

# Coupling Elastic Solids with SPH Fluids

Nadir Akinci    Jens Cornelis    Gizem Akinci    Matthias Teschner  
University of Freiburg

## Abstract

We propose a method for handling elastic solids in SPH fluids. Our approach samples triangulated surfaces of solids using boundary particles. To prevent fluid particle tunneling in case of large expansions, additional boundary particles are adaptively generated to prevent gaps and undesired leakage. Furthermore, as an object compresses, particles are adaptively removed to avoid unnecessary computations. We demonstrate that our approach produces plausible interactions of SPH fluids with both slowly and rapidly deforming solids. **Keywords:** physically-based animation, fluid simulation, Smoothed Particle Hydrodynamics, fluid-solid coupling

## 1 Introduction

Simulating incompressible flow using Smoothed Particle Hydrodynamics (SPH) is a popular topic in computer animation, since its particle based nature allows straightforward handling of fine details, such as splashes and droplets. The basic SPH interpolation works well, only if the neighboring particles have similar properties. For all other cases, boundary handling techniques that are specific to the problem of interest need to be applied. There exist several schemes for handling rigid boundaries for SPH, for both one-way [1, 2, 3, 4] and two-way fluid-rigid coupling [5, 6, 7]. However, for handling the interaction of SPH fluids and deformable solids, only few works exist [8, 9, 10, 11]. When handling deformable boundaries, some important problems arise. On one hand, particle deficiency issues at the boundary need to be handled so as to prevent

spatial and temporal discontinuities of the physical properties (e.g. density, pressure and velocity) of the fluid particles. On the other hand, in the case of large deformations, leakage through the boundaries must be prevented. Additionally, pairwise forces need to be symmetric for avoiding temporal artifacts. Our work addresses these open issues by adapting the rigid boundary handling method in [7] for deformation handling. In the remainder of this section, we discuss the existing works about solid boundary handling in SPH (Sec. 1.1), and then summarize our contribution (Sec. 1.2).

### 1.1 Handling Solid Boundaries in SPH

Triangles are one of the most practical mesh representation primitives for defining solids in computer graphics. However, in SPH simulations, directly coupling triangle meshes with fluid particles poses some challenges. First of all, density estimation of fluid particles near boundaries is only possible for simpler boundaries [3], but is difficult for arbitrarily complex ones. Furthermore, for meshes with ambiguous surface orientation (e.g. non-manifold meshes), computing fluid-solid forces correctly is a challenging problem. These issues cause spatial and temporal discontinuities of the physical properties of fluid particles, which in turn introduce falsified forces to the solids in the case of two-way coupling.

In order to avoid these issues, for handling solid boundaries in SPH, most of the existing works rely on boundary particles, e.g. [8, 2, 4, 7]. In this context, there exist distance-based penalty methods [8, 2], frozen particle methods [1, 5] and methods that use combinations of both techniques [9, 4]. In purely distance-based

penalty methods, non-penetration can be guaranteed by using stiff boundary forces. However, SPH field quantities near the boundaries cannot be estimated correctly, which results in stability issues and artifacts. In frozen particle approaches, field quantities can be approximated more correctly, which avoids spatial and temporal discontinuities, e.g. [4]. In these approaches, however, satisfying non-penetration is very crucial. For this purpose, either more than one layer of frozen particles [1], frozen particles accompanied with position correction [4], or frozen particles accompanied with an additional distance-based force term [9] are used.

### 1.1.1 Two-way Rigid-Fluid Coupling

Some approaches have extended SPH boundary handling to two-way rigid-fluid coupling. In this context, in [12], fluid is considered as a collection of particles that exchange impulses with the surrounding rigid bodies. In [13], another impulse-based formulation is proposed for the two-way coupling of SPH fluids with particle-based rigid bodies. In [6], a direct forcing method for both one- and two-way rigid-fluid coupling is employed. There exist some notable problems in those two-way coupling methods. First of all, they are not purely based on hydrodynamic forces. Secondly, they rely on surface normal information for rigid bodies so as to generate forces, which is problematic for non-manifold mesh features. More recently, a momentum-conserving two-way coupling method for SPH fluids and rigids, that is completely based on hydrodynamic forces is proposed in [7]. In this approach, the problem of inhomogeneous boundary sampling is also addressed by governing particle contributions based on the represented volumes of the boundary particles. Their approach also supports fluid interaction with thin structures and non-manifold geometry. Deformable objects, however, are not addressed in [7]. In this paper we extend [7] to support solids with large deformations (see Fig. 1).

