Article

Visual Simulation of Detailed Turbulent Water by Preserving the Thin Sheets of Fluid

Jong-Hyun Kim ¹ , Wook Kim ², Young Bin Kim ³ and Jung Lee ^{4,*}

¹ Department of Software Application, Kangnam Univers, Yongin 17021, Gyeonggi-do, Korea; jonghyunkim@kangnam.ac.kr

² Department of Computer Engineering, Korea University, Seoul 03909, Korea; marverous@naver.com

³ Graduate School of Advanced Imaging Science, Multimedia & Film, Chung-Ang University, Seoul 03909, Korea; ybkim85@cau.ac.kr

⁴ Department of Convergence Software, Hallym University, Chuncheon 24268, Korea

* Correspondence: airjung@hallym.ac.kr

Received: 19 September 2018; Accepted: 2 October 2018; Published: 15 October 2018

Abstract: When we perform particle-based water simulation, water particles are often increased dramatically because of particle splitting around breaking holes to maintain the thin fluid sheets. Because most of the existing approaches do not consider the volume of the water particles, the water particles must have a very low mass to satisfy the law of the conservation of mass. This phenomenon smears the motion of the water, which would otherwise result in splashing, thereby resulting in artifacts such as numerical dissipation. Thus, we propose a new fluid-implicit, particle-based framework for maintaining and representing the thin sheets and turbulent flows of water. After splitting the water particles, the proposed method uses the ghost density and ghost mass to redistribute the difference in mass based on the volume of the water particles. Next, small-scale turbulent flows are formed in local regions and transferred in a smooth manner to the global flow field. Our results show us the turbulence details as well as the thin sheets of water, thereby obtaining an aesthetically pleasing improvement compared with existing methods.

Keywords: turbulent fluids; water simulations; liquid sheets

1. Introduction

In the fields of smoke, fire, and water simulation, many techniques have been studied to improve the visualization detail of surfaces and flows. Expression of turbulence and vorticity in fluid flow is very important in Computer Graphics. Chen et al. [1] synthesized turbulence using a hybrid method. Zhao et al. [2] proposed random forcing method on an upsampled grid to synthesize fluid turbulence. The original vortex particles method [3] was extended by Yoon et al. [4]. The vortex particles are applied onto an upsampled grid with higher resolution. Kim et al. [5] proposed a method for buoyant sources by extending the vortex particle approach. They computed the source terms on the grid and the computed terms are mapped to the vortex particle system. Weissmann et al. [6] simulated vorticity using vortex filaments entirely, and source terms are used for obstacle-induced shedding. Barnat and Pollard [7] expressed vortex filaments based on Weissmann et al.'s method by using an improved re-sampling scheme.

The representation of turbulence details that arise from interactions between fluid and solid objects or large fluid bodies is a prevalent topic in computer graphics. In particular, the generation and preservation of small-scale details is an important aspect when animating turbulent fluids. Thus, physics-based [5,8–11], noise-based, and texture-based [12–14] methods have been proposed for use in fluid simulations. There are adaptive-grid and multigrid methods for turbulent flows around solid

objects [15–17] and self-turbulent flows [18,19]. It is possible to temporarily model the generation and evolution of small-scale vortices using the Navier–Stokes equations locally with a higher grid resolution. However, it becomes difficult to preserve the turbulence details if the vorticities leave the high-resolution region. Thus, various vortex particle methods [3,5] have been proposed based on the hybrid particle-grid framework to effectively express small-scale details of fluid.

In computer graphics, most existing particle-based fluid methods have tried to capture the details on the free surfaces of water, e.g., self-turbulent flows generated by a strong vortex [12,13,18,19] and turbulent flows around an object [15–17,20]; in contrast, this paper focuses on synthesizing the missing turbulence details while still preserving the fluid sheets. Recently, methods have been investigated for conserving thin fluid sheets, which represent small-scale details [21–23]. However, previous studies used methods that enforced mass conservation even when the number of water particles increased during the representation of thin fluid sheets [21,22]. Thus, they often assigned small mass values to regions with geometrically large volumes, so the stormy turbulent flows that are assumed to emerge in water simulation were oversmoothed because of the resulting low density (see Figure 1).

Figure 1. Streams of falling water represented by our method. The fluid sheets of water are expressed visually and the turbulent details are captured around the fluid sheets, thereby obtaining improved results compared with existing methods. (a) Existing method [21,22]; (b) Proposed method; and (c) Proposed method—animation sequence.

To the best of our knowledge, modeling realistic turbulence formation and preserving thin fluid sheets in the global flow using a single method is still a challenging problem. In this study, we present a new method that achieves both of these goals in a particle-based fluid framework. Our method improves the simulation of turbulent water phenomena in a particle-based water simulator while using a fluid sheets solver. Small-scale turbulences are generated and maintained in local regions by adjusting the uneven mass distribution according to the volumes of the water particles. The creation of local turbulence and the preservation of thin sheets are combined in a novel manner. Thus, the ghost density of all the water particles, including newly inserted thin particles, is calculated to maintain the fluid sheets while preserving the turbulence details. The calculated ghost density of the increased water particles depends on the number of water particles that exist before checking the conservation of mass. To overcome any unsuccessful conservation of mass caused by the increased number of water particles, the ghost density is used to ensure mass conservation. Namely, the mass is geometrically

redistributed according to the distribution of the water particles; thus, the turbulence details are preserved near the fluid sheets.

