Visualization of Material Interface Stability

Harald Obermaier* University of California, Davis, USA Fang Chen University of Kaiserslautern, Germany Kenneth I. Joy University of California, Davis, USA Hans Hagen University of Kaiserslautern, Germany

ABSTRACT

Material interfaces and free surfaces are a topic of increasing interest in the field of computational fluid dynamics. In parts, reconstructed interfaces from such multi-fluid simulations behave like classic integral surfaces as known in the visualization community, while other regions of the surface undergo topological changes or behave orthogonally to what is expected by the underlying flow field. Thus, the analysis of the flow field in connection with material interface shape and topology is a challenging task. We develop a technique that facilitates visualization and analysis of such complex material interface behavior over time. For this matter, we track a surface parametrization of time-varying material interfaces and identify locations of interaction between material interfaces and fluid particles. Splatting and surface visualization techniques produce an intuitive representation of the derived interface stability. Our results demonstrate, how the interaction of the flow field with the material interface can be highlighted by appropriate extraction and visualization techniques and how the developed techniques can aid analysis of mixing and material interface consistency.

Index Terms: I.3.5 [Computational Geometry and Object Modeling]: Curve, surface, solid, and object representations—; I.6.6 [Simulation Output Analysis]—

1 INTRODUCTION

Modern scientific simulation methods benefit from an increase in processing power and can be used to simulate more and more complex physical phenomena as more computational resources become available. Flow simulations, one of the most widely used simulation types in physical modeling and visualization, are, for example, no longer limited to simulating just one fluid, but have successfully been applied to multi-fluid problems with highly heterogeneous fluid properties. These multi-fluid simulations are capable of modeling fluid interactions and are an important tool to investigate fluid mixing in a number of application areas such as chemical extraction processes, petroleum industry, and combustion systems.

While numerous techniques in flow visualization perform domain segmentation by analyzing properties of the underlying field, such as topology or divergence oriented methods, multi-fluid simulations directly imply a separation of the flow field into multiple domains. In many cases it is required to locate, reconstruct or track these interfaces or boundaries between two or more given materials. In recent years, several different methods to reconstruct material interfaces from output of such multi-fluid simulations have been introduced in the visualization community, most of which can extract snapshot-like interfaces in single simulation time steps (cf. [6, 20, 2]) and approximate interface topology. However, coherent extraction of volume-accurate time-varying material interfaces is still an open problem. The work presented in this paper does not focus on the extraction of accurate snap-shot representations of multi-fluid interfaces, but aims at providing a stability or coherency visualization of extracted material interfaces and for the first time allows a completely consistent display and tracking of material interface evolution. Therefore, the main goal of the developed techniques is to give insights into interface coherency, interface behavior, and fluid mixing.

Given reconstructed material interfaces and a time-varying flow field, we make use of time-surface integration to track and perform consistent parametrization of interface meshes. Several challenges with respect to surface matching and parametrization accuracy are handled by our dynamic remeshing and parametrization seam tracking techniques. The resulting parametrized mesh is used for feature tracking and identification of so-called interface instabilities. These instabilities are visualized as volume rendered particles that detach from the multi-fluid interface. Furthermore, direct user interaction techniques on the parameterized surface are presented that facilitate interactive tracking of interface features over time.

For the first time, our techniques support the analysis of material interface stability and relation to multi-fluid mixing. This work contributes to the visualization community in the following ways:

- Consistent parameterization of material interfaces throughout a complete simulation.
- Visualization of material interface stability and coherency.
- Interactive visual interface tracking.

The remainder of this work is organized as followed. Section 2 gives an overview of related work. Fundamentals and background information on multi-fluid simulations, resulting material interfaces and their reconstruction are presented in Section 3. In Section 4 we analyze interface stability and develop tracking as well as stability extraction algorithms. Interaction and visualization techniques for the parameterized interfaces and surface instabilities are presented in Section 5. An analysis of the developed methods is performed in Section 6. Section 7 concludes this paper.

2 RELATED WORK

This work addresses, combines and develops techniques from three areas: material interface reconstruction (MIR), flow integration, and mesh parametrization or texturing. In the following we give a brief overview of relevant related work in these fields.

For MIR, Noh and Woodward [20] first introduced the Simple Line Interface method, where cells are partitioned with simple axisaligned lines or planes in order to match and preserve the volume. Later Youngs [30] and Rider et al. [21] developed the Piecewise Linear Interface Calculation algorithm. Recent MIR approaches find a smooth and continuous interface based on fractional material data. Such discrete approaches on MIR are done by Bonnell et al. [6], Meredith and Childs [18], and Anderson et al. [2]. The former two approaches construct boundaries by calculating intersections between cells in material space with cells that represent the dominance of one material, while the latter uses a volume-adaptive active interface model to generate high-quality boundary meshes. All material interface meshes shown in this work were extracted by a variant of Bonnell's method.

^{*}e-mail: hobermaier@ucdavis.edu

One important characteristic of fluid interfaces is time dependency. In classic vector-field visualization, the dynamics of fluids are captured by integral lines or surfaces. A large body of work has been devoted to address the computation and visual capabilities of these integral surfaces. Following Hultquist's [11] fundamental work on the construction of stream surfaces, van Wijk [28] presented stream surface construction based on implicit surfaces, which can handle irregular surface topology. Computation and visualization of such surfaces has been studied in mesh-based [23] and in point-based settings [22]. A recent line of research addresses the computation efficiency and accuracy of integral surfaces [9, 14, 7] by applying mesh adaptivity schemes during extraction.