### 1.1.2 Two-way Deformable-Fluid Coupling

Interaction of SPH fluids with deformable objects is firstly presented in [8], where the

employed boundary forces are based on the Lennard-Jones potential. In this approach, boundary particles are automatically generated per triangle according to Gaussian quadrature rules. Boundary particles generated in this fashion, however, are not optimal for computing field variables (e.g. density and forces) for the nearby SPH particles, since the boundary particles are distributed non-homogeneously for different triangles with different sizes and aspect ratios. In [9], two way coupling of SPH with thin deformable shells is realized. In this work, deformable objects are also simulated using SPH, based on [14]. This approach combines SPH forces with the explicit collision handling scheme of [15] and position correction to prevent leakage in the case of deformations where the boundary particle spacing only changes marginally. The authors also state that for thin shells with low Young's moduli, their approach causes leaks. Recently, an SPH and cloth coupling method using continuous collision detection to prevent tunnelling [10], and an SPH and Finite Element Method coupling method have been proposed [11]. However, both techniques use penalty forces for the coupling, and do not handle the fluid densities correctly near the boundary. Therefore they are prone to the problems we have described in the beginning of this section. An image based fluid-deformable interaction model has been briefly discussed in [16]. In the context of Eulerian fluids, two-way fluid-shell coupling have been demonstrated in [17].

## 1.2 Contribution

In this paper, we extend the two-way rigid-fluid coupling method described in [7] to support elastic solids. In our approach, triangle deformations are evaluated and deformed parts are locally resampled with boundary particles as necessary. Adding or removing boundary particles does not introduce disturbance to the neighboring fluid particles, since the boundary particle contributions are adjusted according to [7]. Therefore, in contrast to [9], our approach can handle large deformations. Since the boundary particles are generated in a close to uniform fashion, field variables near the boundaries can be approximated more correctly, in comparison

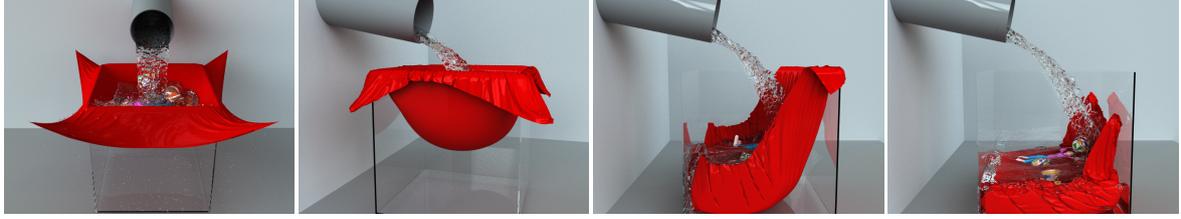


Figure 1: An elastic cloth stretches as it is filled with water. Our approach prevents leakage even in the case of large expansions.

to [8]. As it is based on [7], our method only relies on symmetric hydrodynamic forces and does not require any additional force or position correction to enforce non-penetration, which is in contrast to [9, 8, 10, 11]. Consequently, our method allows versatile interaction of SPH fluids with both 2D and 3D elastic solids.

## 2 Fluid-Elastic Coupling

As mentioned in the previous section, the approach explained in [7] works well for fluid-rigid coupling. However, a static setup of boundary particles is not sufficient for deformable objects, since large deformations may cause gaps between boundary particles, which may cause undesired fluid leakage. In order to avoid this problem, dynamic boundary particle generation is necessary. Our algorithm takes the triangle mesh of the deformable object as input and generates boundary particles for the mesh based on its vertices, edges and triangles, respectively. This section explains the main parts of our two-way fluid-elastic coupling algorithm.

### 2.1 Boundary Forces

In [7], the contribution of a boundary particle is related to its volume, which is estimated by using the inverse of the number density of a boundary particle  $b_i$  based on its boundary particle neighbors as

$$V_{b_i} = \frac{1}{\delta_{b_i}} = \frac{1}{\sum_k W_{ik}}, \quad (1)$$

where  $\delta_{b_i}$  denotes number density of the particle,  $k$  denotes boundary neighbors,  $W$  is a kernel function with smoothing length  $h$  and  $W_{ik}$  is a shorthand for  $W(\mathbf{x}_i - \mathbf{x}_k, h)$ . Hereby, the contribution of a boundary particle in an SPH

interpolation of a nearby fluid particle is given as  $\Psi_{b_i}(\rho_0) = \rho_0 V_{b_i}$ , where  $\rho_0$  is the rest density of the fluid the boundary particle is interacting with. Using this volume based contribution concept, the density of a fluid particle is given as  $\rho_{f_i} = \sum_j m_{f_j} W_{ij} + \rho_{f_i \leftarrow b}$ , where  $j$  denotes fluid particle neighbors, and the density contributed by boundary neighbors is given as  $\rho_{f_i \leftarrow b} = \sum_k \Psi_{b_k}(\rho_{0_i}) W_{ik}$ . Using  $\rho_{f_i}$ , the resultant pressure is computed by either using [18] or [19]. Afterwards, pressure and viscosity forces are applied to both types of particles. We refer the reader to [7] for those force computations.

Different from [7], in addition to pressure and viscosity force, we use the cohesion force of [19] between fluid and boundary particles to generate fluid-solid adhesion as

$$\mathbf{F}_{f_i \leftarrow b_k}^a = \beta \Psi_{b_k}(\rho_{0_i})(\mathbf{x}_{f_i} - \mathbf{x}_{b_k}) W_{ik}, \quad (2)$$

where  $\beta$  is the adhesion coefficient. In (2), we used  $\Psi_{b_k}$  instead of the mass of the boundary particles similar to the pressure and viscosity forces in [7]. This additional force term is added to make the fluid particles adhere to the boundaries. Finally, the adhesion force acting from a fluid particle to a boundary particle is symmetric (i.e.  $\mathbf{F}_{b_k \leftarrow f_i}^a = -\mathbf{F}_{f_i \leftarrow b_k}^a$ ).

