Inverse Modeling of Incompressible Gas Flow in Subspace

X. Zhai\textsuperscript{1}, F. Hou\textsuperscript{1,2}, H. Qin\textsuperscript{3}, and A. Hao\textsuperscript{1}

\textsuperscript{1}Beihang University, China
\textsuperscript{2}Nanyang Technological University, Singapore
\textsuperscript{3}Stony Brook University, America

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure1}
\caption{Results on a simulated “ejection” dataset. (a) The existing flow; (b) Sample Empirical Mode Decomposition (EMD) bases in different frequency scales for modal reduction (mapped to HSV color space for visualization); (c) Re-simulation of the flow from physical inverse modeling; (d) Flow edited in subspace by adding a torus boundary; (e) Flow edited in the original space by applying a complex boundary.}
\end{figure}

Abstract

This paper advocates a novel method for modeling physically realistic flow from captured incompressible gas sequence via modal analysis in frequency-constrained subspace. Our analytical tool is uniquely founded upon Empirical Mode Decomposition (EMD) and modal reduction for fluids, which are seamlessly integrated towards a powerful, style-controllable flow modeling approach. We first extend EMD, which is capable of processing 1D time series but had shown inadequacies for 3D graphics earlier, to fit gas flows in 3D. Next, frequency components from EMD are adopted as candidate vectors for bases of modal reduction. The prerequisite parameters of the Navier-Stokes equations are then optimized to inversely model the physically realistic flow in the frequency-constrained subspace. The estimated parameters can be utilized for re-simulation, or be altered toward fluid editing. Our novel inverse-modeling technique produces real-time gas sequences after precomputation, and is convenient to couple with other methods for visual enhancement and/or special visual effects. We integrate our new modeling tool with a state-of-the-art fluid capturing approach, forming a complete pipeline from real-world fluid to flow re-simulation and editing for various graphics applications.

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

submitted to COMPUTER GRAPHICS Forum (10/2015).
1. Introduction

Since early 1990’s, various gas/fluid simulation methods for graphics and animation have been developed, and the majority of existing techniques are founded upon solving the incompressible Navier-Stokes equations or their possible variants. Along with the rapid advancement in graphics, computational fluid dynamics, and relevant engineering disciplines, tremendous fluid-capture methods have been proposed in recent years, such as Particle Image Velocimetry (e.g., Tomo-PIV [ESWVO06]), time-resolved Schlieren systems [AIH+08], surface modeling [DLJY11, YLY+12], optical flow based methods [LS08, GKH+12], etc. Strongly inspired by the most recent works [WLC+09, LPS+13, GKH+14] at-tempting to bridge the gap between simulation and capturing, in this paper we propose to seek novel strategies to further integrate captured data and generated flow sequences.

Our inverse-modeling technique aims at generating gas sequences that best match the existing/captured flows by simulation. Different from previous works on fluid capturing [GKH+14] and guiding control [NCZ+09], we fulfill it by inferring the prerequisite parameters of the flows according to the incompressible Navier-Stokes equations, so the flows can be naturally reproduced and edited through simulation without any prior knowledge on those parameters.

The parameters estimation step, as known as the inverse-modeling step, is the key component of our work. Given that the effect of viscosity and external forces are hard to differentiate, we model the flows by an optimization process to best fit the formulation. The inverse-modeling step is conducted in subspace for acceleration. Flows after this step can be physically modified towards desired behaviors through multiple approaches, for instance, various driving forces, thickened air, varied boundaries, etc. More complex modifications, e.g., non-linearities like vorticity confinement, can be implemented in the original space to combat the limited degree of freedom in subspace.

Traditionally, the subspace of flow fields is constructed by PCA bases [TLPO06], which are far less meaningful in frequency domain and may have inter-frequency energy mixing in some circumstances. Instead, we seek new bases via EMD [HSL+98], which separates flows into frequency bands to better capture the intrinsic features. The original EMD algorithm is proposed for 1D time series and 2D signal processing [HS05], and proves ideal for handling non-stationary and non-linear signals. However, the usage of EMD in fluid area is far from being adequate, since previous techniques only serialize the 3D flow field with space-filling curves [GLR13, RLL+13]. We ameliorate EMD to better expose the intrinsic features of fluid flows.