Texturing of surfaces, adaptive meshes, or flow based texturing has been researched by several authors over the last few years. Texture synthesis models [26, 29, 4] focus on how texture can be aligned to a vector field or given surface shape. However, these models are applied individually to surfaces at each time frame or show loose correspondence only, hence the dynamics of material surfaces are commonly not fully captured. To study how texture evolves along the fluid, van Wijk introduced an image-based flow visualization technique for curved surfaces [29]. In the meantime, Neyret [19] and Bargteil et al. [3] proposed texture advection methods to animate a textured surface which is passively advected by the flow field. Texture advection has been applied to iso-surfaces and stream surfaces by Laramee et al. [17, 16].

3 MATERIAL INTERFACES

Material interfaces are given as fluid boundaries in multi-fluid simulations. These simulations and their output can take several forms [15, 10, 27], one class of which, the *volume of fluid* (VOF) ([10]) method is particularly wide spread and used throughout this work. In the following we introduce the necessary background of VOF simulations, material interfaces and their reconstruction.

3.1 Volume of Fluid Simulations

In VOF methods, a fluid identification function is advected during simulation and discretized in the form of a *volume fraction* function f at output. For two fluids F_1 and F_2 in a 3D time-varying simulation such a volume fraction function

$$f: \mathbb{R}^3 \times \mathbb{R} \to [0, 1] \tag{1}$$

is given as the percentage of fluid F_1 present in a given volume or cell $V \subseteq \mathbb{R}^3$ of the data set at time $t \in \mathbb{R}$. Consequently, an arbitrary cell or volume V of the domain contains a fraction of f(V, .) of fluid F_1 and a fraction 1 - f(V, .) of fluid F_2 . When generalized to multiple fluids, this volume fraction function becomes vector valued.

During advection of the fluid identification function, mass of the fluids as well as fluid identity of flow particles is conserved. The VOF method is well-known as a simulation technique that can handle the occurrence of complex fluid interface behavior including interface topology changes such as bubble break-off (see Figure 1).

3.2 Interface Reconstruction

Material interface reconstruction is concerned with the extraction of a geometric representation of fluid boundaries. In the context of VOF methods, this requires the processing of all cells or volumes V that are not covered entirely by one fluid, i.e., $f(V, .)_i \neq 1 \quad \forall i \in \{F_i\}$.

In such a cell the concrete geometric representation of the interface is not unique as illustrated in Figure 1. More plausible representations can be obtained by incorporating volume fraction data from neighboring cells into the extraction process. However, neither the shape, nor the topology of the extracted interfaces is unique for a non-trivial volume fraction function. Thus, different MIR techniques may extract different surface representations, leading from discontinuous representations that only approximate the



Figure 1: Left: Two time steps of a 2D VOF simulation with possible topology change. The interface lies in the region with a volume fraction between 0 and 1. Right: Four reconstructed interfaces for the highlighted cell with f = 0.5. One fluid is shown in blue, the other in white. All four extracted interfaces are exact with respect to reconstructed fluid volumes. An infinite number of other representations exist that reconstruct fluid volumes correctly, leaving the true interface shape and topology unknown.

given volume fractions [20, 21] to continuous and smooth methods that adhere to the specified volume fraction data [6, 2].

In the remainder of this work, we assume that we are given an extracted material interface mesh for every time step of the simulation along with the corresponding flow field. Note that virtually all MIR techniques, even the discontinuous ones, yield a representation that can be converted into a suitable triangulated mesh.

3.3 Interface Behavior

In addition to topology variation caused by the volume fraction function itself, or resampling, ambiguity of interface reconstruction can lead to sudden changes in topology over time as well. Consequently, subsequent interfaces often exhibit significantly different local topology. This is a stark contrast to other surface representations known in the visualization community, namely integral surfaces such as path- or time surfaces, and significantly complicates consistent and coherent visualization of interface surfaces over time (see Section 4). In this paper we analyze the behavior of material interfaces and time surfaces, using the following notions:

A polygonal *mesh* is a set of vertices $v_i \in \mathbb{R}^3$ connected by edges $e_j = \{v_{j_1}, v_{j_2}\}$ that delineate mesh faces, such as triangles. A point $x \in \mathbb{R}^3$ is said to lie on the mesh if it is located on this piecewise linear representation given by the mesh faces.

Following this notion, a material interface in a 3D time-varying VOF simulation with *n* time steps t_1, \ldots, t_n consists of an ordered set of interface surface meshes $\{M_1, \ldots, M_n\}$. Since common MIR is performed per time step, M_i and M_j with $i \neq j$ have no explicit relationship apart from the fact that they were both extracted by the same MIR algorithm (this constraint may be weakened as emphasized in the results section). Thus, MIR meshes of subsequent time steps do not have to show a consistent behavior or be comparable in a meaningful way. This allows for arbitrary difference in topology between subsequent interface meshes.