### 2.2 Force Transfer

For rigids, the net force and torque terms can be easily computed by accumulating the forces on all boundary particles of the rigid. For deformable objects, however, deformations result in many more degrees of freedom than just position, orientation, linear and angular momentum. Here, we will assume a mass-point-based deformable solver, where the vertices represent mass points and edges represent constraints (e.g. as in [20]). When generating the boundary particles per triangle as described in Sections 2.4

and 2.5, the barycentric coordinates describing the position of a particle with respect to the positions of the corresponding triangle’s vertices are stored with the particle. As the mesh deforms, the particles are synchronized with the mesh according to these barycentric coordinates similar to [8]. When distributing the forces from boundary particles to the mass points, barycentric coordinates are again used for force weighting.

### 2.3 Boundary Requirements

The above density and force approximations are valid only if the contribution of boundary particles to a fluid particle  $\rho_{f_i \leftarrow b}$  is defined uniformly on the surface of the solid, since this term or its derivative is used in all force terms. We determined that, for a 2D simple cubic boundary particle alignment with a spacing equivalent to fluid particle spacing,  $\rho_{f_i \leftarrow b} \approx 0.35\rho_0$  when a fluid particle’s normal distance to a boundary is less than  $h/2$ . When  $\rho_{f_i \leftarrow b}$  is considerably smaller than this value at some point on the surface, fluid particles can leak through the solid at that point. We will refer to such boundary particle alignments as under-sampled. Such alignments can frequently occur when simulating expanding deformable objects, as the distances between neighboring boundary particles get larger. The inverse situation, i.e. over-sampling, occurs when there are too many boundary particles that do not further improve the approximation of  $\rho_{f_i \leftarrow b}$ . Although [7] can handle over-sampling by adapting boundary particle to fluid particle contributions according to (1), over-sampling causes performance overhead since the number of neighboring particles increases. It should be noted that, since  $\rho_{f_i \leftarrow b}$  essentially depends on the boundary particle number density  $\delta_b$ , we can use  $\delta_b$  at boundary particle positions as a metric to decide if the resampling of a region is necessary. We experimentally determined that  $3.6\Gamma(h)$  and  $10\Gamma(h)$  are appropriate thresholds for detecting under-sampling and over-sampling respectively, where  $\Gamma(h)$  is SPH kernel function dimensional factor and defined as  $\Gamma(h) = \frac{1}{h^d}$  and  $d$  denotes the simulation dimension. Our algorithm starting with Sec. 2.4 deals with creating such a sampling without causing any under-sampling, and with minimal over-sampling. The analysis of our approach in terms of field vari-

able approximation is given in Sec. 2.6.

### 2.4 Initial Boundary Particle Generation

As the boundary forces explained in Sec. 2.1 are SPH based, we aim to generate an initial boundary sampling, where we keep the sampling density of the boundary close to fluid particle sampling density. We observed that in incompressible SPH simulations, spacing between fluid particles remains around half of the SPH smoothing length. This implies that for SPH based fluid-boundary interaction, the sampling spacing of the boundary particles should be as close to fluid sampling spacing as possible to achieve the same sampling density. This motivated us to create a geometric boundary particle generation scheme, where the spacing between boundary particles is close to fluid particle spacing. Our algorithm not only satisfies the under/over-sampling requirements given in the previous section (i.e.  $\rho_{f_i \leftarrow b} \approx 0.35\rho_0$  and  $3.6\Gamma(h) < \delta < 10\Gamma(h)$ ), but it is also very efficient and suitable for locally adaptive resampling (see Fig. 3 and 4).

For the initial boundary particle sampling, our algorithm generates particles in three steps: Placing particles at vertices, particle generation for edges and sampling triangle interiors with particles.

The first step in edge sampling is determining the number of particles  $n_e$  to generate, which is simply done by dividing the edge length  $|\mathbf{e}| = |\mathbf{v}_a - \mathbf{v}_b|$  by the particle diameter  $d = \frac{h}{2}$  as  $n_e = \lfloor \frac{|\mathbf{e}|}{d} \rfloor$ . The displacement vector  $\mathbf{p}_e$  between the particles is computed as  $\mathbf{p}_e = \frac{\mathbf{e}}{n_e}$ . Afterwards, particles can be placed along the direction of the edge starting from any of its vertices with  $\mathbf{p}_e$  displacements.

