

Wake Synthesis For Shallow Water Equation

Zherong Pan¹ Jin Huang^{†1} Yiying Tong² Hujun Bao¹

¹State Key Lab of CAD&CG, Zhejiang University, China, ²Michigan State University, USA

Abstract

In fluid animation, wake is one of the most important phenomena usually seen when an object is moving relative to the flow. However, in current shallow water simulation for interactive applications, this effect is greatly smeared out. In this paper, we present a method to efficiently synthesize these wakes. We adopt a generalized SPH method for shallow water simulation and two way solid fluid coupling. In addition, a 2D discrete vortex method is used to capture the detailed wake motions behind an obstacle, enriching the motion of SWE simulation. Our method is highly efficient since only 2D simulation is required. Moreover, by using a physically inspired procedural approach for particle seeding, DVM particles are only created in the wake region. Therefore, very few particles are required while still generating realistic wake patterns. When coupled with SWE, we show that these patterns can be seen using our method with marginal overhead.

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

1. Introduction

Liquid animation is among the most stunning visual effects in computer graphics. The field has witnessed great progress in the last ten years, readily handling viscous flow, surface tension and multiphase scenarios. However, a full 3D Navier-Stokes simulation is often costly for interactive applications like real time games. Instead, the shallow water equation (SWE) serves as a cheap alternative for flow with large horizontal to vertical scale ratio. By using depth averaged velocity field, one dimension is effectively removed and the costly pressure projection is not required.

However, it is not fluid itself but the relative motion of fluid and solid body that accounts for the most interesting effects in this field such as wake flow and von Karman vortex shedding, which is illustrated in Figure 1. When the relative velocity increases, these wakes will show turbulent motion. Unfortunately, these motion details can not be captured in the standard shallow water simulation. This is mainly due to the coarse numerical approximation and low resolution adopted in current SWE solvers for interactive applications. Moreover, since empirical force model is adopted for solid fluid coupling, the flow field around moving objects is physically incorrect. Although

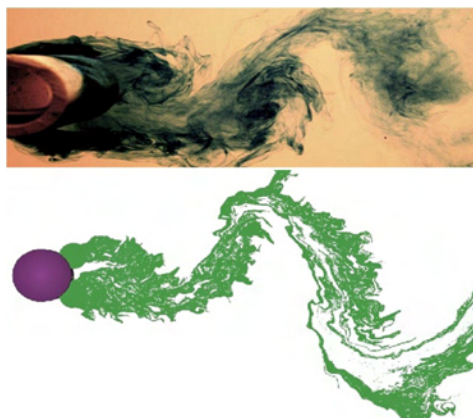


Figure 1: *Vortex shedding (top) is a natural phenomenon usually seen when high speed fluid flows over a cylinder. With a physically inspired seeding strategy, we can faithfully reproduce such effect (bottom).*

more sophisticated method, such as the one used in [Lia08], can be applied to model them, the high cost induced makes these methods impractical. Furthermore, in real-time games, it is common to use particle systems and billboards to synthesize these motion details, but the repeated patterns introduce significant visual artifacts.

[†] Corresponding Author: hj@cad.zju.edu.cn

In this paper, we propose a method to synthesize the wakes behind moving objects. In our approach, we use two different representations for the fluid, one for conventional SWE and the other for wake detail enhancement. For SWE simulation, we adopt a generalized SPH method recently introduced in [SBC*11] and [LH10]. For wake detail enhancement, we use a localized 2D version of the discrete vortex method (DVM) similar to [PK05] around each moving object. However, DVM requires a careful placement of the particles to resolve the flow field. Therefore, we propose a physically inspired procedural method for particle seeding. We show that this effectively concentrates the particles on the wake region and greatly reduces the number of particles required in DVM, while still generating phenomena like vortex shedding and turbulence as relative velocity increases. In the field of computational fluid dynamics, a similar idea is adopted in [HA97]. In order to couple the two simulations, we use SWE's velocity field as the ambient flow for DVM. While, DVM's velocity field is taken as an external force term on SWE. This is similar to the famous vorticity confinement method [FSJ01] widely used in grid based solvers.

The two simulations used in our method are both 2D, so the method is efficient enough for real-time applications. Our vortex particle seeding method is physically inspired and sensitive to both the velocities and shapes of obstacles. We show that this method is capable of generating different wake patterns including Karman vortex shedding. Moreover, the method is independent of the underlying SWE implementation and is applicable to a conventional grid based SWE solver as well. Although we make some non-physical assumptions, we show that our method achieves good balance between visual effect and performance, which is the main goal of this work. In short, the contributions are:

- Efficient enrichment of the wake details for SWE simulation.
- Physically inspired seeding strategy for wake synthesis.
- Flexible integration into various SWE implementations.

2. Related Works

Three Dimensional Fluid Simulation Full 3D grid based Navier-Stokes simulation has gained popularity since the work of [Sta99] and [FF01]. And a number of extensions have been proposed in works such as [HK05], [LGF04] and [LSSF06]. However, these methods are still too slow for interactive applications. Instead, particle-based methods, like the famous Smoothed Particle Hydrodynamics introduced in [MCG03], can achieve real-time performance with lower memory cost. Later, 3D simulations were coupled with 2D ones to get better performance in the work of [IGLF06]. And real-time performance has been achieved by [CM11] using such technique, but the whole GPU is occupied in their method. Therefore, these methods are still impractical for interactive applications.