In contrast to material interfaces, a time surface in a 3D timevarying flow field $g : \mathbb{R}^3 \times \mathbb{R} \to \mathbb{R}^3$ has an explicit relationship between subsequent mesh representations. In time surface construction, individual mesh representations $\{T_1, \ldots, T_n\}$ are created by flow advection of vertices of an initial mesh along trajectories

$$s(x,t) = x + \int_{t_1}^t g(s(x,\tau),\tau)d\tau, \quad s(x,t_1) = x,$$
 (2)

that correspond to the common definition of integral path-lines [7]. This iterative character of time surface creation guarantees that subsequent representations possess a shared surface parametrization. Therefore, a mesh parametrization function p with $p: T_i \to \mathbb{R}^2$ exists for every T_i , and the property of temporal parametrization consistency between subsequent time surface meshes T_i and T_{i+1} ,

$$p_{i}(s(v_{i},t_{j})) = p_{i+1}(s(v_{i},t_{j+1}))$$
(3)

holds. In other words, equations (2) and (3) ensure that time surface parameterization is preserved during advection. Thus, there is a unique correspondence between subsequent time surface meshes that implies a consistent parameterization but makes global topology changes of time surfaces impossible. However, suitable choice of this mapping to \mathbb{R}^2 (see Section 4.3) allows consistent application of visualization techniques such as texturing.

In summary, time surfaces correspond well to flow behavior but not necessarily to material boundaries, since they cannot model topology changes. Material interfaces, however, while corresponding to snapshots of material boundaries, do not inherently possess a coherent parametrization. Both surface and mesh types can be extracted from VOF data, a fact that allows the combination of both surface properties as demonstrated in the remainder of this work.

4 MATERIAL SURFACE STABILITY

The definitions given in the previous section allow us to define the notion of material surface stability. The central question we want to answer in this work is: How do time surfaces and material interfaces correspond and how stable is their behavior with respect to one another? The answer to this question can help evaluate MIR methods, track interface features, and highlight interface mixing.

4.1 Definition

A material interface M_i and a time surface T_i in time step *i* correspond if the time surface was created by advection of a material interface M_j , $j \le i$. Analysis of the evolution of corresponding time surfaces and MIR meshes reveals one of the following behaviors:

- 1. *Stability*: If there exists a pair of positions x_M and x_T on a material and a corresponding time surface at time step j with $||x_M^j x_T^j|| < \varepsilon$ then there exists a position x_M^{j+1} such that $||x_M^{j+1} p_{j+1}^{-1}(p_j(x_T^j))|| < \varepsilon$. Such a point x_M^j is therefore a stable material interface point with respect to the flow field and stays within an ε band of the time surface. Note that p^{-1} locates a point on the surface in parameter space.
- 2. *Detachment*: If there exists a pair of positions x_M and x_T with $||x_M^j x_T^j|| < \varepsilon$ and there is no position x_M^{j+1} with $||x_M^{j+1} p_{j+1}^{-1}(p_j(x_T^j))|| < \varepsilon$. Such a point x_T^j is therefore being detached from the material interface by flow advection. That means it leaves an ε band of the material interface.
- 3. *Attachment*: If there is no such position x_T^{j+1} for a given x_M such that $||x_M^{j+1} x_T^{j+1}|| < \varepsilon$. Such a point x_M^{j+1} in the flow field is therefore being attached to the material interface representation by flow advection. That means it enters the ε band of the material interface.



Figure 2: Illustration of two consecutive 2D material interfaces. The three points A, B, C are representatives of stable, detaching, and attaching flow particles. A remains on the interface and can be used for parametrization. B detaches from the interface, whereas C attaches to the interface in the second time step.

An illustration of these cases in shown in Figure 2. In the following, we denote detachment as material surface instability, whereas stability is used for parametrization transfer. There are numerous causes for attachment or detachment behavior in surface evolution of reconstructed interface meshes. Most frequent causes include natural topology changes in material interfaces, a non-unique extraction process with respect to VOF interface topology (Section 3), sampling resolution or strategy of the reconstruction method, and accuracy of the reconstruction with respect to fluid volumes.

4.2 Algorithm Outline

Assuming a VOF simulation was run and provides extracted material interface meshes for the whole length of a time-varying vector field g, our algorithm to detect and process material surface stability, detachment, and attachment takes the following steps:

- 1. Parameterize the first material interface M_1 with a function p, select i = 1 as current time step.
- 2. Use material interface M_i as seed surface of a time surface T_i with corresponding parametrization p as defined on M_i .
- 3. Advect the time surface T_i to the next time step i + 1.
- 4. Parameterize M_{i+1} by comparing it to T_{i+1} and identifying regions of surface stability.
- 5. Advect instable regions of M_i through the flow field.
- 6. Repeat from step 2 with interface M_{i+1} and time step i+1.

In the following sections, we explain these six steps in detail.

4.3 Initial Parametrization

Material interfaces of all time steps serve as seeding structures for time surfaces as indicated by step 2. Thus, obtaining a correct tracking of stable and unstable interface regions requires the presence of a well-defined parametrization function p for the first material interface M_1 . This parametrization is created explicitly for the first MIR mesh and transferred to subsequent interfaces (see Section 4.5).

Mesh parametrization is a well-researched area and has led to a number of techniques including patch-based parametrization, texture synthesis, and global parametrization techniques based on mesh flattening [26, 5, 24]. We prefer a global parametrization scheme over local patch-based schemes as they ensure that p is an injective function. Such schemes reduce the amount of parametrization seams and allow interactive modification of the mesh in a shared parameter space as shown in 5.2.

The global parametrization scheme of our choice cuts the mesh along areas of strong distortion and unwraps the resulting open 2manifold to the plane by angle based flattening. For this matter we use custom implementations of seam layouting [25] and mesh flattening [24]. Note that this parametrization function p is discontinuous along the generated seam edges but minimizes angular distortion during parametrization. Simpler global parametrization strategies such as sphere mapping are applicable as well, but often introduce high distortions, singularities, and complex discontinuities and are frequently non-injective functions. The choice of initial parameterization has no direct effect on instability classification, but governs parameters of stable regions and therefore influences the final visualization (see Section 6).