After sampling triangle edges, the next step is sampling the triangle interiors. The approach we employ is similar to scanline algorithms known from rendering. In our algorithm, firstly the shortest edge  $\mathbf{e}_s$  is determined, and its normal direction  $\hat{\mathbf{s}}$  in the direction of the triangle interior is computed as  $\hat{\mathbf{s}} = \frac{\mathbf{e}_s \times (\mathbf{e}_l \times \mathbf{e}_s)}{|\mathbf{e}_s \times (\mathbf{e}_l \times \mathbf{e}_s)|}$ , where  $\mathbf{e}_l$  is the longest edge. Afterwards, the number of required sweeping steps  $n_t$  for the triangle is computed as  $n_t = \lfloor \frac{h_t}{d} \rfloor$ , where  $h_t$  is the height of the triangle in the sweeping direction

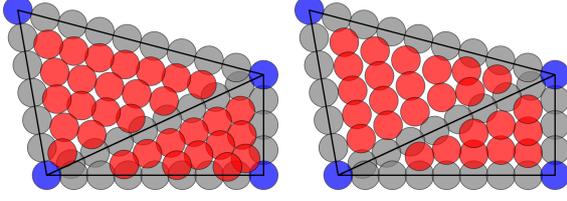


Figure 2: Two adjacent triangles sampled with scanlines in the direction of their longest edges (left). Same triangles sampled in the direction of their shortest edges (right), which results in better particle distribution, with less over-sampling. Blue, gray and red particles denote vertex, edge and interior boundary particles respectively.

$\hat{s}$ , which is computed by projecting  $e_l$  onto the sweeping direction  $\hat{s}$  as  $h_t = \hat{s} \cdot e_l$ . For each sweeping step, the line intersection positions  $\mathbf{i}_1$  and  $\mathbf{i}_2$  with the other two edges are computed as explained in [21]. Finally, particles are placed between the edge intersections with a displacement vector  $\mathbf{p}_s = \frac{\mathbf{i}_1 - \mathbf{i}_2}{n_s}$  where  $n_s = \left\lfloor \frac{|\mathbf{i}_1 - \mathbf{i}_2|}{d} \right\rfloor$ .

According to our experiments, choosing the shortest edge as the scan direction results in a better boundary particle distribution with less over-sampling (see Fig. 2). The employed step sizes in the particle generation result in a boundary sampling that is sufficiently dense according to the criterion introduced in Sec. 2.3. Furthermore, separate handling of triangle vertices and edges from triangle interiors allows robust handling of arbitrary mesh configurations without any under-sampling. On the other hand, the approach generates some over-sampling, especially between the edges of skinny triangles. However, due to the employed boundary handling scheme that can handle over-sampled boundaries (despite some performance overhead), intersections between particles can be handled when computing boundary forces.

## 2.5 Resampling of Deformed Regions

One way to completely avoid the under-sampling problem is to regenerate all boundary particles in each simulation step using the algorithm explained in Sec. 2.4. However, in stable solid simulations, deformations are temporally coherent. We utilize this information by intro-

ducing measures for determining whether a triangle needs to be resampled, or if the existing sampling is sufficient. Therefore, we avoid complete boundary sampling of a deformable object in each simulation step by using samplings from previous simulation steps as possible.

As discussed in Sec. 2.3, when under- or over-sampling is detected (i.e.  $3.6 \Gamma(h) < \delta < 10 \Gamma(h)$ ), the triangle enclosing the boundary particle, and all its edges can be resampled. Even though the resampling criterion based on  $\delta_b$  is closely related to the SPH concept, it has some important disadvantages. First of all, since it relies on  $\delta_b$  values, lazy updating of  $\delta_b$  is not possible (i.e. updating only when a boundary particle is in the neighborhood of the fluid). Furthermore, it needs an additional evaluation of the  $\delta_b$  values, which requires an additional neighborhood search among boundary particles. So as to avoid these performance issues, we derived a heuristic for resampling detection that is purely based on geometric measures, which aims to generate a full boundary neighborhood for fluid particles. In our geometric approach, triangle edges and interiors are handled and resampled separately. For detecting whether edge resampling is necessary, the number of required boundary particles  $n_e$  is computed (see Sec. 2.4). Afterwards, the edges where  $n_e$  changes between two consecutive simulation steps are resampled. For detecting whether triangle interior resampling is necessary, we employ additional rules. In each simulation step, firstly the shortest edge  $e_s$  of a triangle is determined. When  $e_s$  becomes another edge between two simulation steps, the triangle is resampled. If not, the number of necessary scanline steps  $n_t$  is computed. By multiplying the number of particles sampled onto the smallest edge  $n_{e_s}$  of the triangle with  $n_t$ , the number of particles  $n_{pa}$  that are necessary to sample the considered parallelogram is computed. When  $n_{pa}$  for a triangle changes between two consecutive simulation steps, the triangle's interior is resampled. A comparison of the two resampling strategies (i.e. the  $\delta_b$  based and the geometric heuristic based) is shown in Fig. 3 and 4. Note that our geometric approach results in similar samplings when compared to  $\delta_b$  based sampling, where  $3.6 \Gamma(h) < \delta < 10 \Gamma(h)$  for any point in a triangle is always satisfied.