Two Dimensional Fluid Simulation Game developers usu-

ally find good balance between surface details and simulation cost by using 2D heightfield based fluid simulation. The most popular method of this kind is the shallow water equation (SWE) and wave equation (WE). SWE provides more information than WE by taking advection into consideration. This method was introduced to computer graphics for the first time by [KM90]. Numerous extensions have been proposed later on. [OH95] used a particle system to enhance the splashing effect. [FM96] and [CLHM97] introduced solid fluid coupling. [TMFSG07] synthesized breaking wave effect for SWE. [TSS*07] simulated bubbly flow under SWE framework. [LO07] proposed a new advection scheme to reduce numerical viscosity. Recently, [CM10] proposed an even better SWE scheme and coupled several techniques to achieve better visual quality. But they only used billboards and particle systems for wake details. Apart from grid-based solvers, [LH10] and [SBC*11] showed that a standard 2D SPH method can be used for SWE simulation through a generalized pressure definition. In this paper, we adopt their method as our main SWE solver.

Instead of shallow water equation, [YHK07] showed that wave equation can also provide plausible results. However, since their method don't describe the entire flow field, interesting phenomenon such as whirlpool can not be represented. Therefore, we choose to adopt SWE as our underlying solver.

Although SWE is physically based modelling, the solid fluid coupling methods currently used are mostly based on empirical models. [CLHM97] used movable boundary condition to achieve one way solid to fluid coupling. Later, [YHK07] proposed an empirical model for two way coupling, which is also used in SPH based method of [SBC*11]. All these methods are physically inaccurate in modelling relative motion of obstacle and fluid, result in smeared out wake details.

Discrete Vortex Method (DVM) This kind of method is based on the curl form of Navier-Stokes equation. However, since the pioneering work of [GLG*95], DVM has not gained much popularity due to its various limitations, e.g. vorticity stretching and tiling causes numerical instability in 3D; particle placement is tricky to resolve the flow field; boundary condition is hard to enforce. A more sophisticated version of DVM is introduced by [PK05], a 2D version similar to which is used in this paper.

Turbulence Synthesis Turbulence modelling has a long history in fluid dynamics, for which [Pop00] provides an introduction. In computer graphics, a variety of methods have been proposed to synthesize the small scale details. [FSJ01] proposed vorticity confinement, but this approach still depends on the underlying grid resolution. Later, [SRF05] alleviated these problems through a hybrid vortex particle and grid based method but particle seeding in their method is random. Recent works such as [NSCL08], [SB08] and [PTSG09] models turbulence generation and energy transfer in a more accurate way. Since these methods adopt a stochastic approach, they cannot model periodic patterned motions that we focus on.

Other works, e.g. [ZYF10] and [YYF12], incorporate similar idea into particle based fluid solvers. All these methods base themselves on a physically correct background mean flow. Unfortunately, in our case, the background SWE flow field is far from accurate around moving objects, making these methods inapplicable.

3. Overview

Our method is a hybrid version (Algorithm 1) of shallow water equation (SWE) and discrete vortex method (DVM). In the following sections, we elaborate each substep in one simulation iteration. In Section 4, we introduce the basic SWE and DVM solvers. In Section 5, we describe our method for physically inspired particle seeding, deleting and SWE-DVM coupling. Final results and analysis are given in section 6 and 7.

Algorithm 1 Simulation Loop

```

Step SWE and couple with rigid body.
for each rigid body do
  Interpolate SWE's velocity field to DVM.
  Step DVM and perform particle seeding, deleting.
  Apply DVM's velocity field as external force on SWE.
end for

```

4. SWE and DVM Simulation

In this section, we give a brief description of the two simulation methods. For the SWE simulation, we adopt the generalized SPH scheme of [SBC*11]. For the DVM simulation, a simplified version of [PK05] is used.

4.1. Shallow Water Simulation

The shallow water equation (SWE) is the depth averaged version of the Navier-Stokes equation. It can be written as:

$$\frac{Dh}{Dt} = -h\nabla \cdot \vec{u} \quad (1)$$

$$\frac{D\vec{u}}{Dt} = -g\nabla(h+H) + \vec{a}_{ext}, \quad (2)$$

where h is the water column height, H is the height of bottom terrain, g is the gravitational acceleration, \vec{u} is the horizontal velocity and \vec{a}_{ext} represents the external force term.

Currently, most SWE solvers are grid based. These solvers usually suffer from detail loss due to numerical viscosity and high memory cost. Therefore, we choose to adopt a generalized SPH method recently proposed by [SBC*11] and [LH10]. The method preserves volume naturally and has lower memory cost. Here we briefly review this approach.

