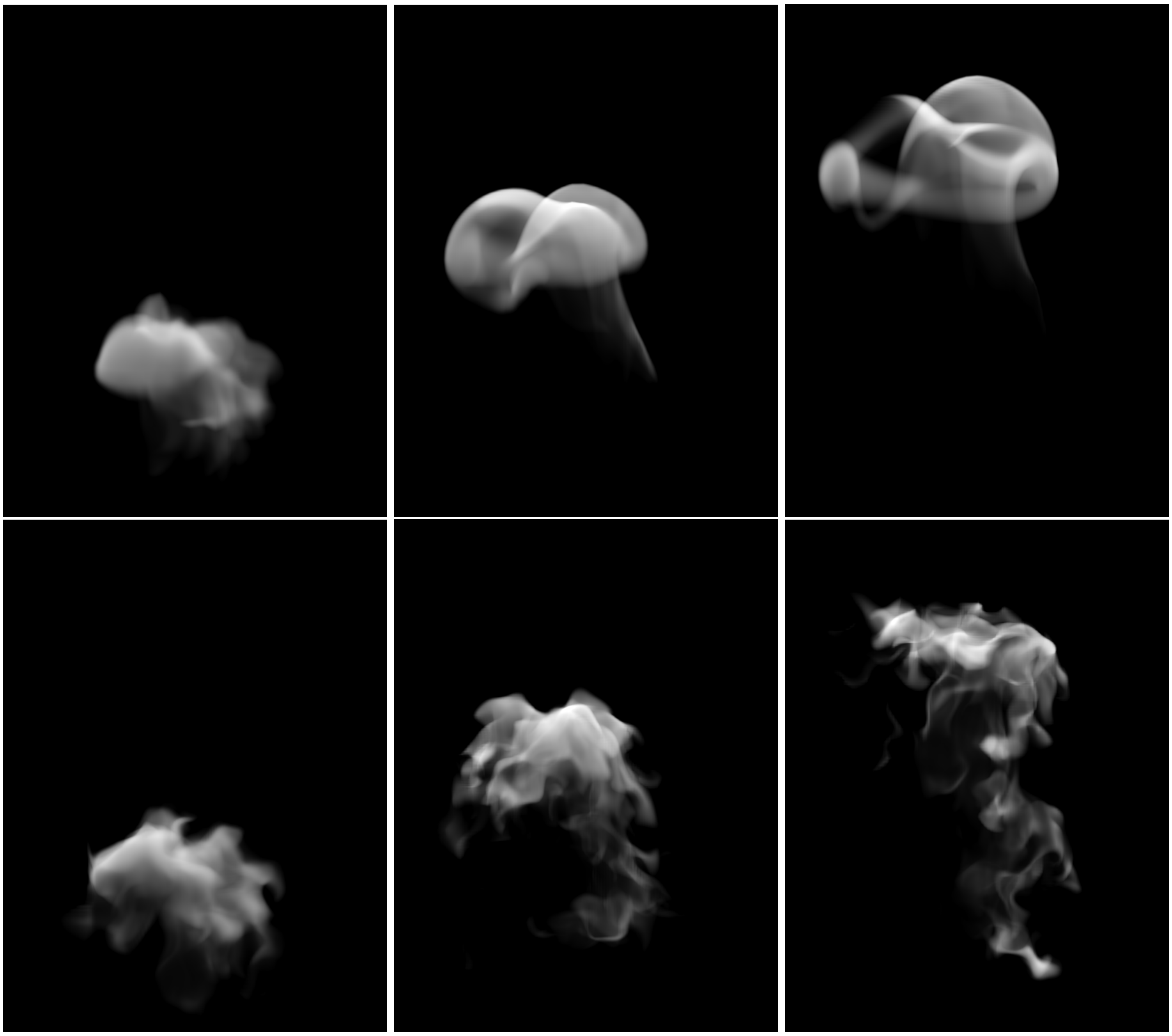

**Fig. 4** Demonstration of noise injected into the CM. Top row: Sequence of frames in the low resolution gas simulation, advected using SELMA on the CM. Bottom row: The same sequence of frames of the simulation, with additional vector-valued perlin noise injected into the CM at each frame prior to sampling onto the SELMA grid.

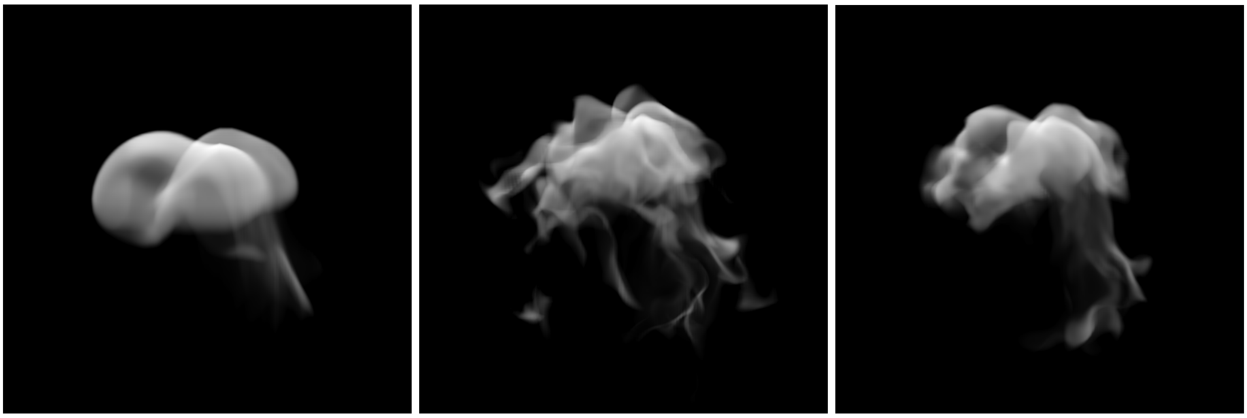

**Fig. 5** Demonstration of blending flows using the CM. Left: The low resolution gas flow with SELMA advection. Middle: The same flow, but with a perlin-noise-based velocity injected in the SELMA map at each frame to evolve additional detail. Right: Density field from blending the Characteristic Maps from the two other cases.

which is identical in form to the inhomogenous incompressible evolution equation, with the velocity divergence acting as the source. Using that form

$$\ln \rho(\mathbf{x}, t) \;=\; \ln \rho^0(\mathbf{X}(\mathbf{x}, t, 0)) - \int_0^t dt' \; Q(\mathbf{X}(\mathbf{x}, t, t'))$$

where $Q(\mathbf{x}, t) \equiv \nabla \cdot \mathbf{u}(\mathbf{x}, t)$. When external sources are used, the compressible solution can be built similar to that for the incompressible case. Notice that in the homogeneous case, the impact of compressibility is to insert the factor $T$ with the density. Since the source term $S$ acts as an injection of density, it is reasonable to expect that a factor of $T$, as defined in equation 12, would accompany $S$. In fact, it is quickly verified through explicit differentiation that

$$\begin{aligned}\rho(\mathbf{x}, t) = {} & T(\mathbf{x}, t, 0)\; \rho^0(\mathbf{X}(\mathbf{x}, t, 0)) \\ & + \int_0^t dt' \; T(\mathbf{x}, t, t') \; S(\mathbf{X}(\mathbf{x}, t, t'), t')\end{aligned}$$

is the exact solution.