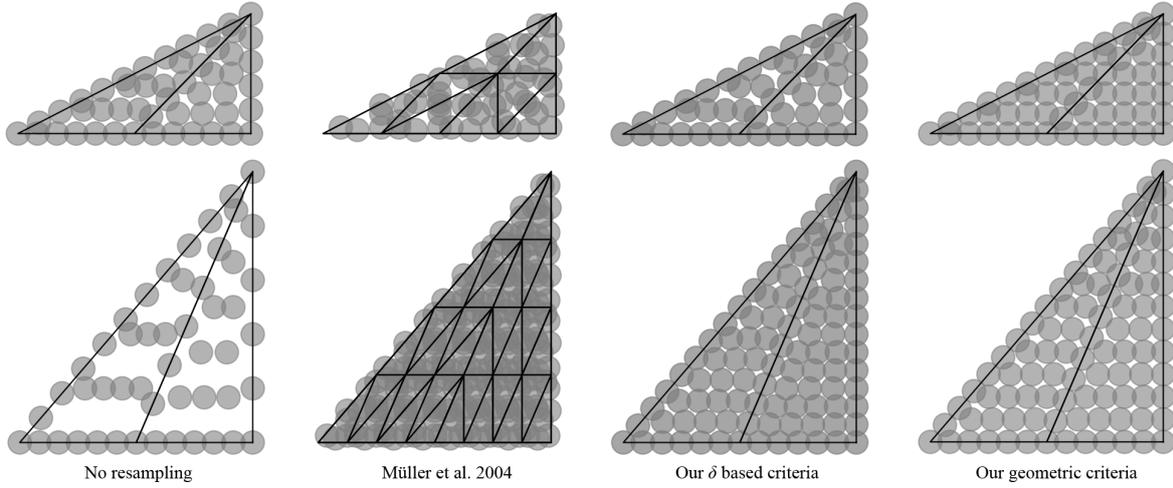


Figure 3: The initial sampling of two adjacent triangles with boundary particles based on four different boundary particle resampling schemes are shown in the top row. After the triangles are deformed, the resultant samplings are shown in the bottom row. The left most column shows the outcome when no new particles are generated. The seven point rule used in [8] tends to generate non-uniform sampling patterns. Note that our geometric resampling criterion is in good agreement with our  $\delta$  based criterion and both produce good sampling patterns without under-sampling and with minimal over-sampling. (The plotted particle diameter is equivalent to  $\frac{h}{2}$ )

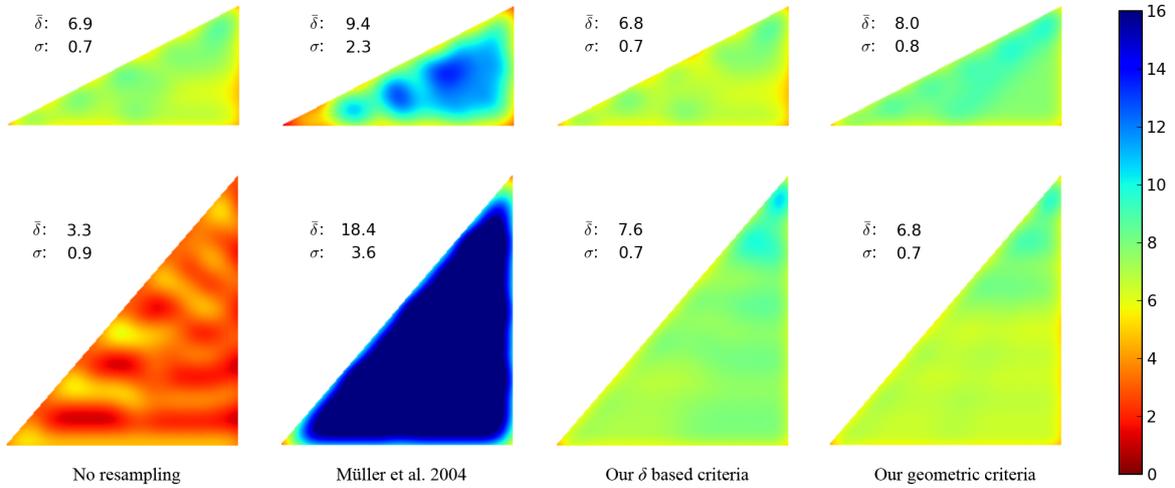


Figure 4: The number density  $\delta$  is used as a metric to evaluate under- and over-sampling. The  $\delta$  value for  $h = 1$  is computed at each point inside the triangles in Fig. 3, where  $\bar{\delta}$  denote arithmetic mean and  $\sigma$  standard deviation.  $\delta \lesssim 3.6$  denotes under-sampling and may result in fluid particle tunneling through the boundary. Whereas,  $\delta \gtrsim 10$  denotes over-sampling and causes unnecessary computational overhead. Note that the underlying formulation of [7] results in smooth boundary forces as long as the under-sampled regions are avoided. Using [8] results in a non-uniform  $\delta$  field, with variable  $\bar{\delta}$  and  $\sigma$  as the triangles deform. Such large variances can cause disturbance to the nearby fluid particles. Both of our approaches result in a uniform  $\delta$  field, with minimal temporal variance of  $\bar{\delta}$  and  $\sigma$  as the triangles deform. These properties make our approach a good match for generating boundary particles in [7].