4.4 Interface Advection

After parametrization a material interface mesh is used as seeding structure for time surface advection. In the simplest case, we set $T_i = M_i$ and advect the surface to the next time step by using an adaptive Runge Kutta integrator of order five [8]. In this case mesh connectivity and resolution stays static during advection.

More complex time surface seeding and advection strategies include subsampling of the interface mesh and generating adaptive time surfaces [14] to save integration time, as described later in this paper.

4.5 Stability Classification

The parametrization of the advected time surface T_{i+1} has to be transferred to the next material interface mesh M_{i+1} to facilitate stability analysis and consistent interface parametrization. For this matter we perform two-sided mesh matching.

Given a distance threshold ε and a vertex v on M_{i+1} in the next time step, we loop through all cells, edges, and vertices of T_{i+1} to find the closest interpolated position on T_{i+1} . This element search is locally constrained by first voxelizing T_{i+1} and its elements into a uniform grid with cell sizes corresponding to ε . If the closest position $x \in T_{i+1}$ satisfies $||x - v|| < \varepsilon$, parameter values of x as obtained by barycentric interpolation on cells of T_{i+1} are assigned to v, declaring it as part of a stable region. If there is no such closest position, v represents an attachment point. Since attachment points have not been part of the material interface previously, they are assigned distinct parameter values outside of the mesh's regular parameter space that identify them as attachment points.

The second matching is performed by swapping the roles of T_{i+1} and M_{1+1} . For every vertex on T_{i+1} we find a closest point $x \in M$. If $||x - v|| < \varepsilon$, v is part of a stable region. If there is no such x, v represents a detachment point, or a material surface instability.

There are certain cases especially during interface merging events, e.g., merging of a bubbles, where several non-neighboring positions $x \in T_{i+1}$ satisfy the distance constraint $||x - v|| < \varepsilon$. There are several options to handle these cases. a) Mark v as possessing multiple positions in parameter space, b) assign the parameter value of the closer position to v, or c) store v as non-tracked, unparameterized vertex with parameter values outside of the valid parameter space. In extreme cases, these situations can lead to discontinuous parameters on M_{i+1} , in which case aforementioned seam treatment procedure is applied again. The examples shown in the results section make use of option a) for computational simplicity.

4.6 Instability Tracking

Instability points as identified in the previous section indicate regions, where subsequent extracted material interface meshes are not coherent with fluid motion, e.g., in regions with strong topological changes. Tracking these detachment regions over time can be used to evaluate consistency of the extracted material interfaces, analyze flow divergence and quantify mixing, as these instabilities used to mark borders between different liquids.

To this end we seed instability particles on these regions and advect them through the complete simulation. Such instability particles store information about the time step of detachment as well as their last valid parameter values for visualization purposes. In the examples used, we seed particles at instable vertices and adjacent triangle center positions.

4.7 Improvements and Optimizations

The steps detailed in the previous sections describe a full run of our algorithm. There are, however, several additional optimization and improvement considerations that we detail in the following.

4.7.1 Seam Treatment

Global parametrization schemes introduce seams when applied to closed meshes. If these seams are not handled specifically during parametrization transfer, severe visual artifacts can occur that worsen as the simulation progresses.

Given a cell *c* of interface mesh M_i , an edge $e = (v_1, v_2)$ of *c* contains a discontinuity of the parametrization function if vertices v_1 and v_2 map to opposing sides of a seam. If mesh parametrization is used for texturing, the cell is incorrectly mapped with a large part of the parametrization texture. Figure 3 depicts this phenomenon.

Such an invalid cell or triangle can be identified by analyzing its properties in parameter space: It either has a flipped normal in parameter space, extremely large edges in parameter space, or causes strong local compression of parameter space when compared to the last time step. Several triangles that fulfill all of these properties are visible in Figure 3.



Figure 3: Left column: Images of flattened material interface mesh before advection (top) and after advection without and with seam treatment. Parametrization texture is shown in the background. Large triangles across parameter space indicate invalid seam treatment. Right column: Material interface mesh before advection (top) and after advection without and with seam treatment. Close ups of parametrization seams reveal artifacts if discontinuities are not respected during the parametrization process.

Since the correct projection into parameter space is unknown for an invalid triangle and cannot be computed analytically, we employ an iterative edge-bisection technique to create sub-triangles that match the parametrization seam. For an edge $e = (v_1, v_2)$ this bisection technique is described in Algorithm 1 (see also Figure 4).