Most fluid capturing methods that provide velocity fields of the flows, such as optical-flow based methods, are compatible with our technique. We integrate our tool with a state-of-the-art fluid capturing method from Gregson et al. [GKH+12, GKH+14]. The real-world fluid is captured with Stochastic Tomography [GKH+12], then the velocity fields are reconstructed by the fluid tracking algorithm [GKH+14]. We analyse the velocity fields with our proposed technique to acquire re-simulated sequences, forming a complete pipeline from real-world fluid to flow re-simulation and flow editing for various graphics applications. These lead to the main contributions of this paper:

1. We ameliorate EMD algorithm for fluid analysis. The new algorithm can better expose the intrinsic features of fluid flows compared with previous techniques.

2. We devise an analytical tool based on EMD and modal reduction, capable of inverse modeling gas sequences and apply physical editing on existing flows.

3. We integrate our technique with a state-of-the-art fluid capturing method, offering a pipeline to bond the fluid simulation in graphics with real-world flows.

2. Related Work

Our work is closely relevant to fluid simulation, fluid editing, model reduction and Empirical Mode Decomposition. We briefly review them in the following categories.

Fluid Simulation aims at calculating the flow motion by solving the incompressible Navier-Stokes equations. The grid-based fluid solver was first brought into graphics by Foster and Metaxas [FM96], and continued to retain its popularity with the unconditionally stable solver [Sta99]. Since then many techniques have been proposed to eliminate the numerical dissipation and to enrich the details [FJS01, ZB05, KLLR05, SFK03], and boosted by McNamara et al. [MTPS04]. Fattal et al. [FL04] and Shi et al. [SY05] used target shapes to guide smoke, while Rasmussen et al. [REK04] applied soft or hard control over liquids by introducing particles. Thureen et al. [TKPR09] achieved fluid guiding in SPH and LB-M. Other techniques apply fluid control to produce finer results out of a coarse guidance. Nielsen et al. [NCZ+09] proposed an optimization scheme to ensure the consistency between a high-resolution smoke simulation and a low-resolution velocity field. Yuan et al. [YCYZ11] yielded high-resolution smoke animation following the low-resolution version by Lagrangian Coherent Structure, and Nielsen et al. [NB11] guided high-resolution liquid simulation by its

submitted to COMPUTER GRAPHICS Forum (10/2015).
Fluid Capturing has gained abundant attention in graphics recently. Capture measurements roughly include surface modeling [DLIY11, YJLY12], density estimation [AIH+08, GKH12] and velocimetry [ESWVO06, LS08]. Recent researches exhibit a trend of applying physical constraints or guidance rather than purely relying on capturing methods. Wang et al. [WLZ09] utilized physically-based surface optimization to reconstruct water surface from videos, while Li et al. [LPS13] acquired water surface from a single viewpoint through shallow water model. Most lately, Gregson et al. [GIT+14] presented a proximal optimization framework that related the captured density data to corresponding velocity fields. Inspired by their work, we propose a novel method aiming at inverse modeling from existing velocity fields to the simulation progress that produce them, which can also be regarded as an analytical measurement capable of reusing the intrinsic properties of fluid.

Model Reduction has been widely used in graphics, including non-linear deformation [JF03] and Finite Element Method [B05]. Treuille et al. [TLP06] first introduced model reduction with PCA bases to fluid simulation and generated real-time flows. Subsequently, Wickie et al. [WST09] generalized this technique to modular tiles to fit more complicated scenes. DeWitt et al. [DWLF12] replaced PCA eigenvectors with Laplacian eigenfunctions which are suitable for arbitrary flows. Kim and Delaney [KD13] proposed cubature approach to support semi-Lagrangian advection and other non-linearities. The inverse modeling step in this paper produce these, which can also be regarded as an analytical measurement capable of reusing the intrinsic properties of fluid.