The basic discretization used in SPH is the smoothed kernel interpolation. Given an arbitrary set of particles each carrying physics property A , we can evaluate this property at any location \vec{x} through the following equation:

$$A(\vec{x}) = \sum_j \frac{m_j}{\rho_j} A_j W(\vec{x} - \vec{x}_j, l), \quad (3)$$

where m is the constant particle mass, ρ is the fluid density

and $W(\vec{x})$ is the interpolation kernel function. For the choice of these kernel functions, we follow [MCG03]. One advantage of this formulation is that differential operators need only to be applied on the kernels as follows:

$$\nabla A(\vec{x}) = \sum_j \frac{m_j}{\rho_j} A_j \nabla W(\vec{x} - \vec{x}_j, l) \quad (4)$$

$$\nabla^2 A(\vec{x}) = \sum_j \frac{m_j}{\rho_j} A_j \nabla^2 W(\vec{x} - \vec{x}_j, l). \quad (5)$$

To extend an existing 2D SPH solver for SWE, the key idea is to take the particle density in standard SPH as water column height. As in Figure 2, denser particles correspond to higher water column. Therefore, water column mass m_i and height h_{i0} are defined on each particle and kept constant across the simulation. In this way, mass is trivially preserved and the continuity Equation 1 can be discarded. Equation 2 is easy to discretize using Equation 3 and 4:

$$h_i^a = \sum_j \frac{m_j}{\rho_0} W(\vec{x}_i - \vec{x}_j, l) \quad (6)$$

$$\frac{\partial \vec{u}_i}{\partial t} = -g \sum_j \frac{m_j}{\rho_0 h_{j0}} h_j^a \nabla W(\vec{x}_i - \vec{x}_j, l) - g \nabla H + \vec{a}_{ext}, \quad (7)$$

where h^a is the averaged water column height updated in each timestep. However, merely using the above equation will lead to numerical instability. Therefore, we add a viscosity term following [MCG03]:

$$\frac{\partial \vec{u}_i}{\partial t} = \mu \sum_j \frac{m_j}{\rho_0 h_{j0}} (\vec{u}_j - \vec{u}_i) \nabla^2 W(\vec{x}_i - \vec{x}_j, l). \quad (8)$$

Note that we use the summation formulation (Equation 6) for h^a , which is more stable than the differential update formulation used in the original work of [SBC*11] and [LH10]. This formulation is also used in [AS05] to solve SWE.

Based on the above solver, two way solid fluid coupling can be approximated using an empirical force model. For fluid to solid coupling, [SBC*11] used three force terms on each rigid body, i.e. buoyant, lift and drag force. Buoyant force is calculated by tracing two rays from each particle to find the amount of water displaced. Lift and drag forces are applied following [YHK07]. For solid to fluid coupling, [SBC*11] used a force model based on fully elastic collision.

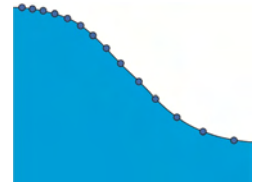


Figure 2: 1D illustration of SPH based SWE discretization. Denser particles correspond to higher water column.

4.2. Discrete Vortex Method

DVM is known for its ability to create rich flow details using very few particles. The governing equation of DVM is the curl form of Navier-Stokes equation written as:

$$\frac{\partial \vec{\omega}}{\partial t} = -(\vec{u} \cdot \nabla) \vec{\omega} + (\nabla \vec{u}) \cdot \vec{\omega} + \mu \nabla^2 \vec{\omega}. \quad (9)$$

Since we use the 2D version of DVM, the vorticity $\vec{\omega}$ reduces to a scalar since only the \vec{Z} component is nonzero and the stretching term of $(\nabla\vec{u}) \cdot \vec{\omega}$, which frequently causes numerical instability in 3D cases, vanishes. In this way, Equation 9 can be simplified to give:

$$\frac{\partial\omega}{\partial t} = -(\vec{u} \cdot \nabla)\omega + \mu\nabla^2\omega. \quad (10)$$

To discretize this equation, DVM uses a set of lagrangian particles each carrying a vorticity ω_i . Under this representation, a split-step method can be employed by considering the two terms right hand side of Equation 10 (advection, diffusion) in separate substeps to update the position and vorticity of each particle.

For the advection substep, we have to calculate the velocity field \vec{u} . In DVM, velocity field consists of three terms [PK05], giving:

$$\vec{u} = \vec{u}_\infty + \vec{u}_V + \vec{u}_P. \quad (11)$$

Here, \vec{u}_∞ stands for the velocity of ambient flow. In our method, we interpolate SWE's velocity field to DVM for this term. \vec{u}_V stands for velocity induced by the vorticity carried on DVM particles. Finally, \vec{u}_P stands for the velocity induced by boundary condition. Since we use large vorticity damping in DVM, particles are deleted quickly and seldom penetrate boundary. So, we neglect term \vec{u}_P .

The most tricky part of DVM lies in the approximation of \vec{u}_V term. Here, a velocity field is reconstructed from the vorticity field through Biot-Savart integration, whose discretized form can be written as:

$$\vec{u}_V(x) = -\sum_{i=0}^N V \frac{((\vec{x} - \vec{x}_i) \times \vec{\omega}_i) q(\|\vec{x} - \vec{x}_i\|/\sigma)}{\|\vec{x} - \vec{x}_i\|^2}, \quad (12)$$

where V is the particle volume kept constant across simulation, σ is the smoothing range, $q(\lambda) = \frac{1}{2\pi}(1 - \exp(-\lambda^2/2))$ and $\vec{\omega}_i = \omega\vec{Z}$. Performing this summation directly on every particle location will cost $O(N^2)$. Instead, [PK05] used the Fast Multipole Method [GR87, AG91] to accelerate this process to $O(N\log N)$ or even $O(N)$.