Problem Statement. In previous methods for preserving fluid sheets [21,22], every attribute except for the mass is linearly interpolated when the particles split. When splitting occurs to obtain a pair of particles, (i, j) , the new mass is given by $m = (m_i + m_j)/3$. The new masses of particles i and j decrease to $m_i^{new} = \frac{2}{3}m_i$ and $m_j^{new} = \frac{2}{3}m_j$ after the split. The fluid velocity, u , and particle density, ρ , on the grid are affected by the changes in mass.

As shown in Figure 2, the water particles split near the fluid sheets to cover the breaking holes. From a visual perspective, the quality of the water surface increases but the mass of the water particles near the thin sheets decreases because of the aggressive mass conservation during the split. As the fluid sheets grow, the water particles split further and their masses decrease. As shown in the inset in Figure 2, the fluid motion appears to be oversmoothed around the region with turbulent motion, which is caused by the interaction between the fluid sheets and underlying water. To address this issue, we propose a fast and efficient method for synthesizing the missing water turbulence near the thin fluid sheets while still maintaining the conservation of mass. Hence, the proposed method obtains more aesthetically pleasing results compared with existing methods.

Figure 2. Problem with a previously proposed thin sheet preservation algorithm [21,22] (scalar value: Mass of particles; resting mass: 1.0; gray rectangle: Interacting region between the underlying flow and thin fluid sheet areas; arrow: Velocity of particle).

2. Related Work

In this section, we briefly review previous researches related to the central issues addressed in this study, i.e., turbulence details and thin sheet details of fluids.

Turbulence Details. Stam [24] proposed an unconditionally stable semi-Lagrangian method for turbulence details that is used widely in the fluid simulation despite numerical dissipation. To express the small-scale features of smoke and overcome numerical dissipation, Fedkiw et al. [25] proposed the vorticity confinement method. However, it cannot calculate the amount of confinement force to be injected without causing instability. To ensure the stability, He et al. [26] proposed the adaptive vorticity confinement method by adjusting the coefficients for vorticity according to the vorticity magnitude.

To express the details of turbulence in smoke simulations, the vortex particle method is often used. To compute the velocity of turbulence, Park and Kim [27] solved the vorticity-related Poisson equation for a stream function. They evaluated the velocity field by tracking the gradient of the stream function. To express the details of turbulent flow, Selle et al. [3,5] integrated vortex particles using background grids. Angelidis and Neyret [28,29] used a vortex structure based on filament primitives to express the detailed features. Pfaff et al. [11] proposed a vortex sheet method being applied on a high-resolution triangular mesh to synthesize the detailed turbulent flow between interfaces.

Some previous studies have used high-level turbulence models to express subgrid details. For example, Schechter and Bridson [30] captured energy cascading with a subgrid turbulence model. Kim et al. [13] expressed the turbulent flow using combination of the curl-noise function and a Kolmogorov wavelet form. Pfaff et al. [14] expressed the turbulence features near the boundaries of obstacles using a new energy model. They simulated vortex shedding using the distributed vortex particles near the boundary layers. They also extended this method to simulate anisotropic details of turbulence [12]. Mauricio et al. [31] accurately simulated vortex shedding by creating the vorticity directly on solid surfaces without modeling the boundary. Kim et al. [9] also expressed the stretching and wiggling motions of liquid with vortex stretching. Hong et al. [8] expressed the cellular patterns and wrinkling on the surface of a flame using the combination of detonation shock dynamics and the Navier–Stokes equation. Recently, Mercier et al. [10] substantially enhanced the details of the turbulent surface by adding a wave simulation to water surfaces. However, this method has been suffered from numerical dissipation causing the missing turbulence because it only improves the details of the original water surface just like a bump mapping method. The method adds a wave simulation near the surface particle to express the motion of the water including the valley with strong flow or turbulence. Advecting particles in this way makes it difficult to represent thin sheets or numerically lost turbulence due to motion dependent on the wave simulation, so it is not appropriate to apply directly to our method.

Thin Sheet Details. The level-set method [32] is frequently used in the field of fluid simulation. In this method, the distance to the nearest surface, which is defined implicitly as a zero contour, is saved for each node. This method has been refined to obtain many other novel approaches; for example, Enright et al. [33], Wang et al. [34], and Mihalef et al. [35] attempted to reduce the numerical loss using Lagrangian particle approaches.

Bargteil et al. [36] increased the accuracy of fluid simulations by improving the signed distance field calculated from the reconstructed surface. Heo and Ko [37] preserved the details of thin surfaces using the spectrally refined level set and the high-order reinitialization method. However, these methods are insufficient for fully preserving the ruptures on a thin surface. As an alternative approach for solving this issue, mesh-based surface tracking has been explored in the fields of medical image analysis and fluid dynamics [23,38,39]. In these methods, explicit surface elements are typically advected via underlying flows to represent the thin features, but it is still difficult to address self-intersections and the complex topology changes that occur near interfaces. These problems can be temporarily addressed by resolving only the colliding parts into resampling meshes, but the algorithm becomes highly complex because of the diversity of possible situations.

There are various methods for reconstructing surfaces from the point cloud [40–43] used in particle-based fluid simulation methods, the fluid surfaces are defined implicitly based on the distances among particles and their density. However, these methods obtain poor results and blobby ruptures when the particles are sparse. Recently, Ando and Tsuruno [22] proposed an algorithm for preserving thin sheets in a fluid-implicit particle (FLIP)-based water simulation. In this method, particles are inserted into breaking holes to fully restore the thin sheets. They also extended this method to adaptively sample anisotropic particles and represent thin sheets more efficiently [21]. These methods well represent thin sheets of fluid. However, in the process of splitting water particles, inappropriate mass redistribution criterion causes water particles to have too small a mass, resulting in the problem of turbulence being lost.