Empirical Mode Decomposition (EMD) was first introduced by Huang et al. [HSL+98] for temporal signal processing. Its data-dependent nature is ideal for analysing non-stationary data. In graphics and image processing, researchers have extended EMD to handle 2D or 3D signals. Linderhed [Lin11] presented a 2D EMD method to compress images. Wang et al. [WHZQ15] applied EMD for 3D geometry processing. The difficulty of applying EMD on multi-dimensional data lies in constructing the envelopes. Wu et al. [WHC11] proposed Multi-dimensional Ensemble EMD that partitioned data into 1D slices for decomposition. To interpolate the exact envelopes, Damerval et al. [DMF05] used Delaunay triangulation and piecewise cubic polynomial interpolation. Xu et al. [XLLR06] presented a mesh fitting method, and Linderhed [Lin11,Lin09] made comparisons on existing solutions and suggested using thin-plate spline for smooth result with continuous second derivative. Rehman and Mandic [RM10] even extended EMD to process vector values by projections on hyperspheres. Recently, EMD has been adapted into fluid simulation [GLRH13] and modulation [RLL+13]. Their sifting operations are achieved by serializing 3D data to 1D with a space-filling curve, and then using 1D EMD algorithm. However, this method loses connectivity information in 3D, resulting in spurious noises. In this paper, we apply a fully 3D EMD variant that can eliminate such deterioration, and hence is much more appropriate for flow field. We further use the resulting components as candidate bases in model reduction.

3. Empirical Mode Decomposition on Flows

EMD, originally designed for analyzing time-varying signals in 1D [HSL+98], needs to be extended to 3D and other modifications for processing flow fields. We briefly review the standard EMD method in Section 3.1, Section 3.2 explains the strategy to apply EMD to 3D signals on Eulerian grids, and Section 3.3 discusses the EMD method further extended to process 3D flow fields. The modified EMD algorithm for 3D flows is listed in Algorithm 1.

### 3.1. Conventional EMD

The functionality of EMD is to decompose a signal \( x(t) \) into a finite number of Intrinsic Mode Functions (IMFs) \( c_i, i = 1, 2, ..., q \) and a residue \( r_q \), which characterize the features of the signal on different scales:

\[
x(t) = \sum_{i=1}^{q} c_i + r_q.
\]

Conceptually, the IMFs and the residue can be considered as the counter-part of “frequency components” in other decomposition methods. However, in comparison with traditional Fourier and wavelet analysis, their frequencies are not pre-defined but data-dependent, and they can offer spatially-varying “instantaneous frequencies”. This self-adaptive nature provides the suitability for handling high-frequency non-stationary and non-linear signals [HSL+98,WHC11]. The EMD algorithm that extracts IMFs through "sifting" is documented in [HSL+98]. In each iteration, an IMF representing features with higher frequency ranges is extracted, leaving the lower frequency part as residue or input of the subsequent iterations. Thus, IMFs are arranged spontaneously in the order of their frequency ranges.

### 3.2. EMD on 3D Data

The original EMD method is powerful to deal with scalar data defined in 1D space, but needs modifications to fit the velocity fields of flows, three-component vector signals in 3D space. Although vector signals can be processed by multivariate EMD [RM10], its overhead rejects the feasibility, since Eulerian grids for gas simulation usually contain millions of cells. Recall that in traditional fluid solvers, three components of the velocity fields are often treated as individual channels and handled respectively. Likewise, we also carry out the sifting procedure on each scalar component,
and similar solution of applying EMD on vector fields can be found in [GLRH13, RLL⁺13]. Experiments demonstrate that the results are still adequate with this simplification.

Now we explain EMD for 3D signals. Since we are only concerned with discrete 3D signals, the extrema points are simply extracted by comparing the candidate data point with its nearest 26-connected neighbors. The challenge lies in fitting a space to the 3D scattered data points representing the extrema points (Line 8 in Algorithm 1). In 2D case, Linderhed [Lin11, Lin09] compared existing solutions, including Directional EMD, triangle-based cubic spline interpolation and thin-plate spline (TPS) interpolation, and suggested using TPS interpolation for smooth result with continuous second derivative. Conceptually, TPS is the generalization of the natural cubic splines in 1D. The construction is based on choosing a function \( f \) that exactly interpolates the data points and minimizes the bending energy,

\[
E[f] = \int_\mathbb{R}^3 \|D^2 f\|^2 dx
\]  

(2)

where \(D^2 f\) is the matrix of second-order partial derivatives of \( f \) and \(\|D^2 f\|^2\) is the sum of squares of the matrix entries. The implementation details of TPS can be found in [Ebe02]. Given that the huge computational costs prevents previous works from using TPS interpolation, we resort to CUDA for acceleration and obtain acceptable efficiency.