However, the constant factor in such method is large. Since the number of particles used in our case is small and high accuracy is not required, a simple TreeCode [LK01] will suffice.

For each moving object, we create a localized background grid behind the object and evaluate velocity at every grid point. As illustrated in Figure 3, the grid closely wrapping the DVM particles is updated in each timestep. Therefore, the DVM grid size is independent of the entire area of the

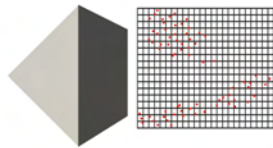


Figure 3: A wedge moving in flow. DVM particles are seeded behind it (red). And a localized grid is created wrapping these particles.

scene. To update the positions of DVM particles, we interpolate velocities from the grid for these particles as well.

For the diffusion substep, [PK05] adopted the method of particle strength exchange (PSE), where each vortex particle exchanges some of the vorticity with surrounding particles in each timestep. However, evaluating this term requires a neighbour search. Instead, we choose to simply set a large global damping on the vorticity.

5. Wake Synthesis

When a rigid body is moving in the fluid, a boundary layer will form around the body where the behaviour of fluid is under the combined influence of the ambient mean flow and surface friction. In this region, vorticity is created and confined. These vortices will then detach from the surface and be emitted into the mean flow, generating interesting wake motion behind the body. Depending on the magnitude of relative velocity, this motion can be laminar or turbulent.

Although DVM is efficient in describing and preserving vorticity, the placement of DVM particles is a challenging problem. To properly reproduce periodic wake patterns like Karman vortex shedding, the naive random seeding strategy used by [SRF05] will not work. In this section, we describe our method of particle seeding and SWE-DVM coupling. The method can effectively reproduce various wake patterns with few DVM particles.

5.1. Physically Inspired Particle Seeding and Deleting

Since very few particles are used, each one has to be carefully placed to resolve the flow field in DVM. [PK05] seeded two rows of particles with different sign of vorticity to create a steady and constant mean flow. However, since obstacles in SWE may move in arbitrary direction, their method is not applicable in our case.

We, instead, propose a physically inspired procedural method for particle seeding, which is sensitive to object shapes and velocities. The method is based on the boundary layer theory of CFD. In real life, surface friction causes the flow velocity to be zero at wall, creating a boundary layer. On the back face of the obstacle, velocity decreases and pressure increases along mean flow direction, which is called adverse pressure gradient. When the mean flow travels long enough under such pressure profile, vortices are created and confined. They will finally detach from body surface and be emitted into the ambient flow field. For more details on boundary layer theory, see [SG00].

We take the simple assumption that every rigid body is convex, common in real-time physics engines. To model boundary layer separation, detecting points are placed on all the edges of a rigid body. In addition, with each detecting point, two potentials are defined corresponding to each incident face. We accumulate these potentials in each timestep. When a threshold is reached by either of the potentials, a vortex is seeded in the downstream and the potential is reset for next seeding.

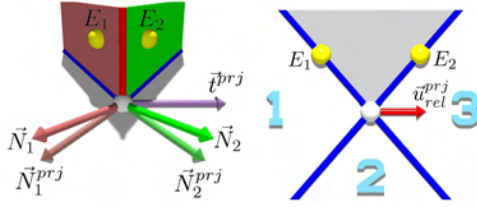


Figure 4: Left (front view): Separation detecting point (white), seeding potentials (yellow), two intersecting lines (blue) and XY-projected vectors. Right (top view): If relative velocity lies in region 3, E_1 increases and E_2 decreases.

Separation Detection For every edge of a convex rigid body, we place a detecting point P_d at the center. And two seeding potentials, E_1 and E_2 , are defined one for each of the incident face. Since we work on a 2D version of DVM, normal vectors are projected onto the XY-Plane and the projected mean tangent is calculated at each timestep. We use superscript prj for the XY-Projected vectors. This is illustrated in Figure 4 (left).

Very small vortices are visually neglectable and the average vortex radius is related to the characteristic length of the body. So, we choose to only capture vortices with radius $R = AL$, where A is a user defined value and L is the average of the three dimensions of the minimum bounding box. In our experiment, R affects the behaviour of wake details slightly as long as $0.1 \leq A \leq 0.3$. We use $A = 0.15$ in all our examples. Moreover, adverse pressure profile can only occur on the back face of the body. And the seeding possibility is proportional to the tangent velocity magnitude according to [PTSG09]. To model this effect, we choose to accumulate E_1 and E_2 above through the following equations:

$$E_1(t + \Delta t) = \max\left(\exp\left(\left(\frac{u_{E_{ff}}^1}{M} - 1\right)\Delta t\right)E_1(t), 1\right) \quad (13)$$

$$E_2(t + \Delta t) = \max\left(\exp\left(\left(\frac{u_{E_{ff}}^2}{M} - 1\right)\Delta t\right)E_2(t), 1\right) \quad (14)$$

where $M > 0$ is the minimal velocity magnitude for vortex seeding. We assume that the potential of vortex seeding increases exponentially in time when the effective velocity is above M . Although these two equations are not physically based, they give natural results and provide enough controllability over the particle seeding frequency. Here $u_{E_{ff}}^1$ and $u_{E_{ff}}^2$ are defined as:

$$u_{E_{ff}}^1 = \begin{cases} \text{abs}(\vec{u}_{rel}^{prj} \cdot \vec{t}^{prj}) & \text{if } \vec{u}_{rel}^{prj} \text{ lies in region 3} \\ 0 & \text{otherwise} \end{cases} \quad (15)$$

$$u_{E_{ff}}^2 = \begin{cases} \text{abs}(\vec{u}_{rel}^{prj} \cdot \vec{t}^{prj}) & \text{if } \vec{u}_{rel}^{prj} \text{ lies in region 1} \\ 0 & \text{otherwise,} \end{cases} \quad (16)$$

where $\vec{u}_{rel}^{prj} = \vec{u}_F^{prj} - \vec{u}_B^{prj}$ is the projected relative velocity at that point and \vec{t}^{prj} is the projected mean tangent as in Figure 4. In this way, only the relative velocity which meets adverse pressure profile, i.e. lies in region 1 and 3 in Figure 4 (right), will contribute to potentials. For example, if \vec{u}_{rel}^{prj} lies in region

3, we decide that vorticity created and confined on the red face will meet adverse pressure profile on the green face. Therefore, E_1 increases for the red face and E_2 decreases for the green face. When either of the above two potentials reaches T , a predefined global threshold, we assume that a vortex should detach from the corresponding face and seed one vortex.

Depending on the topology of obstacle mesh, several edges may have projected positions close to each other, emitting overlapped vortices. Therefore, although we accumulate potentials on all edges, we only emit vortices on edges intersecting the SWE surface. Afterwards, all the potentials on detecting point P_d whose position \vec{x} satisfies $\|\vec{x}^{prj} - \vec{x}_0^{prj}\| < R$ are reset to one for next seeding, where \vec{x}_0^{prj} is the projected detecting point from which a vortex has just been seeded. This is analogous to [PTSG09].

Particle Seeding and Deleting

When the above step determines that a vortex should be emitted from a detecting point P_d , we have to find the seeding position P_s and strength of the vortex ω . To do this, we observe that the seeded vortex should be right outside the body with no collision. Moreover, the vortex should have one boundary point attached to the mean flow. Therefore, we perform a continuous collision detection (CCD) between the body and a sphere with radius R along the direction of \vec{u}_{rel}^{prj} to find the closest point satisfying the above requirements as illustrated in Figure 5. Since we take the convex body assumption, this vortex will not be blocked by other part of the body and the CCD can be performed very efficiently using such algorithms as GJK [GJK88].

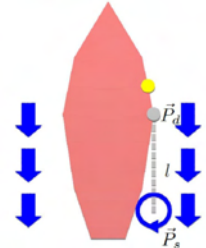


Figure 5: When the potential (yellow) of one detecting point (gray) exceeds threshold T , we find the seeding position by CCD along the gray dashed line, which is parallel to the relative velocity around the boat (thick blue arrows).

In order to determine the strength ω of the vortex, a simple approximation is $\omega = \|\vec{u}_{rel}^{prj}\|/R$. However, we propose to add a fading term to this equation, giving:

$$\omega = \|\vec{u}_{rel}^{prj}\| \exp(-l/A)/R, \quad (17)$$

where l is an additional piece of information acquired during the CCD, which is the distance between detecting and seeding position projected on \vec{u}_{rel}^{prj} . This can be written as:

$$l = \text{abs}\left(\left(\vec{P}_d - \vec{P}_s\right) \cdot \frac{\vec{u}_{rel}^{prj}}{\|\vec{u}_{rel}^{prj}\|}\right). \quad (18)$$

As in Figure 6, when a streamlined object, which usually suppress vortex separation, is moving in the flow, l is large. While for a



Figure 6: A streamlined object results in large l (left), while a sharp corner results in small l (right).

sharp corner, which usually amplify vortex separation, l is small. Therefore, by introducing such a term, this effect can be approximated.

The results of our seeding method are illustrated in Figure 7. It can be seen that only a small number of DVM particles are needed to generate interesting wake patterns and that our method is sensitive to velocities and obstacle shapes. Instead of seeding one particle in the vortex center, we seed several particles uniformly distributed over the vortex to better resolve the flow field.

Finally, DVM particles are simply deleted when their vorticity or velocity is below some threshold or when they penetrate boundary.

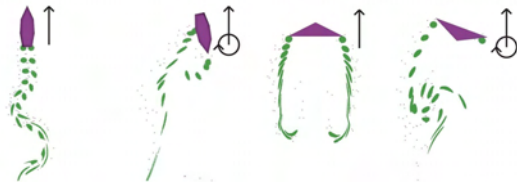


Figure 7: Using physically inspired seeding strategy, the result is under the control of obstacle shapes and velocities. And our method is able to reproduce interesting wake patterns, e.g. Karman vortex shedding. Green particles are tracers and others are DVM particles. The black arrow shows the velocity of each obstacle (purple region).

5.2. Coupling the Two Simulation

In order to use the above method to enhance the wake details in SWE, two way coupling between SWE and DVM is required. For SWE to DVM coupling, we simply interpolate SWE's velocity field to \vec{u}_∞ in Equation 11 as has been shown in Section 4.2. For DVM to SWE coupling, we take DVM's velocity field \vec{u}_V as an external force term for SWE, i.e. $\vec{a}_{ext} = \vec{u}_V \lambda$ in Equation 2. Where λ is the user controlled wake strength and \vec{u}_V is same as Equation 11. This treatment is analogous to the vorticity confinement method used in [SRF05]. Since no force is applied outside the DVM grid, one may expect to get smooth fading force by using larger grid. But we found that using a closely wrapping grid is enough.