3. Preserving the Thin Sheet and Turbulence Details of Water

In the proposed method, we generate thin fluid sheets and turbulence details using three-dimensional water particles with an existing particle-based water solver, i.e., FLIP [42]. Furthermore, we express the detailed turbulent flow around fluid sheets. This algorithm operates as follows.

1. Water particles are advected and their density is calculated.
2. Singular value decomposition (SVD) is used to extract thin particles from water particles.

3. The distances and relative velocities between thin particle pairs are used to find candidate positions.
4. The candidate positions are used to insert and break new water particles.
5. To restore the missing turbulence, the ghost density is calculated for all of the water particles, including the newly inserted water particles. The ghost mass is then calculated based on the density to ensure the conservation of mass.
6. The ghost mass is used to rasterize the velocities of particles on an Eulerian grid and advect the water particles using a FLIP solver.
7. The fluid surfaces are reconstructed.

3.1. Preserving Thin Sheets

3.1.1. Extracting Thin Particles

The proposed method is based on Ando's methods [21,22] for turbulent flow modeling, which we briefly review here. First, the particle distribution is analyzed using anisotropic kernels. The particle directions are then determined by calculating the weighted average covariance per particle, C_i , as follows:

$$C_i = \frac{\sum_j (\mathbf{p}_j - \bar{\mathbf{p}}_i)(\mathbf{p}_j - \bar{\mathbf{p}}_i)^T W_{smooth}(\mathbf{p}_j - \bar{\mathbf{p}}_i, \alpha_1 d_0)}{\sum_j W_{smooth}(\mathbf{p}_j - \bar{\mathbf{p}}_i, \alpha_1 d_0)}, \quad (1)$$

where d_0 denotes the initial space between particles, α_1 denotes a scaling constant for distance d_0 ,

$$\bar{\mathbf{p}}_i = \frac{\sum_j \mathbf{p}_j W_{smooth}(\mathbf{p}_j - \mathbf{p}_i, \alpha_1 d_0)}{\sum_j W_{smooth}(\mathbf{p}_j - \mathbf{p}_i, \alpha_1 d_0)}, \quad (2)$$

and

$$W_{smooth}(\mathbf{r}, h) = \begin{cases} 1 - \|\mathbf{r}\|^2 / h^2, & 0 \leq \|\mathbf{r}\| \leq h, \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

The covariance matrices, C_i , calculated using the formulas above are used to obtain the eigenvectors and eigenvalues by SVD, as follows, from which the stretch and direction among adjacent particles are extracted:

$$C_i = \begin{bmatrix} e_1^T \\ e_2^T \\ e_3^T \end{bmatrix}^T \begin{bmatrix} \sigma_1 & & \\ & \sigma_2 & \\ & & \sigma_3 \end{bmatrix} \begin{bmatrix} e_1^T \\ e_2^T \\ e_3^T \end{bmatrix}, \quad (4)$$

where e_n denotes the principal axes ordered by the variance and σ_n denotes the degree of stretch. The thin particles are extracted by examining $\sigma_3 \leq \alpha_2 \sigma_1$, where α_2 denotes a threshold that controls the degree of thinness. Figure 3 shows the selected thin particles, and the difference between the eigenvalues increases as the water particles collide with the ground and spread out. As a result, the more strongly the particles are spread, the wider the selection range of the thin particles.

Figure 3. Selected thin particles from water particles (thin particles are colored violet and lie on the outer edge of the fluid body).

3.1.2. Finding Candidate Positions

The extracted thin particles are used to add particles to the breaking holes. We look for pairs of particles, (i, j) , that connect the breaking holes. The midpoint of a pair, $(\mathbf{p}_i + \mathbf{p}_j)/2$, is considered as a candidate position for splitting. We select a pair that satisfies all of the following conditions:

$$\alpha_3 d_0 \leq \|\mathbf{p}_i - \mathbf{p}_j\| \leq \alpha_4 d_0 \quad (5)$$

$$\sum_k W_{smooth}((\mathbf{p}_i + \mathbf{p}_j)/2 - \mathbf{p}_k, \alpha_3 d_0) = 0 \quad (6)$$

$$(\mathbf{p}_i - \mathbf{p}_j) \cdot (\mathbf{u}_i - \mathbf{u}_j) > 0 \quad (7)$$

where α_3 and α_4 denote constants that determine the minimum and maximum distances between candidate positions, respectively, and u is the velocity of a particle. Equation (5) checks whether the two particles are located at an appropriate distance from each other. Equation (6) checks whether the candidate midpoints are sparsely distributed to add a new particle at radius α_3 . Equation (7) checks whether the pair of particles are moving away from each other over time. After these pairs have been identified, the midpoints are placed into list $\mathbf{S}$ without duplication.