The boundary constraints are more important in 3D than in 1D/2D cases. The extrema points are very sparse and since the interpolation methods only interpolate between points, the boundary need special care. Similar to Linderhed [Lin11], we add extra data points at the boundaries to the set of extrema points (Line 7 in Algorithm 1). The extra points are placed at the corners and some additional points at the boundary equally spaced between the corners. Without these extra points, the areas not covered by the interpolation yield bad results in the sifting process.

Our method has the advantage of being fully 3D, which retains the global topologies in the original signal. A comparison between our method and previous EMD technique on flows [GLRH13, RLL⁺13] can be seen in Fig. 2. The method we compare against is implemented by first serializing the 3D signals with Hilbert space-filling curve and then applying 1D EMD algorithm. We confine each method to only compute 2 IMFs and the rest velocity is directly treated as residue. As shown in the comparison, our method manages to arrange properties in different frequency bands to their relevant frequency components, and generate high-consistency, low-noisy results, which is superior to previous technique mainly because of the global topologies it retains.

![Figure 2](https://example.com/figure2.png)

**Figure 2:** EMD methods comparison at the 29th frame of the captured “bloom” dataset. (a) is the original flow field, (b) is the result from 1D EMD with Hilbert space-filling curve, and (c) is the result of our 3D EMD. For visualization, the 3D vectors are mapped to HSV color space, and the central slice of the 3D domain is displayed. Our 3D EMD succeeds in extracting high-consistency, low-noisy frequency features, while the original method lacks global topologies and induces spurious noises.

### 3.3. Flow Field Mode Decomposition

Making use of the 3D EMD method, the velocity field \( \mathbf{u} \) can be decomposed as

\[
\mathbf{u} = \sum_{i=1}^q \mathbf{u}_i + \mathbf{u}',
\]  

(3)
where \( u^i, i = 1, 2, ..., q \) are the IMFs carrying flow features from local to global scales, and \( u' \) is the residue after those IMFs have been extracted.

Nevertheless, to better fit fluid applications, especially the downstream modal reduction, we have to make a little bit more modification on the EMD pipeline. Treuille et al. [TLP06] specified that the velocity fields used to generate bases should meet two important properties: being free of divergence and satisfying the free-slip boundary condition. We fulfill these requirements by restricting the envelopes to the free-slip boundary condition (Line 9 in Algorithm 1), and executing pressure projection to the pre-sifted IMFs (Line 17 in Algorithm 1). Since the input velocity fields satisfy these properties, the final residues should also do.

The IMF stopping criterion we adopt is the criterion suggested in [HSL98], which sets an upper threshold \( \sigma_{\text{max}} \) of the standard deviation between two consecutive sifting procedures and a maximum number of iterations \( k_{\text{max}} \). During the EMD step, we do not repeatedly carry out the algorithm to the very end, because after several sifting processes the flow field is already smooth enough as a total low frequency signal and further decomposition becomes dispensable. Instead, the number of IMFs \( q \) is fixed to a constant, and the residue signal after \( q \) sifting processes is directly saved in \( u' \).

In all experiments of this paper, we set \( q = 2 \), so each velocity field \( u \) will have three frequency components (\( u^{\text{low}}, u^{\text{mid}}, \) and \( u^{\text{high}} \)) after EMD step.

### 4. Physical Inverse Modeling on Flows

The EMD method introduced in Section 3 is a powerful tool that separates flow fields into various frequency components, which hereinafter refer to the outcome of EMD including IMFs and residue. In this section, we propose a inverse-modeling method based on subspace analysis to re-simulate or generate new sequences with varying conditions from a known gas flow.