5.3. Multiple Objects

In our method, DVM solvers are localized behind each rigid body. If multiple bodies exist, we choose to ignore the interaction between vortex particles emitted by different bodies in order to keep the DVM background grid local and small. Note that wakes can still interact with each other through SWE coupling, as illustrated in Figure 13.

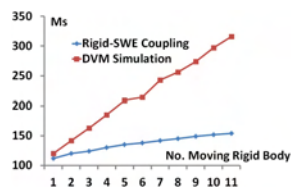


Figure 8: The cost of DVM solver is plotted against number of rigid bodies and compared with Rigid-SWE coupling required in standard SWE.

In fact, our method scale linearly with the number of moving rigid bodies, similar with [YHK07], as is shown in Figure 8 and the cost of DVM coupling is much less than rigid-SWE coupling.

6. Results

In our implementation, we use Bullet Physics Engine [Cou10] for rigid body simulation and collision detection. OpenMP [Ope02] is used for acceleration with three threads in all examples. Table 1 shows the cost of each substep. The extra computational cost induced by DVM is minor, compared with the SWE simulation. On average, the cost of DVM solver is only one-fifth that of the SWE solver. PovRay [Tea91] is used for rendering in all our examples.

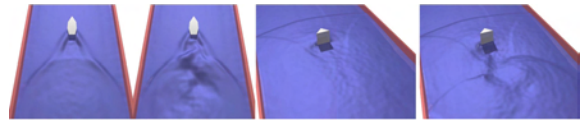


Figure 9: With the enhancement of DVM, our method can produce more details than pure SWE simulation, here $\mu = 0.02$, $\lambda = 1.8$.

Figure 9 show the final results of our method. Clearly, the method can greatly enrich wake details. In Figure 10, we drop six light boxes behind the boat. These boxes are then influenced by the wake flow and drift apart. In Figure 13, wake details created by five driving boats interact with each other through SWE coupling.

Scene	cost	FPS
Boat in Tank	74k/116/136/25	5
Wedge in Tank	150k/36/260/20	3
Dropping Box	74k/116/150/30	5
Multiple Objects	450k/750/764/136	1

Table 1: Scale and Cost of each step. From left to right: No. SWE Particles, No. DVM Particles, SWE Update time(ms) and DVM Update time(ms). SWE Update time includes two-way coupling. DVM Update time includes particle seeding, deleting and SWE-DVM coupling.

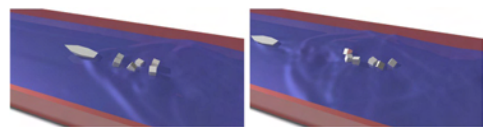


Figure 10: We drop several boxes in the wake region (left), these boxes will then drift apart with the wake flow (right), here $\mu = 0.02$, $\lambda = 1.8$.

Parameter Selection Since the vortices in DVM are transferred into SWE by coupling, we can set large damping on the vorticity of DVM and delete DVM particles earlier than in conventional DVM methods. After these particles are deleted, the vortices will be preserved in SWE for a while. These vortices are easily affected by the various parameters of the underly-

ing SWE simulation. In our experiment, fluid viscosity μ and coupling strength λ (Figure 11) are important parameters controlling the wake effects in the final result (refer to the accompanying video for more details). We find $\mu \in [0.02, 0.1]$ and $\lambda \in [0.0, 1.8]$ works well in all our tests.

On the other hand, the seeding of DVM particles is mainly controlled by the vortex radius R and potential threshold T . These parameters can be selected based on physical principles and used to enforce user control. For the potential threshold T , since vortices must not overlap, two vortices have to be separated by $2R$, so we can take $T = \exp(2R/M)$. Moreover, in different flow field, wake patterns may differ. We propose to use smaller R and M when viscosity is low, which will give turbulent flow patterns. While larger values will produce laminar patterns, suitable for highly viscous fluid. As mentioned before, as long as these parameters follow the above criteria, the results are always plausible.

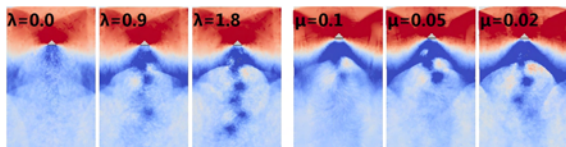


Figure 11: A wedge is moving in flow with heightfield rendered using pseudo-color. Left: Different wake patterns generated by lifting λ , using $\mu = 0.02$. Right: Different wake patterns generated by lowering μ , using $\lambda = 1.8$.

Grid-Based SWE In Figure 12, we couple our method with a grid-based SWE solver. For this solver, we follow [HHL*05] and [KP07]. The solver imposes little overhead compared with conventional SWE solvers used in [KM90] with much lower numerical viscosity. In fact, our method reaches real-time performance (19 FPS for Figure 12 with 100x600 cells) even on CPU with such a solver.

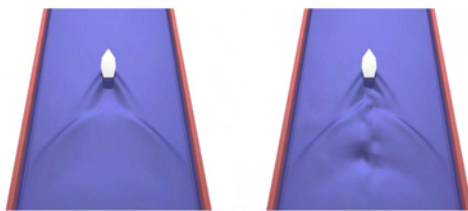


Figure 12: Applying our method to a grid-based solver. Detail enhancement can still be seen (right) compared with pure SWE (left).