3.1.3. Inserting and Removing Particles

When the candidate positions are very dense, they are not suitable for inserting new particles. In this section, we describe how to filter unnecessary particles and insert a minimal number of particles. $\mathbf{I}$ is the final list of the indices of inserted candidates. In this list, the first particle, I_1 , is the most remote candidate within $\mathbf{S}$:

$$I_1 = \underset{j \in \text{size} \mathbf{S}}{\operatorname{argmin}} \rho(\mathbf{s}_j), \quad (8)$$

where ρ_j is the density of particle j . After determining I_1 , from $\mathbf{S}$, we omit every nearby candidate around $\mathbf{p}_{I_1}$ that exists within a radius of $\alpha_3 d_0$. In the next step, we find nearby particles around $\mathbf{p}_{I_1}$ in $\mathbf{S}$ that exist beyond a radius of α_3 and add them to the list $\mathbf{N}_{I_1}$. If we find a particle, the closest candidate is inserted within $\mathbf{N}_{I_1}$:

$$I_2 = \underset{j \in \mathbf{N}_{I_1}}{\operatorname{argmin}} \|\mathbf{s}_j - \mathbf{s}_{I_1}\|. \quad (9)$$

If the search is formulated as $I_{n+1} = \text{search}(I_n)$, which means that I_2 can be found in I_1 , then this task can be repeated to find $I_3, I_4, \dots, I_{n+1}$ in a serial manner. If the next search fails, Equation (8) is used instead.

As mentioned in the problem statement, the attributes of particles are subject to interpolation after splitting. Thus, the following formula is used to calculate the mapped velocity, u , and particle density, ρ , in the Eulerian grid:

$$u(\mathbf{x}) = \frac{\sum_i m_i \mathbf{u}_i W_{sharp}(\mathbf{p}_i - \mathbf{x}, \alpha_5 d_0)}{\sum_i m_i W_{sharp}(\mathbf{p}_i - \mathbf{x}, \alpha_5 d_0)}, \quad (10)$$

$$\rho(\mathbf{x}) = \sum_i m_i W_{smooth}(\mathbf{p}_i - \mathbf{x}, \alpha_1 d_0), \quad (11)$$

where α_5 denotes a scaling constant for velocity u , and:

$$W_{sharp}(\mathbf{r}, h) = \begin{cases} h^2 / \|\mathbf{r}\|^2 - 1, & 0 \leq \|\mathbf{r}\| \leq h, \\ 0, & \text{otherwise.} \end{cases} \quad (12)$$

In contrast, inserted particle i is removed if it satisfies either of the following conditions:

$$\rho_i > \alpha_6 \rho_0 \text{ and } \sigma_3 \geq \alpha_2 \sigma_1, \quad (13)$$

$$\|\mathbf{p}_i - \mathbf{p}_j\| < \alpha_7 d_0, \quad \text{for any } j \neq i, \quad (14)$$

where α_6 denotes the maximum density coefficient and α_7 denotes the minimum distance between particles. A particle is removed if it satisfies Equations (13) and (14). When a particle is removed, its mass is added back to the source particle before splitting.

3.2. Synthesizing Turbulence Details

The algorithm described in previous section fully represents the fluid sheets at breaking holes. However, the original mass of the water particles is distributed by aggressive conservation of mass whenever the particles split. A problem with this method is that the frequent splitting of water particles near the fluid sheets significantly decreases their mass. Thus, if the fluid sheets are larger, the water particles split more frequently and the mass values decrease. To illustrate this issue, the fluid sheets generated by stretching and their masses are colored in Figure 4, which shows that the mass value is smaller in the center because of the continuous insertion of thin particles caused by stretching. As shown in Figure 4a, the total mass value is increased due to the newly inserted orange particles, and the mass value around the orange particles is kept small in order to satisfy the mass conservation condition. In Figure 4b, blue particles have mass values close to zero, but they are expressed as a voluminous fluid surface, often resulting in awkward results.

Figure 4. Visualization of stretching thin sheet particles and their masses. (a) Thin sheet particles (orange: Inserted particles); (b) Mass of particles (red: High; blue: Low).

Therefore, the stormy turbulence of water, which occurs when the fluid sheets collide with the underlying water (see Figure 5a), cannot be represented. In existing methods, the region where fluid sheets are generated has a smaller mass (see blue particles in Figure 5a), so any interactions with the underlying water barely change the mass, thereby hindering the generation of turbulence details and leading to oversmoothing. In contrast, the proposed method accurately captures the variations in mass at the point of interaction (see Figure 5b). Though the water particles composed of thin sheet particles are clustered, the intensity of the motion is weak because of their small mass values. However, due to the mass redistribution, our results well represent the motion of the water due to the interaction. We introduce ghost density and ghost mass to address this challenge. First, to calculate the mass while considering the newly inserted water particles, the ghost density ρ^G is obtained as follows:

$$\rho^G(\mathbf{x}) = \sum_i W_{smooth}(\mathbf{p}_i - \mathbf{x}, \alpha_5 d_0), \quad (15)$$

where the mass term is omitted to determine the density according to the distribution of water particles. Thus, Equation (15) yields a weight that is used to provide a larger mass to the region where many particles are distributed geometrically. This value is then used to calculate the ghost mass m^G :

$$\sigma^{target} = \sum_i m_{0,i}, \quad (16)$$

$$\rho_i^{G,scale} = \frac{\rho_i^G}{\max(\rho_i^G, \dots, \rho_{n-1}^G)}, \quad (17)$$

$$\rho^{sum G,scale} = \sum_i^n \rho_i^{G,scale}, \quad (18)$$

$$m^G = \frac{\rho^{G,scale}}{\rho^{sum G,scale}} \sigma^{target}, \quad (19)$$

where m_0 denotes the initial mass of particles, which is set at 1.0, and σ^{target} is a value used for the conservation of mass. Therefore, the conservation of mass is maintained through σ^{target} during the simulation even though the number of particles changes. Equations (16)–(19) are used to distribute the mass value according to the distribution of water particles, thereby assigning a relatively larger mass to regions where water particles are congested; thus, the mass near the fluid sheets will not differ significantly. This distribution method aims to avoid ignoring the motions of particles with relatively low masses when the difference in the mass of the water particles is significant. The ghost density and ghost mass facilitate the robust restoration of turbulent water flows, which are blurred by previously proposed methods.