The basic idea is to apply EMD on each frame of the original flow, reuse the features carried with frequency components by modal reduction, and estimate the parameters implied within the original gas flow in subspace. After estimation, the flow can be re-simulated or modified towards desired behaviors. We concisely introduce modal reduction for fluid flows in Section 4.1, followed by the algorithm to estimate flow parameters in the incompressible Navier-Stokes equations in Section 4.2, and flow editing schemes are finally discussed in Section 4.3.

#### 4.1. Modal Reduction

Modal reduction means representing a high dimensional vector and its time evolution in a lower dimensional space. Specifically, a flow field \( u \in \mathbb{R}^m \) can be represented as \( r \in \mathbb{R}^n \), where \( m \) is independent of the spatial resolution \( n \) and \( m \ll n \). Usually, the \( m \)-dimensional subspace is linear, so an orthonormal projection matrix \( B \) can be found to connect the two spaces \( u = Br \), with equality if \( u \) lies exactly in the subspace spanned by \( B \). Linear differential equation \( \dot{u} = Mu \) can also be projected into subspace using Galerkin projection \( \dot{r} = B^TMB\dot{r} \), by which the incompressible Navier-Stokes equations can be solved within in subspace. For more information about fluid simulation in subspace, please refer to [TLP06].

The process to find bases \( B \) is somewhat tricky. Previous methods [TLP06, KD13] apply PCA to compress a set of example velocity fields, and the first \( m \) eigenvectors are set as columns in \( B \). However, we have found that in some circumstances this method fails to correctly capture the features in all frequencies, especially the high-frequency details, since high-frequency vectors usually have small eigenvalues and are excluded from bases \( B \). Consequently, high-frequency parts of the flow can’t be represented within the subspace, and the kinetic energy induced by high-frequency behaviors is transferred from fine-to-coarse scales through the bases in the re-simulation. This inter-frequency energy mixing can result in an overall falsely enlarged energy in all frequency bands, see Fig. 5.

We solve this problem by taking advantages of EMD, fulfilling its potential in frequency domain. First, we create a set of example fields for each frequency component, \( u^{\text{freq}} = \{u_i^{\text{freq}}\} \), where \( \text{freq} \) can be \( \text{low, mid, or high} \). Subsequently, PCA is performed on each set for compression, and the first several eigenvectors of each set are selected into the bases \( B \). Finally, the combined bases \( B \) are orthonormalized by Gram-Schmidt process. Note that after orthonormalization the basis functions will no longer be purely \( \text{low, mid, or high} \) frequency, but this is totally fine since the subspace remains the same. Essentially, the PCA-based methods select basis functions regardless of their frequency, while our method ensures that the basis functions are chosen from all frequency bands, hence the EMD-based bases are free from the inter-frequency energy mixing. A comparison between PCA bases and EMD bases can be seen in Fig. 5, and the corresponding frequency spectra are presented in Fig. 6. The composition of bases \( B \) from each frequency band can be adjusted to accomplish style control, and typically \( \frac{m}{n} \) vectors from each set is used for general purpose.

#### 4.2. Parameters Estimation

Having the reduction bases \( B \), now we present the algorithm to estimate the parameters in fluid equations. The incompressible Navier-Stokes equations are as follows:

\[
\dot{u} = -(u \cdot \nabla)u - \nabla^2 u + \nabla p + f, \quad (4)
\]

\[
\nabla \cdot u = 0, \quad (5)
\]

where \( p \) denotes the pressure, \( \nu \) the viscosity, and \( f \) the external forces. When integrating, the pressure is solved by a
Poisson equation $\nabla^2 p = \nabla \cdot \mathbf{u}$, therefore only the viscosity and external forces are left unknown. The purpose of inverse modeling is to estimate these parameters and recreate a simulated flow sequence according to Navier-Stokes equations. Given the large size of the velocity field $\mathbf{u}$ and the time-consuming pressure projection step, the estimation is hard to implement within acceptable time without modal reduction.