Algorithm 1 Pseudocode for edge bisection.			
1:	$w_1 \leftarrow v_1, w_2 \leftarrow v_2$		
2:	while $\frac{\ w_1-w_2\ }{\ v_1-v_2\ } > \delta$ do		
3:	$v \leftarrow (w_1 + w_2) \cdot 0.5$		
4:	if edgeIsInvalid $(p(v), p(w_1))$ then		
5:	$w_2 \leftarrow v$		
6:	else if edgeIsInvalid $(p(v), p(w_2))$ then		
7:	$w_1 \leftarrow v$		
8:	end if		
9: end while			

In Algorithm 1, control over accuracy of seam approximation is given by choice of δ . In practice we chose δ around 2e - 4. Lower values do not slow seam approximation significantly, but produced no noticeable difference due to small triangle sizes in the used test data sets. After one run of the algorithm, we have a position v on the edge that approximates the seam location and two locations in parameter space $p(w_1)$ and $p(w_2)$ that lie on opposing sides of the parametrization seam. Once this algorithm is run for each edge



Figure 4: Left: A triangle (black) stretches incorrectly over a large region of parameter space, as it crosses the periodic boundary (blue). From left to right: Edge bisection with point-location in parameter space finds intersections between triangle edges and parametrization seams that allow retriangulation of the mesh (gray polygons).

of the invalid triangle, we are left with one, two, or three intersected edges. The affected triangle is remeshed by splitting edges at p. Incorporation of $p(w_1)$ and $p(w_2)$ into the re-meshing process ensures that all vertices of the new triangles are located on the same sides of the seam. Figure 5 shows the triangle splitting and re-parametrization process. In the special case of three cuts, parametrization of one of the new sub-triangles is extrapolated by using laws of cosine, as common in mesh unwrapping [24]. If more than one seam crosses a triangle edge, the triangle is split and the procedure is applied to all sub-triangles.



Figure 5: Each edge of a triangle may be cut by a parametrization seam. Splitting and reparametrization of the new triangles is straightforward for the one and two cuts cases (examples of seam locations shown in blue). In the three-cut case shown on the right, we compute parameter values of the third vertex of the middle triangle by using known parameter values of the remaining two vertices together with angle informations in the triangle.

4.7.2 Parametrization Accuracy

During parametrization transfer from the time surface to the material interface mesh, significant parametrization details can be lost due to resampling of parameter space on a mesh vertex level. This becomes especially critical if an adaptive time surface was used for advection and interface mesh vertices show very low correspondence in position and density. Therefore, we offer the option to locally retriangulate the material interface mesh during the parametrization transfer process, if an accurate representation of small scale parametrization features is desired. That means vertices of the time surface are not only used to transfer surface parametrization, but serve to remesh their closest triangle of the material interface mesh as well. This can be seen as projecting the time surface connectivity onto the material interface mesh. Computationally light-weight, the major drawback of this retriangulation step is an increased resolution of subsequent time surfaces and therefore largely increased advection times.

4.7.3 Mesh Subsampling

Analysis of algorithm performance in Section 6 indicates that time surface advection is the computationally most expensive step. Mesh subsampling is an optimization strategy to reduce this computational overhead, which is closely related to parameterization accuracy. We simplify a material interface mesh by removing a mesh vertex v if the following constraints hold for v and its projection point v_{proj} on the simplified mesh

$$curv(v) < \delta_1$$
$$\|p(v) - p(v_{proj})\| < \delta_2$$

where $p(v_{proj})$ is obtained by barycentric interpolation on the cell of the simplified mesh that contains v_{proj} . By constraining the deviation between parameterizations of the full and the simplified mesh, we ensure no significant parametrization detail is lost. If additionally the maximal resulting mesh edge size is constrained, the mentioned constraints guarantee that neither prevalent surface features, i.e., features with large mesh curvature, are missed, nor regions with non-linear parametrization behavior are merged. Feature preservation is controlled by choice of δ_1 , whereas parametrization details are preserved by δ_2 . These two parameters influence computation times and parametrization accuracy as demonstrated in the results section. Note that strong mesh simplification is best used with subsequent adaptive time surface integration to insure no important flow features are missed. To improve coherency with highresolution surface advection, the surface can be upsampled during instability classification. For additional performance improvement, particle and time surface advection, as well as closest-point identification for parametrization transfer is parallelized on a per-vertex level (see e.g., Buerger et al. [7]).

5 VISUALIZATION AND INTERACTION

This section serves to define visualization challenges that come with material interface stability visualization and detail the proposed visualization techniques to address these challenges. Furthermore, we detail a direct interaction method that allows userspecified visual feature tracking by direct surface drawing.

5.1 Visualization

The data to be visualized consists of a set of material interface meshes with according parametrization and a set of detachment points that indicate material surface incoherencies and mixing regions. For visibility reasons neither detachment regions nor material surfaces should be rendered in an opaque manner, thus more sophisticated transparency rendering techniques are required.

For simple material interface mesh visualization we make use of shaders for depth-peeling, texturing, and per pixel Phong shading. Attachment regions are identified by large texture coordinate offsets and rendered either in black or fully transparent. Transparency of the surface is modulated by normal direction relative to the viewer to allow a clear look at interior structures.

Visualization of the instability particle set could be performed in numerous ways including splatting, rasterization, particle geometry, or point set visualization [13, 12]. We opt for two alternatives: Fast density-based volume rendering performed by counting particles present in cells of a low resolution 3D texture. This texture is subsequently visualized by slicing-based volume rendering. Alternatively we use particle splatting if individual particle properties are to be shown. In contrast to the splatting technique, density map visualization requires no particle sorting for correct transparency rendering and rendering performance is independent of the number of particles. A trade-off, however, is the lack of individual particle information present in density maps.

A challenge that arises from our choice of visualization is the combination of two different transparency techniques, namely volume-rendering on proxy geometry for particles and depthpeeling for surfaces. We combine both approaches by employing the depth-buffer layers created during depth-peeling to perform depth-based interval clipping. First, layers of the material interface mesh are rendered to off-screen buffers b_i by the application of standard depth-peeling techniques. The final image is then composed by traversing the stack in back to front order: We blend the color of layer b_i to the frame buffer, followed by drawing the transparent proxy geometry (e.g., slices for volume rendering or textured splats) with parts that exceed the depth range specified by depth-buffers b_i and b_{i+1} being clipped in a fragment shader. Depth buffers b_0 and b_n represent the near and far clipping plane.