Ando's method simply splits water particles to represent only thin sheets shapes [21,22]. The mass redistribution criteria used in the method cause turbulence to disappear in areas where thin sheets are interacted with the underlying water or coupled to the object. To reduce this problem, using the averages that you have suggested can be an alternative. However, it is likely that the turbulence flow will not be restored properly because only the mass of newly added thin sheets particles is updated in the thin sheets area. Equations (16)–(19) are formulas that redistribute mass while satisfying mass-conservation using the total number of particles including particles used to represent thin sheets.

Figure 6 shows the distributed masses of the water particles obtained using the existing approach and the proposed method. The existing method simply distributes the original mass during the splitting process, so the thin particles have relatively low masses (see the aqua line in the figure). In contrast, the proposed method obtains refined masses by redistributing the mass according to the density of the water particles, thereby synthesizing the mass of the reduced thin particles (see the grey line in the figure). In the chart above, the 4th, 5th, 6th, and 7th particles in aqua color have relatively small mass values because they are particles split around thin sheets. Because this problem dissipates turbulence, the ghost density ρ^G using Equation (15) and the mass-conservation Equation (15) are used to redistribute the mass. As a result, we avoided the problem of allocating a small mass value by distributing the mass at the density based on the number of particles, rather than simply splitting it. In the above chart, the mass value of the 4th, 5th, 6th, and 7th particles marked with grey color is increased because the number of particles near the thin sheets is relatively larger than that of the other regions.

In order to represent the turbulence of the lost water, the mass value is redistributed adaptively, but the total mass of the fluid is not changed by Equation (16–19). In summary, while retaining the total mass of the fluid, it restores turbulence that has been lost only by local changes in the mass ratio. Figure 6 shows an example of changing the mass ratio of each particle while maintaining the total mass of the mass. Figure 6 is an intuitive illustration of the change in mass, and the results of the actual algorithm can vary. In Figure 5, the turbulence is lost in the aqua line in Figure 6, and the restored turbulence is the grey line in Figure 6.

In the proposed method, m^G is used when rasterizing the velocity of water particles in the Eulerian grid and calculating the density of the water particles. According to this formula, stormy turbulence is well-represented when the fluid sheets collide with water, as shown in Figure 5b.

Figure 5. Thin sheets and water-particle mass represented using the existing approach and the proposed method. Particles are colored according to their mass (red: High; blue: Low). (a) Existing method [21,22]; (b) Proposed method.

Figure 6. Corrected masses of water particles (inserted thin particles: 4, 5, 6, 7; resting mass: 1.0; aqua: Previous method [21,22]; grey: Proposed method).

4. Implementation

The proposed method was implemented using an Intel i7-2600k 3.40 GHz CPU with 16 GB RAM and an NVIDIA GeForce GTX 480 graphics card. A FLIP-based [22] water simulation was executed for the underlying water, where every step of the water solver was performed in parallel.

All the physical quantities were stored for the FLIP grid using the staggered marker-and-cell method [44]. An additional grid was employed for surface reconstruction. We compute the level-set values from the simulated water particles [45]. Next, the triangular meshes were reconstructed using the marching cubes algorithm [46]. A variational framework was used for the water-solid boundary conditions [47]. The water-air interface was approximated considering the minimum distance from the air cell center, $\mathbf{x}_{air}$ to each particle position $\mathbf{x}$ in the water cell, $\min(\|\mathbf{x} - \mathbf{x}_{air}\| - r)$, where r denotes the radius of a particle. We used a preconditioned conjugate gradient algorithm to solve this linear system.

5. Results

We tested various scenarios—dam breaking, a sphere dropping onto underling water, a rotating box in a pool of water, and a water box falling over a box cliff—to verify the preservation of details for thin sheets and turbulent flow while using the proposed method.

Figures 7 and 8 show the visual improvement obtained by using the proposed method in the dam breaking simulation. The proposed method successfully generated smooth thin sheets and a dynamic water surface. The existing method [21,22] smoothed the water features around the thin sheets (see Figure 7a), whereas the proposed method achieved a better representation of the thin sheets and captured the water surfaces in more detail (see Figure 7b). As you can see, thin sheet surfaces, represented by the addition of thin sheet particles, can represent a clear fluid surface with no holes. However, turbulence around the thin sheet surface is lost because the mass value of the particle converges to a value close to zero. As a result, the smoothing surface is represented in Figure 7a, and our results show that we have successfully recovered the lost turbulence.

Figure 8 shows a comparison of scenes where the thin sheets interacted with the underlying water. The existing method provided a good representation of the fluid sheets, but the turbulent details emerging around the thin sheets were not represented. As shown in Figure 8a, no secondary motion could be seen despite the interaction, whereas the proposed method represented both the fluid sheets and turbulent details simultaneously (see Figure 8b).

Figure 7. Dam breaking test. Detailed water features are captured around the fluid sheets. (a) Existing method [21,22]; (b) Proposed method.

Figure 8. Turbulent flow around fluid sheets (right: zoomed-in images). (a) Existing method [21,22]; (b) Proposed method.

Figure 9 shows the simulation results obtained via the proposed method. In this simulation, a water ball was dropped into a shallow water pool. Observing the particle views shown in the inset image of Figure 9, the proposed method nearly resolved the numerical dissipation problem caused by mass redistribution, as the particles were advected more turbulently. The advected particles directly affect the turbulence details on the water surfaces, so the visual output was improved.