Similar with Treuille et al. [TLP06], we discretize the advection term using finite difference as $\mathbf{u} = A_0 \mathbf{u}$, and the diffusion term as $\mathbf{u} = \nabla p$, where $A_0$ is dependent on $\mathbf{u}$ but $D$ is not. In subspace, each vector in the bases has its own advection matrix $A_0$, and the total advection term can be denoted by $A = \sum r_i A_0$. The corresponding linear differential equation of Navier-Stokes equations in subspace is

$$\mathbf{f} = \hat{A} \mathbf{f} + \nabla \hat{D} \mathbf{f} + \hat{\mathbf{f}},$$

where $\hat{A}$ is the contracted advection tensor $\hat{A} = \sum r_i B_i^T A_0 B_i$, $\hat{D}$ is the diffusion matrix $\hat{D} = B_i^T D B_i$, and $\hat{\mathbf{f}}$ is the reduced external force $\mathbf{f} = B_i^T \mathbf{f}$. The pressure term and the divergence constraint vanish in subspace, because all vectors in the bases are inherently free of divergence. So it is with the boundary condition. The reduced matrices $\hat{D}$ and $B_i^T A_0 B_i$ are supposed to be precomputed for acceleration.

Suppose an inviscid flow ($\nu = 0$), the external forces can be easily calculated from Eq. 6, and vice versa. However, if the viscosity and external forces exist at the same time, it is hard to distinguish. Our solution is to assume that the external forces are as small as possible throughout all frames, that is, to minimize the objective function

$$\arg\min_{\nu} \sum_{\mathbf{r}_i} \| B(e^{-\Delta t (\hat{A} + \nabla \hat{D})} \mathbf{r}_{i+1} - \mathbf{r}_i) \|^2, \text{ s.t. } \nu \geq 0$$

for appropriate viscosity, where $F$ denotes the set of frames. This method essentially maximizes the viscous effect, which is true in most cases, and yields nice results. Afterwards, the external force of each frame can be calculated through Eq. 6.

For the flows on which we have prior knowledge about their inflow source, denoted by $S$, an ameliorant is to exclude the external forces near $S$ (manually picked with no need to be exact) and minimize the magnitude of external forces in other regions. In practice, we remove the rows corresponding to the vicinity of $S$ from bases $B$. For performance consideration, down-sampling is adopted both spatially and temporally. The bases $B$ is down-sampled by removing rows randomly, and we select a random subset from $F$ to apply the optimization. The optimization is solved by Interior Point Method. Despite the large size of $B$, this optimization can be very fast since only $B^T B$ is used, which is rather small and can be pre-computed. In case of deterioration by down-sampling, this procedure can be repeated several times to average the estimated viscosity.

### Table 1: Default parameters used in our experiments.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$q$</td>
<td>2</td>
</tr>
<tr>
<td>$\sigma_{\max}$</td>
<td>0.2</td>
</tr>
<tr>
<td>$k_{\max}$</td>
<td>10</td>
</tr>
<tr>
<td>$m$ for the captured datasets</td>
<td>75</td>
</tr>
<tr>
<td>$m$ for the “ejection” dataset</td>
<td>90</td>
</tr>
</tbody>
</table>

#### 4.3. Flow Editing

Flows after inverse modeling can be reproduced by re-simulation, and modified towards desired behaviors through multiple approaches, for instance, various driving forces, thickened air, varied boundaries, etc. Such subspace editing approaches are detailed in [TLP06, KD13]. More complex modifications, e.g., non-linearities like vorticity confinement, can be implemented in the original space to combat the limited degree of freedom in subspace.

With the frequency-aware property of EMD, style control on the modeled flows can also be achieved. When inverse modeling, the composition of bases $B$ can be adjusted by selecting unequal eigenvectors from different frequency bands. More vectors help capture the intrinsic features more precisely within a frequency band, and fewer vectors smooth out the specific effect. Inter-frequency energy mixing is less conspicuous with EMD-based bases, hence changing the number of vectors from a certain band will not affect the behaviors of other frequency bands. As a matter of fact, this method yields different results by altering the subspace. However, it should be noted that removing columns from the low frequency band is usually not recommended, since the overall flow may well be altered completely if the subspace is overly shrunk that way.

#### 5. Experimental Results

In this section we provide numerical evaluations as well as several applications of our method. We implement our EMD algorithm with CUDA for acceleration. The experiments run on a PC with Geforce GTX970 GPU and Core i7 CPU, and our results are rendered by POV-Ray.