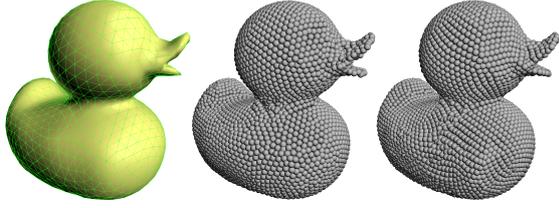


Figure 5: A duck model composed of 2160 triangles (left), sampled with boundary particles by remeshing the model using [22] and placing particles at vertex positions in 1.4 s (as done in [4, 7]) (center). The same mesh is sampled using our algorithm in 15 ms (right).

## 2.6 Comparison to Other Boundary Sampling Strategies and Analysis

In [4] and [7], the remeshing algorithm explained in [22] is used for improving isotropy of triangle meshes. After the remeshing step, boundary particles are placed at vertex positions of the resultant mesh. Even though this approach might generate better samplings compared to our method, especially in low-curvature regions (see Fig. 5), the performance cost of the approach makes it unsuitable for the application to deformable objects, which might require boundary particle generation in each simulation step. Poisson disk sampling has also been used for boundary particle generation in [23], which however requires relaxation iterations and neighborhood search when placing particles. Furthermore, similar to the remeshing based method, it is designed for producing a global boundary particle distribution, which makes it inappropriate for deformable objects, where local particle insertions and deletions are usually sufficient and also very efficient.

[8] proposes a seven point rule to generate boundary particles. Based on a user defined threshold, which is chosen relative to the smoothing length  $h$  of the fluid particles, a triangle is subdivided into four smaller triangles, which are again sampled with seven boundary particles recursively. Because of the employed subdivision technique, the number of particles sampled per triangle is  $4^d \times 7$ , where  $d$  is the recursion depth of the triangle subdivisions. Therefore, for small deformations, the approach cannot generate only few particles per triangle,

but it generates four times more particles in each subdivision. If the sampling threshold is chosen small, it may result in severe over-sampling of the triangles, while a large threshold may cause gaps between the boundary particles. Whereas, our approach only adds or removes small number of particles in case of small deformations. The difference between both methods applied to two neighboring triangles in both the initial pose and after a deformation are depicted in Fig. 3. As can be seen, our method results in more homogenous boundary particle sampling. In contrast, [8] results in under-sampling and over-sampling for different parts of the triangles. Even though smaller sampling thresholds might be used to prevent under-sampling, it might cause further over-sampling for already over-sampled triangles. One important disadvantage of over-sampling is that the required time for both neighborhood search and force computations increases significantly. Furthermore, for approximating field variables of fluid particles near the boundary, our method provides better boundary particle distributions without under-sampling and with much less over-sampling. To quantify the difference, in Fig. 4, we computed the number density  $\delta$  at each point inside the triangles based on the generated boundary particles for the boundary configurations given in Fig. 3. Note that our strategy results in a more homogenous distribution of  $\delta$ . Whereas, if [8] is used, the deviation is considerably larger. These properties make our approach a good match for using with [7] for boundary particle generation.

Another triangle sampling strategy that can be applied to local triangle sampling has been briefly explained in [24] in the context of point-based level sets. However, the case of deforming triangles is not explained in that work.

## 3 Implementation

The simulation loop in our approach is similar to the one used in [7], with some differences that are related to the deformable-boundary interface as discussed in Sec. 2.2. For simulating rigid and deformable bodies, we use Bullet Physics [25]. For SPH simulations, we use the predictive-corrective formulation presented in [18] for computing fluid pressures. For time

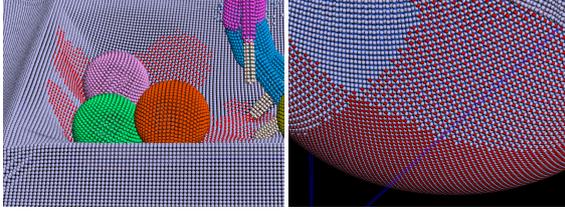


Figure 6: Underlying boundary particles of the stretching cloth in Fig. 1. The red regions denote the additional particles added to the initial boundary sampling.

step selection, we use the scheme presented in [4], where the velocities of boundary particles are also included in the time step estimation criteria. Finally, fluid surfaces are reconstructed using [26], and final renderings are done using mental ray [27].

## 4 Results

In this section, we demonstrate the versatility of our method using different scenarios. For all experiments, we used the geometric approach for resampling detection because of its performance benefits. Several simulation steps were computed for each frame. For each animation sequence, the average computation time per frame was 10 to 30 seconds, excluding surface reconstruction and rendering, depending on the frame complexity. The employed adaptive time-stepping scheme produced time steps roughly between  $5 \times 10^{-3}$  and  $10^{-4}$  seconds. In our simulations, we used different particle radii  $r$  for different scenarios, where the SPH smoothing length  $h$  was always chosen as  $4r$ . Fluid-boundary adhesion constant  $\beta$  was chosen as 0.05. The additional overhead of our approach over [7] was below 5% in all scenes, excluding deformable simulation. The simulations were run on an Intel Xeon X5690 with 12 GB RAM.