Figure 10 shows a stirring box rotating continuously in a pool of water. Observing Figure 10a, the existing method failed to geometrically redistribute the mass according to the amount of particles; thus, the turbulent flows generated by the thin fluid sheets were missing, whereas the proposed method yielded improved turbulent flow. Figure 11 shows a box of water falling over the edge of a cliff. It should be noted that the jumping water sheets moving from the cliff edge hit the pool without any rupture. The proposed method greatly improved the generation of water sheets and turbulent splashing.

These applications to different scenes demonstrate that the proposed method performs well in terms of thin water sheets and turbulence details by redistributing the mass value so it visually decreases, as well as rectifying the awkward motions of thin sheets vanishing into water. The proposed method incurs a slightly greater computational cost compared with existing methods, but it produces a more detailed simulation of water sheets and turbulence. The parameters and configurations in this paper are summarized in Tables 1 and 2.

Figure 9. Visualization of particle masses during water splashing using proposed method (inset image: Particle view, particles are colored according to their mass).

Figure 10. Simulation results obtained for a water pool affected by a stirring box. (a) Existing method [21,22]; (b) Proposed method.

Figure 11. Water box falling over a box cliff. (a) Existing method [21,22]; (b) Proposed method.

Table 1. Parameters used in this paper. Specific parameters generate the example animations presented in this study.

Symbol	Description	Value
α_1	Density radius scale	4.0
α_2	Thin sheet rate	0.2
α_3	Minimum insertion distance	0.8
α_4	Maximum insertion distance	3.5
α_5	Velocity radius scale	1.0
α_6	Maximum thin particle density	0.2
α_7	Maximum thin particle distance	0.2
Δt	Time-step	0.006

Table 2. Sizes of the example scenes presented in this study (Num.: Number, Res.: Resolution, dx : Grid point spacing).

Figure	Num. of Water Particles	FLIP Solver Grid Res.	Surface Reconstruction Grid Res.	Pressure Solver Grid Res.	dx
1	322,454	100 × 100 × 100	256 × 256 × 256	100 × 100 × 100	0.01
4	45,200	150 × 150 × 150	–	150 × 150 × 150	0.006
7,8	1,802,626	128 × 128 × 128	256 × 256 × 256	128 × 128 × 128	0.007
9	843,512	128 × 128 × 128	256 × 256 × 256	128 × 128 × 128	0.007
10	3,079,593	180 × 180 × 180	180 × 180 × 180	180 × 180 × 180	0.005
11	498,143	100 × 100 × 100	256 × 256 × 256	100 × 100 × 100	0.01

6. Discussion and Additional Explanation

In this paper, we deal with the numerical dissipation problem occurring in the process of discretizing and numerically modeling Navier-Stokes equation, which is a physically based equation. The purpose of this research is to restore the thin thin sheets and turbulence flow of fluid, which was difficult to express by visual simulation due to this problem. So we provide a framework to improve fluid visual simulation by numerically modeling fluid thin sheets and turbulent flow.

First, we used FLIP solver [42], which is particle-based fluids, to calculate underlying water simulation, and extracted thin particles anisotropically to find the region where water particles move to irregular (see Section 3.1.1). Based on the position of the water particle, we calculated covariance matrix and PCA (Principal Component Analysis), and found orientation and stretching of water particles. This process roughly finds the region where holes occur, and finally finds the water particles present in the hole region through the conditions proposed in Section 3.1.2. In order to add particles from a region with a relatively large hole, we break a hole by adding particles from a region with a low density of water particles (see Section 3.1.3). At this time, the total mass of fluid is increased by newly added water particles, the constraint of mass conservation is not maintained, which is physically incorrect. In order to solve this problem, we used a simple scheme to divide mass into halves each time water particles are split. Using this mass transfer method, as shown in the Figure 4a, the water particle disappears because the water particle can not correctly express the turbulence flow due to the mass value reduced by the splitting scheme, despite the presence of many water particles. In order to solve this problem, we use Equation (15) to calculate ghost density ρ^G . This value is the density calculated independent of the mass of the water particle, and density is calculated based on the volume of the particle, so you can restore the density value reduced by mass. In this process, ρ^G of particles may differ from position to position, and mass conservation cannot be guaranteed. Therefore, we use ρ^G to calculate ghost mass m^G for maintaining mass conservation via Equations (16)–(19)). This formula redistributes the mass based on the ratio of ρ^G calculated based on the particle volume and the sum of the ghost mass is normalized to be equal to the sum of the initial mass of fluid and the mass conservation condition is satisfied. As a result, ghost mass influences the calculation process of fluid's velocity, pressure, etc (see Equations (10) and (11)), so we can restore the turbulence flow lost in the splitting process.

7. Conclusions

In this study, we produce realistic turbulent flows around thin fluid sheets by redistributing the masses of particles. The proposed method considers the newly inserted particles to calculate the ghost density, which is then used to ensure the conservation of mass and to redistribute the mass of the particles. The calculated ghost mass affects the velocity step where it is mapped according to the density and the grid (see Equations (10) and (11)), thereby generating the turbulent flows and thus the motion of stormy water.