Several applications of our technique are documented below, including re-simulating and flow editing within or without subspace with convincing effects. The captured flow datasets are from Gregson et al. [GKHH12], and recovered by fluid tracking method documented in [GIT*14]. A simulated gas ejection is also included as ground truth in our experiments. Table 1 shows the default parameters used in our experiments and Table 2 lists the experimental performance, indicating the high efficiency of the proposed inverse-modeling and editing technique.

Table 3 verifies the viscosity estimation in Eq. 7. We empirically set the inflow source $S$ on the simulated “ejection”...
Table 2: Time performance (in seconds) of our experiments.

<table>
<thead>
<tr>
<th>Scene</th>
<th>#(Resolution, Frames)</th>
<th>EMD</th>
<th>Modal Reduction</th>
<th>Inverse Modeling</th>
<th>Re-simulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ejection (Fig. 1)</td>
<td>(160 × 80 × 80, 120)</td>
<td>816.197</td>
<td>935.108</td>
<td>1.803</td>
<td>0.710 (∼169 fps)</td>
</tr>
<tr>
<td>Bloom (Fig. 3)</td>
<td>(75 × 165 × 75, 60)</td>
<td>1250.441</td>
<td>492.536</td>
<td>0.602</td>
<td>0.213 (∼281 fps)</td>
</tr>
<tr>
<td>Smoke (Fig. 4)</td>
<td>(75 × 165 × 94)</td>
<td>3665.044</td>
<td>490.953</td>
<td>1.000</td>
<td>0.334 (∼281 fps)</td>
</tr>
</tbody>
</table>

Table 3: Viscosity estimation.

<table>
<thead>
<tr>
<th>Scene</th>
<th>Real viscosity</th>
<th>Estimated viscosity #1</th>
<th>Estimated viscosity #2</th>
<th>Estimated viscosity #3</th>
<th>Estimated viscosity #4</th>
<th>Estimated viscosity #5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ejection</td>
<td>10.000</td>
<td>10.200</td>
<td>3.674</td>
<td>1.927</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bloom</td>
<td>–</td>
<td>10.210</td>
<td>3.665</td>
<td>1.913</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Smoke</td>
<td>–</td>
<td>10.415</td>
<td>3.678</td>
<td>1.901</td>
<td>10.308</td>
<td>3.641</td>
</tr>
</tbody>
</table>

Fig. 1 displays the results on a simulated “ejection” flow. The flow in Fig. 1(a) is inversely modeled using our technique, and re-simulated as seen in Fig. 1(c). The flow is generally well-recovered, except some slight disagreement lying in the high frequency area between the existing flow and the re-simulated one, caused by the discrepancy of their numerical integration methods. The existing flow is simulated by semi-Lagrangian method, while in subspace the differential equation has analytical solution. Kim and Delaney [KD13] reported that the distinction of the integration methods may cause clashes on the high frequency modes. Nonetheless, this flaw is weakened if we apply our technique on captured datasets, where there is no discrepancy of integration methods, hence no disagreement is found in those experiments. An additional torus barrier is placed in the domain as shown in Fig. 1(d) to test the ability of our subspace technique to handle editing on boundary, and the flow interacts correctly with the barrier. To handle complex boundary editing, the subspace lacks the degree of freedom, but it is still convenient to recover the flow in its original space and apply the changes, as shown in Fig. 1(e).

Fig. 3 presents the results on a captured “bloom” flow, which involves dye being poured into still water. The flow is rich of turbulence in the top half of the domain, but becomes stable in the bottom half since the buoyancy and the diffusion effect of the surrounding fluid dampen those details. The re-simulation result is exhibited in Fig. 3(b), which is basically the same as the original flow without noticeable distinction. We edit this captured flow by doubling its viscosity and adding an extra downward impulse in subspace, and the results are shown in the bottom row. In Fig. 3(c) the surrounding fluid becomes more viscous, consequently the flow loses its turbulence quickly as soon as it enters the domain. In contrast, the flow in Fig. 3(d) turns more energetic and presents more details throughout the entire domain.

Fig. 4 shows the results on a captured “smoke” flow, which has abundant high-frequency energy caused by buoy-
Figure 3: Experiments on the captured “bloom” dataset. (a) Existing flow; (b) Re-simulation of the inversely modeled flow with our technique; (c) Flow edited by doubling the viscosity; (d) Flow edited by adding extra impulse.