In the scene corresponding to Fig. 1, an elastic cloth with density  $1200 \frac{kg}{m^3}$  consisting of 90K uniform triangles was stretched above a box using point constraints at its corners. Afterwards, several rigid objects with different densities were dropped onto the cloth, which was later exposed to a steady fluid flow from its top, consisting of up to 1M fluid particles. The weight

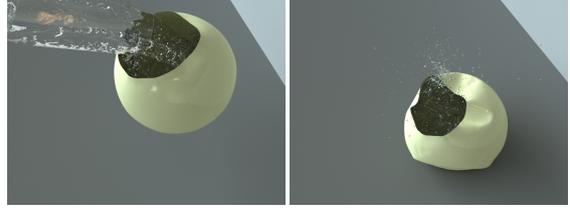


Figure 7: A thin bowl is filled with water and dropped onto the ground. Our approach can handle rapid deformations.

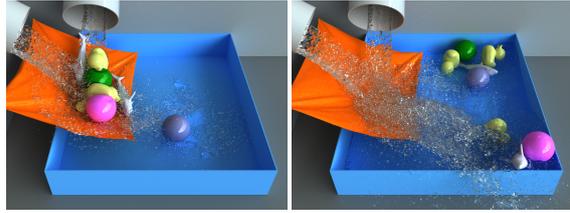


Figure 8: Interaction of fluid with a cloth and several deformable solids.

of the rigid objects and the fluid caused the cloth to slowly stretch, which was handled by updating the boundary particles in order to prevent the leakage of fluid particles through the cloth (see Fig. 6). Afterwards, when the cloth was released and shrunk back to its original size, all additional boundary particles were automatically removed. Our method ensures a sufficient boundary particle sampling that prevents leakage. Because of the regular structure of the cloth, our algorithm generated one boundary particle for each vertex for the initial sampling of the surface, which resulted in 46K boundary particles in 22 ms. Whereas, the number of boundary particles for the cloth’s maximum stretched state was 62K. The average computation time for the resampling was 9 ms per simulation step.

The presented method is also applicable to fast deformations (see Fig. 7). In this scene, a thin bowl with density  $1000 \frac{kg}{m^3}$  consisting of 6754 triangles was constrained and filled with 110K fluid particles. The initial sampling of the bowl took 14 ms that generated 42K boundary particles. When the bowl was filled with fluid, the constraints were released and the bowl was dropped onto the ground. The impact caused a sudden deformation, and consequently a rapid change in the number of boundary particles up to 44K. In this example, the average computa-

tion time for resampling was 2.5 ms per simulation step.

In order to show that our boundary particle sampling also works with volumetric deformable objects, we created another scenario (see Fig. 8). Several volumetric deformable objects and a thin shell cloth with density  $300 \frac{kg}{m^3}$  interacted with a fluid that was represented by 1.2M fluid particles. The application of our geometric resampling approach to a volumetric deformable object took under 1 ms on average.

## 5 Discussion and Future Work

Since our method always places particles at vertex positions, it can cause over-sampling when the average edge length of a triangle mesh is less than the particle diameter. So as to prevent such over-sampled areas from affecting the simulation performance negatively, such meshes can be remeshed before the simulation.

Besides the deformation model employed in the paper, we believe that our approach can work with different deformation models, as long as the deformations can be mapped to a triangle mesh and the forces can be transferred to the deformation model through the mesh vertices.

Our approach is also applicable for the boundary particle sampling of static and dynamic rigid objects with significant performance benefits over existing methods, as discussed in Sec. 2.6.

Because of the local nature of our boundary particle generation approach, it might be possible to extend it such that boundary particles are adaptively generated only for the triangles that are in the smoothing length  $h$  of the fluid particles. This can allow SPH simulations in very large solid domains (e.g. terrains, whole cities), where the boundary particle sampling of the whole domain might be infeasible because of large memory and computational requirements.

## 6 Conclusion

In this paper, we extend the two-way rigid-fluid coupling method described in [7] to elastic-fluid coupling. Our approach firstly generates initial boundary particle setups on solids with appropriate sampling. As the simulated objects deform, our approach efficiently evaluates

the boundary sampling and efficiently resamples only the necessary primitives to prevent both undesired fluid leakage in the case of under-sampling, and performance issues in the case of over-sampling. Since our approach is based on [7], which adjusts boundary particle contributions, adding new boundary particles does not result in discontinuous forces.

Our approach offers several benefits over existing works. In contrast to [9], it can handle large deformations. Furthermore, since our approach generates a more uniform boundary particle sampling compared to [8], field variables near the boundary can be better approximated. Furthermore, in contrast to [9, 8, 10, 11], it does not use additional distance based force terms. As shown in the experiments, our approach allows versatile two-way interaction of fluids with both slowly and rapidly deforming solids.

## Acknowledgments

We thank the reviewers for their helpful comments. This project is supported by the German Research Foundation (DFG) under contract numbers SFB/TR-8 and TE 632/1- 2. We also thank NVIDIA ARC GmbH for supporting this work.