5.2 Interactive Surface Drawing

As opposed to classic texture synthesis methods, the availability of a consistent global parametrization of material interface meshes over time supports tracking of feature development. Visual tracking of features can be aided significantly by facilitating interactive modification of a parametrization texture directly on the mesh.

We give the user the capability to draw on the mesh and track the evolution of this drawing over time to observe stretching, deformation, and movement in general. Our implementation follows standard picking procedures. When the user clicks or drags over the mesh, we redraw a small part of the surface around the cursor position by frame-buffer scissoring and use color based per-triangle picking. The exact position on the triangle is then found by raytriangle intersection in three-space, allowing the computation of selection position in parameter space by barycentric interpolation on the selected triangle. The parametrization texture is then updated accordingly. In the case of line drawing seam discontinuities have to be resolved by seam treatment as proposed in Section 4.7.1

6 RESULTS

We have applied our algorithm to three time-varying 3D VOF data sets and present the obtained results in the following. All data sets were simulated with the OpenFOAM simulation toolset [1] and consist of between 63 and 126 time steps. Material interface meshes are extracted with a marching-cubes variant of the method of Bonnell et al. [6] and possess averages of between 14000 and 60000 vertices per time step. Note that, in theory, comparison of different extraction methods (even for subsequent time-steps) is possible but lies beyond the scope of this paper due to page limits.

6.1 Dam Break

This data set consists of 126 time steps, modeling the first 2.5 seconds of a dam break by letting an unconstrained column of fluid stream over an undersized wall-like obstacle. The extracted material interfaces possess two interesting regions which are highlighted by our parametrization and visualization technique. A large part of the fluid describes a slow and relatively consistent dropping motion, whereas a fast front with complex topology changes is observed past the wall-like obstacle. In later time steps, backflow and splashing affects almost all parts of the fluid interface. Instability visualization near the fast moving front is of special interest for reconstruction accuracy and mixing analysis.

6.2 Fluid Drop

A 2 seconds fluid droplet scenario is represented by 102 time steps. In this simulation an accelerating fluid column is dropped into a lighter liquid, producing a mushroom or bullet-impact like deformation of the fluid column. The initial phase shows an acceleration and deformation of the fluid column, followed by a second phase, the impact of the fluid column with the data set boundary and subsequent rapid topology changes. Rapid stretching of the fluid column in addition to very thin fluid layers pose challenges to MIR.

6.3 Rayleigh-Taylor Instability

The mixing of two fluids, the heavier one on top of the lighter one, is simulated for around 1.6 seconds and 63 time steps in this 3D Rayleigh-Taylor Instability simulation. The interface exhibits almost uniform deformation in the beginning and is soon governed by the forming of Rayleigh-Taylor fingers. Strong local stretching and heavy mixing characterize later time steps and needs to be robustly tracked by our parameterization technique.

6.4 Results and Analysis

Representative snap-shots of all three simulations are shown in Figure 6a. All simulations use a gridded color gradient as parametrization texture to support visualization of stretching and rotation as well as visual tracking of interface regions. Observed stretching together with unique coloring in the dam-break and fluid drop scenarios allows in-depth analysis of shape and topology changes as the interface evolves that are impossible if no consistent parametrization is available. Such stretching regions are seen to be especially prevalent around the fronts of the dam-break and fluid drop scenarios. In the dam-break scenario, our methods makes strong anisotropic deformation visible. It additionally reveals that large parts of the reconstructed interfaces are stable with respect to fluid motion as identified by consistent texturing. In the fluid-drop scenario, our technique shows that the top of the fluid drop is virtually unaffected by early deformation. Density-based instability visualization highlights regions of the flow domain that detached from a material interface in prior time steps and are potential mixing regions of the two fluids in a high resolution VOF simulation of the same phenomenon. This observation can point to regions, where increased fluid exchange might be observed in experiments and facilitates simulation analysis and joint reconstruction error-analysis. A comparison between instability visualization in the dam-break and Rayleigh-Taylor simulations reveals how instabilities in the dam-break scenario are largely caused around the fast advancing front with increased bubble break-off and turbulence, whereas slow Rayleigh-Taylor evolution creates a more even distribution of surface detachment and possible fluid interchange.

A closer look at instability visualization is given in Figure 7 for the fluid-drop scenario, where the surface is affected by rapid topology changes in later time steps. No individual instability particle properties are visible in the computationally simpler density based visualization. However, a close-up is able to reveal significant detachment behavior in the region close to thin fluid layers with strong topology changes, revealing reconstruction inaccuracies. Other parts remain stable as shown by consistent texturing. Particle splatting on the other hand allows depiction of particle properties such as parametrization value in the form of color, indicating where in parameter-space a particle detached from the material interface, thus providing the possibility to back-track surface instabilities. It is interesting to note how instabilities caused around areas with rapid topology changes propagate across the data set in later time steps. Detachment visualization provides insight into reconstruction capabilities and shows shortcomings in reconstruction accuracy and low correspondence between fluid motion and interface locations around regions with strong topology changes and thin fluid layers, indicating point-symmetric discrepancies between reconstruction and fluid motion.