The proposed mass redistribution method is not based on physics, but it mathematically satisfies the conservation of mass to obtain turbulent flows. The proposed method handles turbulent flows by transferring the masses of water particles to particles near the fluid sheets, which causes the turbulent flows to decrease in some other regions. However, this issue is hardly visible because the mass is taken

from a region where the density of the particles is low. The proposed hybrid particle-grid solver could be applied to wispy smoke, viscoelastic materials, and liquid ligaments to improve subgrid vortical details; these applications will be considered in future research.

Author Contributions: J.-H.K., W.K., Y.B.K., and J.L. conceived of and designed the experiments. J.-H.K. and W.K. developed the main algorithm. J.H.K., Y.B.K. and J.L. analyzed the data. J.-H.K. and J.L. wrote the paper.

Funding: This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Science, ICT & Future Planning(No. 2017R1C1B5074984, 2018R1C1B5046461). This research was supported by a Hallym University Research Fund (HRF-201704-014).

Conflicts of Interest: The authors declare no conflict of interest.

References

- Chen, F.; Zhao, Y.; Yuan, Z. Langevin particle: A selfadaptive lagrangian primitive for flow simulation enhancement. *Comput. Gr. Forum* **2011**, *30*, 435–444. [\[CrossRef\]](#)
- Zhao, Y.; Yuan, Z.; Chen, F. Enhancing fluid animation with adaptive, controllable and intermittent turbulence. In Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation, Madrid, Spain, 2–4 July 2010; pp. 75–84.
- Selle, A.; Rasmussen, N.; Fedkiw, R. *A Vortex Particle Method for Smoke, Water and Explosions*; ACM SIGGRAPH: New York, NY, USA, 2005; pp. 910–914.
- Yoon, J.-C.; Kam, H.R.; Hong, J.-M.; Kang, S.J.; Kim, C.-H. Procedural synthesis using vortex particle method for fluid simulation. *Comput. Gr. Forum* **2009**, *28*, 1853–1859. [\[CrossRef\]](#)
- Kim, D.; Lee, S.W.; Young Song, O.; Ko, H.S. Baroclinic turbulence with varying density and temperature. *IEEE Trans. Vis. Comput. Gr.* **2012**, *18*, 1488–1495.
- Weissmann, S.; Pinkall, U. Filament-based smoke with vortex shedding and variational reconnection. *ACM Trans. Gr.* **2010**, *29*, 115–122. [\[CrossRef\]](#)
- Barnat, A.; Pollard, N.S. Smoke sheets for graph-structured vortex filaments. In Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation, Lausanne, Switzerland, 29–31 July 2012; pp. 77–86.
- Hong, J.M.; Shinar, T.; Fedkiw, R. *Wrinkled Flames and Cellular Patterns*; ACM SIGGRAPH: New York, NY, USA, 2007; Volume 26, pp. 47–54.
- Kim, D.; Song, O.Y.; Ko, H.S. *Stretching and Wiggling Liquids*; ACM SIGGRAPH Asia: New York, NY, USA, 2009; pp. 120:1–120:7.
- Mercier, O.; Beauchemin, C.; Thuerey, N.; Kim, T.; Nowrouzezahrai, D. Surface Turbulence for Particle-Based Liquid Simulations. *ACM Trans. Gr.* **2015**, *34*, 10–16. [\[CrossRef\]](#)
- Pfaff, T.; Thuerey, N.; Gross, M. Lagrangian vortex sheets for animating fluids. *ACM Trans. Gr.* **2012**, *31*, 112:1–112:8. [\[CrossRef\]](#)
- Pfaff, T.; Thuerey, N.; Cohen, J.; Tariq, S.; Gross, M. *Scalable Fluid Simulation Using Anisotropic Turbulence Particles*; ACM SIGGRAPH Asia: New York, NY, USA, 2010; pp. 174:1–174:8.
- Kim, T.; Thuerey, N.; James, D.; Gross, M. *Wavelet Turbulence for Fluid Simulation*; ACM SIGGRAPH: New York, NY, USA, 2008; pp. 50:1–50:6.
- Pfaff, T.; Thuerey, N.; Selle, A.; Gross, M. *Synthetic Turbulence Using Artificial Boundary Layers*; ACM SIGGRAPH Asia: New York, NY, USA, 2009; pp. 121:1–121:10.
- Dobashi, Y.; Matsuda, Y.; Yamamoto, T.; Nishita, T. A fast simulation method using overlapping grids for interactions between smoke and rigid objects. *Comput. Gr. Forum* **2008**, *27*, 477–486. [\[CrossRef\]](#)
- Klingner, B.M.; Feldman, B.E.; Chentanez, N.; O'Brien, J.F. *Fluid Animation With Dynamic Meshes*; ACM SIGGRAPH: New York, NY, USA, 2006; pp. 820–825.
- Losasso, F.; Gibou, F.; Fedkiw, R. *Simulating Water and Smoke With an Octree Data Structure*; ACM SIGGRAPH: New York, NY, USA, 2004; pp. 457–462.
- Jang, T.; Blanco i Ribera, R.; Bae, J.; Noh, J. Simulating sph fluid with multi-level vorticity. *Int. J. Virtual Real.* **2011**, *10*, 21–27.
- Yang, B.; Jin, X. Turbulence synthesis for shape-controllable smoke animation. *Comput. Anim. Virtual Worlds* **2014**, *25*, 465–472. [\[CrossRef\]](#)