## References

- [1] R.A. Dalrymple and O. Knio. SPH modeling of water waves. In *Proc. of Coastal Dynamics*, pages 779–787, 2001.
- [2] J.J. Monaghan. Smoothed particle hydrodynamics. *Reports on Progress in Physics*, 68(8):1703–1759, 2005.
- [3] T. Harada, S. Koshizuka, and Y. Kawaguchi. Smoothed particle hydrodynamics on GPUs. In *Proc. of CGI'07*, pages 63–70, 2007.
- [4] M. Ihmsen, N. Akinci, M. Gissler, and M. Teschner. Boundary handling and adaptive time-stepping for PCISPH. In *Proc. of VRIPHYS*, pages 79–88, 2010.
- [5] R. Keiser, B. Adams, D. Gasser, P. Bazzi, P. Dutré, and M. Gross. A Unified La-

- grangian Approach to Solid-Fluid Animation. In *Proc. of EG Symposium on Point-Based Graphics*, pages 125–134, 2005.
- [6] M. Becker, H. Tessendorf, and M. Teschner. Direct forcing for Lagrangian rigid-fluid coupling. *IEEE TVCG*, 15(3):493–503, 2009.
- [7] N. Akinci, M. Ihmsen, G. Akinci, B. Solenthaler, and M. Teschner. Versatile Rigid-Fluid Coupling for Incompressible SPH. *ACM Trans. Graph. (SIGGRAPH Proc.)*, 31(4):62:1–62:8, 2012.
- [8] M. Müller, S. Schirm, M. Teschner, B. Heidelberger, and M. Gross. Interaction of fluids with deformable solids. *CAVW*, 15(34):159–171, 2004.
- [9] T. Lenaerts and P. Dutré. Unified SPH model for fluid-shell simulations. In *ACM SIGGRAPH 2008 posters*, pages 12:1–12:1, 2008.
- [10] Peng Du, Min Tang, Chang Meng, Ruofeng Tong, and Lanfen Lin. A fluid/cloth coupling method for high velocity collision simulation. *VRCAI '12*, pages 309–314. ACM, 2012.
- [11] L. Yang, S. Li, A. Hao, and H. Qin. Real-time two-way coupling of meshless fluids and nonlinear fem. In *CGF*, volume 31, pages 2037–2046, 2012.
- [12] S. Clavet, P. Beaudoin, and P. Poulin. Particle-based viscoelastic fluid simulation. In *Proc. of ACM SIGGRAPH/EG SCA'05*, pages 219–228, 2005.
- [13] Seungtaik Oh, Younghee Kim, and Byung-Seok Roh. Impulse-based rigid body interaction in SPH. *CAVW*, 20(2-3):215–224, 2009.
- [14] B. Solenthaler, J. Schläfli, and R. Pajarola. A unified particle model for fluid-solid interactions. *CAVW*, 18(1):69–82, 2007.
- [15] N. Bell, Y. Yu, and P. J. Mucha. Particle-based simulation of granular materials. In *Proc. of ACM SIGGRAPH/EG SCA'05*, pages 77–86, 2005.
- [16] J. Allard, H. Courtecuisse, and F. Faure. Implicit FEM and fluid coupling on GPU for interactive multiphysics simulation. In *SIGGRAPH Talks*, 2011.
- [17] A. Robinson-Mosher, T. Shinar, J. Grestarsson, J. Su, and R. Fedkiw. Two-way coupling of fluids to rigid and deformable solids and shells. *ACM Trans. Graph. (SIGGRAPH Proc.)*, 27:46:1–46:9, 2008.
- [18] B. Solenthaler and R. Pajarola. Predictive-corrective incompressible SPH. *ACM Trans. Graph. (SIGGRAPH Proc.)*, 28(3):1–6, 2009.
- [19] M. Becker and M. Teschner. Weakly compressible SPH for free surface flows. In *Proc. of ACM SIGGRAPH/EG SCA'07*, pages 209–217, 2007.
- [20] T. Jakobsen. Advanced character physics. In *Game Developers Conference*, pages 383–401, 2001.
- [21] J.F.S. Hill. The pleasures of 'perp dot' products. *Graphics gems IV*, pages 138–148, 1994.
- [22] M. Botsch and L. Kobbelt. A remeshing approach to multiresolution modeling. In *Proc. of ACM SIGGRAPH/EG SGP'04*, pages 185–192, 2004.
- [23] H. Schechter and R. Bridson. Ghost SPH for animating water. *ACM Trans. Graph. (SIGGRAPH Proc.)*, 31:61:1–61:8, 2012.
- [24] R.D. Corbett. Point-based level sets and progress towards unorganised particle based fluids. Master's thesis, The University of British Columbia, 2005.
- [25] E. Coumans. Bullet physics library (version 2.78) [software], 2011. <http://www.bulletphysics.org>.
- [26] Gizem Akinci, Markus Ihmsen, Nadir Akinci, and Matthias Teschner. Parallel surface reconstruction for particle-based fluids. *CGF*, 31:1797–1809, 2012.
- [27] NVIDIA ARC. mental ray 3.9 [software]. <http://www.mentalimages.com/products/mental-ray/about-mental-ray.html>, 2011.