While topology changes, detachment, stability, and attachment are visible in this direct form of visualization, consistent material interface parametrization allows for a visualization in parameter space as well, which allows in-depth 2D analysis of surface behavior. Figure 8 shows such a rendering of multiple time steps of a flattened material interface mesh for the dam-break example. Surface stretching and compression can be observed, as well as surface detachment and topology changes in the form of holes or splitting of the parameterized mesh, facilitating a 2D time-varying topology analysis and giving insights into which parts of parameter-space are affected by detachment behavior.

A central advantage of consistent and injective parametrization is the possibility for consistent parameter space operations, such as interactive interface highlighting. Examples of such user interaction are shown in Figure 9, where the development of user draw-



Figure 6: (a) Visualization of three time steps per column for all simulations. From left to right: Dam-break simulation with volume-rendered instability density in the last frame. Fluid-drop simulation before, immediately before and shortly after impact with the domain boundary drawn without instabilities. Rayleigh-Taylor Instability with density based instability visualization. A close-up of several Rayleigh-Taylor fingers reveals stable (consistent texturing), detachment (red detachment particles), and attachment (no parameterization) behavior. (b) Full resolution and parametrization transfer with low resolution time surface. The contrast-enhanced difference image shows changes in approximation accuracy.



Figure 7: (a) Visualization of the fluid-drop interface after impact, (b) with density-based visualization and (c) splatting. (d) shows a later time step with instability splatting, demonstrating the spreading of detachment situations over large regions of the data set.

ings created directly on the mesh can be observed. As shown, highlighting of interesting surface features such as bubbles or holes and feature tracking is made possible by user interaction. When particle splatting is used for visualization, drawings influence the color of instability particles as well, thus allowing forward and backward tracking of surface instabilities, which is of central importance for visual discovery of cause-effect relationships.

The performance of our method is heavily dependent on the speed of flow field evaluation. Table 6.4 summarizes run-times of our algorithm on a 64bit Intel Core i7 at 2.2 Ghz with 8 GB of memory. The measurements were performed for full detail time surfaces and two levels of interface simplification. Medium and



Figure 8: This image shows the flattened interface meshes in the foreground of a parametrization texture for three different time steps. Detachment causes topology changes in the parameter space of individual interfaces. Parameter space seams are treated correctly by the proposed method.

low detail representations contained between 80% and 40% of the original mesh vertices and were obtained by increasing δ_1 and δ_2 from 4.7.3. Time spent for parametrization transfer and seam treatment amounts to few seconds per time-step of the simulations. It is notable how the Rayleigh-Taylor Instability simulation does not benefit as strongly from simplification, as the other simulations do, since high mesh-complexity avoids the removal of vertices in complex flow regions. Similarly, the high number of instability particles in the fluid drop scenario reduce the impact of mesh simplification after an initially high drop. Figure 6b shows visualizations of the dam break data set created with full detail and simplified mesh advection. Note how the chosen δ_2 limits the loss of medium scale parametrization details in flat areas, while small scale details such as local rotations are lost.

A characteristic of our algorithm is its dependability on discrete mesh representations. While this is a common computer-science problem, where discretization is often necessary from computational and representational points-of-view, our method facilitates level-of-detail approaches and can theoretically work on arbitrarily highly resolved meshes. For reasons of accuracy, used resolutions



Figure 9: Interactive on-mesh drawing for feature tracking: Top: Usermade drawing on the initial interface mesh deforms strongly. Bottom: The inverse approach allows interactive marking on surfaces in later time steps. Animation rewinding supports inverse feature tracking.

Data	Full Detail	Medium Detail	Low Detail
Dam	134	113	93
Drop	139	119	117
R-T	142	135	128

Table 1: Total run-times in minutes for surface advection, stability classification, and instability advection. Medium and low detail computations include time for surface simplification.

should stay within reasonable bounds of the resolution provided by the VOF simulation data.

7 CONCLUSION AND FUTURE WORK

We have introduced a method for material interface stability visualization by using parametrization transfer and comparisons between the development of MIR meshes and time surfaces. The resulting visualizations allow the distinction between detachment, attachment, and stable regions of material interfaces in time-varying VOF data and support visual analysis of material mixing.

The presented work leaves room for future work such as adaptive parametrization textures that refine with mesh stretching. Furthermore, this work is suitable to evaluate consistency of different MIR methods and can be extended to handle multi-fluid scenarios.

ACKNOWLEDGEMENTS

This work was supported in part by the National Science Foundation under contracts IIS 0916289 and IIE 1018097, and the Office of Advanced Scientific Computing Research, Office of Science, of the U.S. Department of Energy under Contract No. DE-FC02-06ER25780 through the Scientific Discovery through Advanced Computing (SciDAC) programs Visualization and Analytics Center for Enabling Technologies (VACET) and by the German Research Foundation (IRTG 1131). We wish to thank the members of the Institute for Data Analysis and Visualization for their support.

REFERENCES

- [1] OpenFOAM (Open Field Operation and Manipulation). http://www.openfoam.com, Sept. 2011. OpenCFD Ltd./SGI.
- [2] J. C. Anderson, C. Garth, M. A. Duchaineau, and K. I. Joy. Smooth, volume-accurate material interface reconstruction. *IEEE Trans. Vis. Comput. Graphics*, 16(5):802 –814, 2010.
- [3] A. W. Bargteil, T. G. Goktekin, J. F. O'Brien, and J. A. Strain. A semi-lagrangian contouring method for fluid simulation. ACM Trans. Graphic, 25(1):19–38, 2006.
- [4] A. W. Bargteil, F. Sin, J. E. Michaels, T. G. Goktekin, and J. F. O'Brien. A texture synthesis method for liquid animations. In *Proc.* of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pages 345–351, 2006.