7. Conclusion

In this paper, we provide an efficient method to synthesize and enhance the wake details for shallow water equation (SWE). This kind of simulation is frequently used in interactive applications and real-time games where a major concern is high performance. Therefore, we choose to overlay a discrete vortex method (DVM), which is known for its capability to generate rich details with very few particles. However, DVM par-

ticles should be properly placed to generate visually plausible wake patterns. To solve this problem, we propose a physically inspired particle seeding method, which is sensitive to object shapes and relative velocities against fluid. With this method, vortex particles are seeded only in the wake region, effectively concentrating the computational resources. Although direct interaction between DVM particles emitted by different objects is ignored, we show that wakes will still interact with each other through SWE coupling (Figure 13).

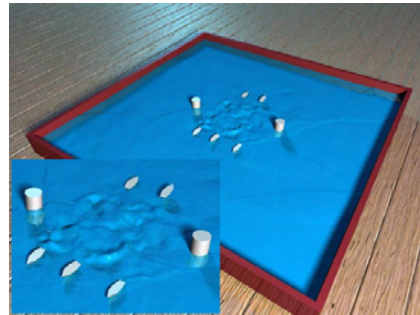


Figure 13: Multiple boats moving towards each other. Although we ignore interaction between DVM particles emitted by different boats, the wakes generated by these boats will still interact through SWE coupling.

Limitation and Future Work The major limitation of our method is that it does not include any 3D flow information to keep simulation cost low. In 3D, vortices under stretching and tilting may generate more interesting wake patterns. An additional limitation is that the method cannot handle fully submerged objects which will also generate wakes. We plan to address these issues in future research, and further improve the performance by utilizing GPU. Finally, some non-physically based assumptions are imposed by our method such as Equation 13 and 14. Therefore, we plan to model particle seeding more accurately by utilizing the ambient SWE main flow.

Acknowledgements

This work was partially supported by China 973 Program (No. 2009CB320801), NSFC (No. 61170139 and 60933007), the National High Technology Research and Development (863) Program of China (No. 2012AA011503), and Yiyong Tong was supported by NSF grants (IIS-0953096, CMMI-0757123 and CCF-0811313). We would also like to thank the anonymous reviewers for their valuable comments and suggestions.

References

- [AG91] ANDERSON C., GREENGARD C.: *Vortex dynamics and vortex methods*, vol. 28. Amer Mathematical Society, 1991. 4
- [AS05] ATA R., SOULAIMANI A.: A stabilized sph method for inviscid shallow water flows. *International journal for numerical methods in fluids* 47, 2 (2005), 139–159. 3
- [CLHM97] CHEN J., LOBO N., HUGHES C., MOSHELL J.: Real-time fluid simulation in a dynamic virtual environment. *Computer Graphics and Applications, IEEE* 17, 3 (1997), 52–61. 2