20. Zhu, B.; Yang, X.; Fan, Y. Creating and preserving vortical details in sph fluid. *Comput. Gr. Forum* **2010**, *29*, 2207–2214. [[CrossRef](#)]
21. Ando, R.; Thürey, N.; Tsuruno, R. Preserving fluid sheets with adaptively sampled anisotropic particles. *IEEE Trans. Vis. Comput. Gr.* **2012**, *18*, 1202–1214. [[CrossRef](#)] [[PubMed](#)]
22. Ando, R.; Tsuruno, R. A particle-based method for preserving fluid sheets. In Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation, Vancouver, British, 5–7 August 2011; pp. 7–16.
23. Wojtan, C.; Thürey, N.; Gross, M.; Turk, G. *Physics-Inspired Topology Changes for Thin Fluid Features*; ACM SIGGRAPH: New York, NY, USA, 2010; pp. 50:1–50:8.
24. Stam, J. *Stable Fluids*; ACM SIGGRAPH: New York, NY, USA, 1999; pp. 121–128.
25. Fedkiw, R.; Stam, J.; Jensen, H.W. *Visual Simulation of Smoke*; ACM SIGGRAPH: New York, NY, USA, 2001; pp. 15–22.
26. He, S.; Wong, H.C.; Pang, W.M.; Wong, U.H. Real-time smoke simulation with improved turbulence by spatial adaptive vorticity connement. *Comput. Anim. Virtual Worlds* **2011**, *22*, 107–114. [[CrossRef](#)]
27. Park, S.I.; Kim, M.J. Vortex fluid for gaseous phenomena. In Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation, Los Angeles, CA, USA, 29–31 July 2005; pp. 261–270.
28. Angelidis, A.; Neyret, F. Simulation of smoke based on vortex lamente primitives. In Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation, Los Angeles, CA, USA, 29–31 July 2005; pp. 87–96.
29. Angelidis, A.; Neyret, F.; Singh, K.; Nowrouzezahrai, D. A controllable, fast and stable basis for vortex based smoke simulation. In Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation, Vienna, Austria, 2–4 September 2006; pp. 25–32.
30. Schechter, H.; Bridson, R. Evolving sub-grid turbulence for smoke animation. In Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation, Dublin, Ireland, 7–9 July 2008; pp. 1–7.
31. Vines, M.; Houston, B.; Lang, J.; Lee, W.S. Vortical inviscid flows with two-way solid-fluid coupling. *IEEE Trans. Vis. Comput. Gr.* **2014**, *20*, 303–315. [[CrossRef](#)] [[PubMed](#)]
32. Osher, S.; Sethian, J.A. Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations. *J. Comput. Phys.* **1988**, *79*, 12–49. [[CrossRef](#)]
33. Enright, D.; Fedkiw, R.; Ferziger, J.; Mitchell, I. A hybrid particle level set method for improved interface capturing. *J. Comput. Phys.* **2002**, *183*, 83–116. [[CrossRef](#)]
34. Wang, Z.; Yang, J.; Stern, F. An improved particle correction procedure for the particle level set method. *J. Comput. Phys.* **2009**, *228*, 5819–5837. [[CrossRef](#)]
35. Mihalef, V.; Metaxas, D.; Sussman, M. Textured liquids based on the marker level set. *Comput. Gr. Forum* **2007**, *26*, 457–466. [[CrossRef](#)]
36. Bargteil, A.W.; Goktekin, T.G.; O'Brien, J.F.; Strain, J.A. A semi-lagrangian contouring method for fluid simulation. *ACM Trans. Gr.* **2006**, *25*, 19–38. [[CrossRef](#)]
37. Heo, N.; Ko, H.S. *Detail-Preserving Fully-Eulerian Interface Tracking Framework*; ACM SIGGRAPH Asia: New York, NY, USA, 2010; pp. 176:1–176:8.
38. Müller, M. Fast and robust tracking of fluid surfaces. In Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation, New Orleans, Louisiana, 1–2 August 2009; pp. 237–245.
39. Wojtan, C.; Thürey, N.; Gross, M.; Turk, G. *Deforming Meshes That Split and Merge*; ACM SIGGRAPH: New York, NY, USA, 2009; pp. 76:1–76:10.
40. Blinn, J.F. A generalization of algebraic surface drawing. *ACM Trans. Gr.* **1982**, *1*, 235–256. [[CrossRef](#)]
41. Yu, J.; Turk, G. Reconstructing surfaces of particle-based fluids using anisotropic kernels. *ACM Trans. Gr.* **2013**, *32*, 5:1–5:12. [[CrossRef](#)]
42. Zhu, Y.; Bridson, R. *Animating Sand as a Fluid*; ACM SIGGRAPH: New York, NY, USA, 2005; pp. 965–972.
43. Adams, B.; Pauly, M.; Keiser, R.; Guibas, L.J. *Adaptively Sampled Particle Fluids*; ACM SIGGRAPH: New York, NY, USA, 2007; Volume 26, pp. 48–52.
44. Harlow, F.H.; Welch, J.E. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Phys. Fluids* **1965**, *8*, 2182–2189. [[CrossRef](#)]
45. Akinci, G.; Ihmsen, M.; Akinci, N.; Teschner, M. Parallel surface reconstruction for particle-based fluids. *Comput. Gr. Forum* **2012**, *31*, 1797–1809. [[CrossRef](#)]

46. Lorensen, W.E.; Cline, H.E. *Marching Cubes: A High Resolution 3d Surface Construction Algorithm*; ACM SIGGRAPH: New York, NY, USA, 1987; pp. 163–169.
47. Batty, C.; Bertails, F.; Bridson, R. *A Fast Variational Framework for Accurate Solid-Fluid Coupling*; ACM SIGGRAPH: New York, NY, USA, 2007; Volume 26, pp. 100–106.

© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).