- [5] C. Bennis, J.-M. Vézien, and G. Iglésias. Piecewise surface flattening for non-distorted texture mapping. SIGGRAPH Comput. Graph., 25:237–246, July 1991.
- [6] K. S. Bonnell, M. A. Duchaineau, D. R. Schikore, B. Hamann, and K. I. Joy. Material interface reconstruction. *IEEE Trans. Vis. Comput. Graphics*, 9:500–511, 2003.
- [7] K. Buerger, F. Ferstl, H. Theisel, and R. Westermann. Interactive streak surface visualization on the GPU. *IEEE Trans. Vis. Comput. Graphics*, 15:1259–1266, 2009.
- [8] J. R. Cash and A. H. Karp. A variable order runge-kutta method for initial value problems with rapidly varying right-hand sides. ACM Trans. Math. Softw., 16:201–222, 1990.
- [9] C. Garth, H. Krishnan, X. Tricoche, T. Bobach, and K. Joy. Generation of accurate integral surfaces in time-dependent vector fields. *IEEE Trans. Vis. Comput. Graphics*, 14(6):1404–1411, 2008.
- [10] C. W. Hirt and B. D. Nichols. Volume of fluid (VOF) method for the dynamics of free boundaries. J. Comput. Phys., 39(1):201–225, 1981.
- [11] J. P. M. Hultquist. Constructing stream surfaces in steady 3d vector fields. In *Proc. of IEEE Vis* '92, pages 171–178, 1992.
- [12] M. Ikits, J. Kniss, A. E. Lefohn, and C. Hansen. Volume rendering techniques. In *GPU Gems: Programming Techniques, Tips and Tricks* for Real-Time Graphics, pages 667–692. Addison-Wesley, 2004.
- [13] S. A. King, R. A. Crawfis, and W. Reid. Fast volume rendering and animation of amorphous phenomena. In M. Chen, A. E. Kaufman, and R. Yagel, editors, *Volume Graphics*. Springer, London, 2000.
- [14] H. Krishnan, C. Garth, and K. Joy. Time and streak surfaces for flow visualization in large time-varying data sets. *IEEE Trans. Vis. Comput. Graphics*, 15:1267–1274, 2009.
- [15] D. Lakehal, M. Meier, and M. Fulgosi. Interface tracking towards the direct simulation of heat and mass transfer in multiphase flows. *Int. J. Heat Fluid Flow*, 23:242–257, 2002.
- [16] R. Laramee, C. Garth, J. Schneider, and H. Hauser. Texture advection on stream surfaces: a novel hybrid visualization applied to CFD simulation results. *Proc. of Eurographics 06'*, pages 155–162, 2006.
- [17] R. S. Laramee, J. Schneider, and H. Hauser. Texture-Based Flow Visualization on Isosurfaces from Computational Fluid Dynamics. In *Proc. of Eurographics 04*', pages 85–90, 2004.
- [18] J. S. Meredith and H. Childs. Visualization and analysis-oriented reconstruction of material interfaces. *Comput. Graph. Forum*, 29(3):1241–1250, 2010.
- [19] F. Neyret. Advected textures. In Proc. of the '03 ACM SIG-GRAPH/Eurographics symposium on Computer animation, pages 147–153, 2003.
- [20] W. Noh and P. Woodward. Simple line interface method. In Proc. of 5th international conference on fluid dynamics, volume 59, pages 330–340, 1976.
- [21] W. J. Rider and D. B. Kothe. Reconstructing volume tracking. J. Comput. Phys., 141:112–152, April 1998.
- [22] T. Schafhitzel, E. Tejada, D. Weiskopf, and T. Ertl. Point-based stream surfaces and path surfaces. In *Proc. of Graphics Interface* '07, pages 289–296, 2007.
- [23] G. Scheuermann, T. Bobach, H. Hagen, K. Mahrous, B. Hamann, K. I. Joy, and W. Kollmann. A tetrahedra-based stream surface algorithm. In *Proc. of IEEE VIS '01*, pages 83–91, 2001.
- [24] A. Sheffer and E. de Sturler. Parameterization of faceted surfaces for meshing using angle based flattening. *Engineering with Computers*, 17(3):326–337, 2001.
- [25] A. Sheffer and J. C. Hart. Seamster: inconspicuous low-distortion texture seam layout. In Proc. of IEEE VIS '02, pages 291–298, 2002.
- [26] G. Turk. Texture synthesis on surfaces. In Proc. of SIGGRAPH '01, pages 347–354, 2001.
- [27] S. O. Unverdi and G. Tryggvason. A front tracking method for viscous incompressible flows. J. Comput. Phys, 100:25–37, 1992.
- [28] J. J. van Wijk. Implicit stream surfaces. In Proc. of VIS '93, pages 245–252, 1993.
- [29] J. J. van Wijk. Image based flow visualization for curved surfaces. In Proc. of IEEE VIS '03, pages 123–130, 2003.
- [30] D. L. Youngs. Time-dependent multi-material flow with large fluid distortion. In *Numerical methods for fluid dynamics*, pages 273–285. Academic Press, 1982.