Figure 4: Experiments on the captured “smoke” dataset. (a) Existing flow; (b) Re-simulation of the inversely modeled flow; (c) Re-simulation with PCA bases that contain identical number of columns as our EMD bases; (d) Effect of vorticity confinement, $\varepsilon = 0.2$; (e) Style control by removing high-frequency bases; (f) Style control by removing middle-frequency bases.

Figure 7: Comparison of the boundary modification on the captured “smoke” dataset. (a) Result of [GIT*14]; (b) Our result.
Our method can be seen as a follow-up of [GIT* 14], but we still provide a comparison with their method in the ‘domain change’ application using the captured “smoke” dataset, as demonstrated in Fig. 7, where a metal ball is placed in the domain blocking the smoke from rising. Noticeable difference can be found in the results although the same settings are used to implement both methods. The main cause is that our method achieves physical simulation and editing with the estimated flow parameters, while theirs can only convey the density passively. Their method advects the flow up to a desired time-frame, after which it switches to a dynamic setting considering the new boundary, which is non-physical to be rigorous. All other stylistic modifications that [GIT* 14] brings about, such as resolution enhancement and guided simulation, even though beyond the topic of this paper, can be accomplished with our method in a more rigorous manner that strictly follows the governing equations.

6. Conclusion and Discussion

Our main goal in this paper is to make new attempts to remove the perceived barrier between fluid capturing and pure fluid simulation in graphics applications via a powerful analytical tool. This paper has detailed a data-driven approach for fluid analysis based on EMD and modal reduction, capable of inversely modeling existing gas flows to discover the simulation parameters and editing them towards desired behaviors, hence unifying fluid capturing and fluid simulation through tight and seamless coupling. Our extensive experiments have illustrated that our extended EMD for flow field, accelerated by CUDA, is much more adequate for fluid decomposition compared with previous space-filling curve scheme. Making use of this EMD algorithm, we have decomposed the 3D gas flow into frequency components representing the intrinsic features, which are also candidate basis vectors for modal reduction. In subspace, we have devised a functional optimization method to isolate the effect of viscosity and external forces, aiming at performing parameters estimation and modifying the gas flow by reusing the existing characteristics extracted from the original flow. We have demonstrated that our EMD bases are superior to the traditional PCA bases in avoiding inter-frequency energy mixing, as well as bringing the flexibility of style control on the modeled flow. With the inverse-modeling technique, physical editing on the existing flows is achievable. Finally, we have integrated our technique with a state-of-the-art fluid capturing method, building a pipeline to bond the fluid simulation in graphics with real-world flows.

Our present work still has several limitations. The re-simulation scheme and flow editing approach, which depend on modal reduction, is unable to handle a new domain with an enlarged boundary yet, since the data we can reuse are confined within the domain of the existing flow. To enlarge the simulated domain, extra efforts must be made to modify the subspace and provide extra information. The proposed EMD algorithm, although proved suitable for gas flow, guarantees no temporal coherence either. A possible solution for remedy is to use some additional temporal constraints during the EMD process simultaneously considering the results from multiple adjacent frames. The fitting of thin plate spline needs to solve a dense matrix. To accelerate the solution, we could use biharmonic B-spline [FW12] instead of thin plate spline, whose basis functions are localized.

The approach we have proposed indeed offers the possibilities to bridge the gap between fluid capturing and simulation, and we believe more applications beyond what we have discussed in this paper shall be explored in order to do a full justice for this method. Future topics may include investigating more advanced fluid-capture systems to construct larger database from various categories of fluids, which are readily available towards flow synthesizing. In addition, we are considering possible investigation of guided simulation in
an expanded domain to handle more complex scenarios with proper boundary conditions.

References


[GT*14] GREGSON J., IHRKE I., THUEREY N., HEIDRICH W., ET AL.: From capture to simulation-connecting forward and inverse problems in fluids. ACM Transactions on Graphics 33 (2014), 2, 3, 6, 8, 9


submitted to COMPUTER GRAPHICS Forum (10/2015).