- [CM10] CHENTANEZ N., MÜLLER M.: Real-time simulation of large bodies of water with small scale details. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2010), Eurographics Association, pp. 197–206. 2
- [CM11] CHENTANEZ N., MÜLLER M.: Real-time eulerian water simulation using a restricted tall cell grid. In *ACM SIGGRAPH* (2011), vol. 30, ACM, p. 82. 2
- [Cou10] COUMANS E.: *Bullet physics engine*, 2010. 6
- [FF01] FOSTER N., FEDKIW R.: Practical animation of liquids. In *ACM SIGGRAPH* (2001), ACM, pp. 23–30. 2
- [FM96] FOSTER N., METAXAS D.: Realistic animation of liquids. *Graphical models and image processing* 58, 5 (1996), 471–483. 2
- [FSJ01] FEDKIW R., STAM J., JENSEN H.: Visual simulation of smoke. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (2001), ACM, pp. 15–22. 2
- [GJK88] GILBERT E., JOHNSON D., KEERTHI S.: A fast procedure for computing the distance between complex objects in three-dimensional space. *IEEE Journal of Robotics and Automation* 4, 2 (1988), 193–203. 5
- [GLG*95] GAMITO M., LOPES P., GOMES M., ET AL.: Two-dimensional simulation of gaseous phenomena using vortex particles. In *Proceedings of the 6th Eurographics Workshop on Computer Animation and Simulation* (1995), pp. 3–15. 2
- [GR87] GREENGARD L., ROKHLIN V.: A fast algorithm for particle simulations. *Journal of computational physics* 73, 2 (1987), 325–348. 4
- [HA97] HANSEN E., ARNEBORG L.: The use of a discrete vortex model for shallow water flow around islands and coastal structures. *Coastal engineering* 32, 2-3 (1997), 223–246. 2
- [HHL*05] HAGEN T., HJELMERVIK J., LIE K., NATVIG J., OFSTAD HENRIKSEN M.: Visual simulation of shallow-water waves. *Simulation Modelling Practice and Theory* 13, 8 (2005), 716–726. 7
- [HK05] HONG J., KIM C.: Discontinuous fluids. In *ACM SIGGRAPH* (2005), vol. 24, ACM, pp. 915–920. 2
- [IGLF06] IRVING G., GUENDELMAN E., LOSASSO F., FEDKIW R.: Efficient simulation of large bodies of water by coupling two and three dimensional techniques. *ACM SIGGRAPH* 25, 3 (2006), 805–811. 2
- [KM90] KASS M., MILLER G.: Rapid, stable fluid dynamics for computer graphics. In *ACM SIGGRAPH* (1990), vol. 24, ACM, pp. 49–57. 2, 7
- [KP07] KURGANOV A., PETROVA G.: A second-order well-balanced positivity preserving central-upwind scheme for the saint-venant system. *Communications in Mathematical Sciences* 5, 1 (2007), 133–160. 7
- [LGF04] LOSASSO F., GIBOU F., FEDKIW R.: Simulating water and smoke with an octree data structure. In *ACM SIGGRAPH* (2004), vol. 23, ACM, pp. 457–462. 2
- [LH10] LEE H., HAN S.: Solving the shallow water equations using 2d sph particles for interactive applications. *The Visual Computer* 26, 6 (2010), 865–872. 2, 3
- [Lia08] LIANG S.: A least-squares finite-element method for shallow-water equations. In *OCEANS 2008-MTS/IEEE Kobe Techno-Ocean* (2008), IEEE, pp. 1–9. 1
- [LK01] LINDSAY K., KRASNY R.: A particle method and adaptive treecode for vortex sheet motion in three-dimensional flow. *J. Comput. Phys* 172 (2001), 879–907. 4
- [LO07] LEE R., O’SULLIVAN C.: A fast and compact solver for the shallow water equations. *Virtual Reality Interactions and Physical Simulation I*, 1 (2007), 51–58. 2
- [LSSF06] LOSASSO F., SHINAR T., SELLE A., FEDKIW R.: Multiple interacting liquids. In *ACM SIGGRAPH* (2006), vol. 25, ACM, pp. 812–819. 2
- [MCG03] MÜLLER M., CHARYPAR D., GROSS M.: Particle-based fluid simulation for interactive applications. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation* (2003), Eurographics Association, pp. 154–159. 2, 3
- [NSCL08] NARAIN R., SEWALL J., CARLSON M., LIN M.: Fast animation of turbulence using energy transport and procedural synthesis. *ACM Transactions on Graphics (TOG)* 27, 5 (2008), 166. 2
- [OH95] O’BRIEN J., HODGINS J.: Dynamic simulation of splashing fluids. In *Computer Animation’95., Proceedings.* (1995), IEEE, pp. 198–205. 2
- [Ope02] OPENMP C.: C++ application program interface, 2002. 6
- [PK05] PARK S., KIM M.: Vortex fluid for gaseous phenomena. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation* (2005), ACM, pp. 261–270. 2, 3, 4
- [Pop00] POPE S.: *Turbulent flows*. Cambridge Univ Pr, 2000. 2
- [PTSG09] PFAFF T., THUREY N., SELLE A., GROSS M.: Synthetic turbulence using artificial boundary layers. In *ACM SIGGRAPH Asia* (2009), vol. 28, ACM, p. 121. 2, 5
- [SB08] SCHECHTER H., BRIDSON R.: Evolving sub-grid turbulence for smoke animation. In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics symposium on Computer animation* (2008), Eurographics Association, pp. 1–7. 2
- [SBC*11] SOLENTHALER B., BUCHER P., CHENTANEZ N., MÜLLER M., GROSS M.: Sph based shallow water simulation. In *Workshop in Virtual Reality Interactions and Physical Simulation* (2011), The Eurographics Association, pp. 39–46. 2, 3
- [SG00] SCHLICHTING H., GERSTEN K.: *Boundary-layer theory*. Springer Verlag, 2000. 4
- [SRF05] SELLE A., RASMUSSEN N., FEDKIW R.: A vortex particle method for smoke, water and explosions. In *ACM SIGGRAPH* (2005), vol. 24, ACM, pp. 910–914. 2, 4, 6
- [Sta99] STAM J.: Stable fluids. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques* (1999), ACM Press/Addison-Wesley Publishing Co., pp. 121–128. 2
- [Tea91] TEAM P.: Persistency of vision ray tracer (pov-ray), 1991. 6
- [TMFSG07] THUREY N., MULLER-FISCHER M., SCHIRM S., GROSS M.: Real-time breaking waves for shallow water simulations. In *Computer Graphics and Applications, 2007. PG’07. 15th Pacific Conference on* (2007), IEEE, pp. 39–46. 2
- [TSS*07] THÜREY N., SADLO F., SCHIRM S., MÜLLER-FISCHER M., GROSS M.: Real-time simulations of bubbles and foam within a shallow water framework. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation* (2007), Eurographics Association, pp. 191–198. 2
- [YHK07] YUKSEL C., HOUSE D., KEYSER J.: Wave particles. In *ACM SIGGRAPH* (2007), vol. 26, ACM, p. 99. 2, 3, 6
- [YYF12] YUAN Z., Y. Z., F. C.: Incorporating stochastic turbulence in particle-based fluid simulation. *The Visual Computer* 28, 5 (2012). 3
- [ZYF10] ZHU B., YANG X., FAN Y.: Creating and preserving vortical details in sph fluid. In *Computer Graphics Forum* (2010), vol. 29, Wiley Online Library, pp. 2207–2214. 3