Workshop on Virtual Reality Interaction and Physical Simulation VRIPHYS (2014)
J. Bender, C. Duriez, F. Jaillet, and G. Zachmann (Editors)

Controlling the Shape and Motion of Plumes in Explosion Simulations

Genichi Kawada[1,2] and Takashi Kanai[1]

[1]The University of Tokyo
[2]Digital Frontier Inc.

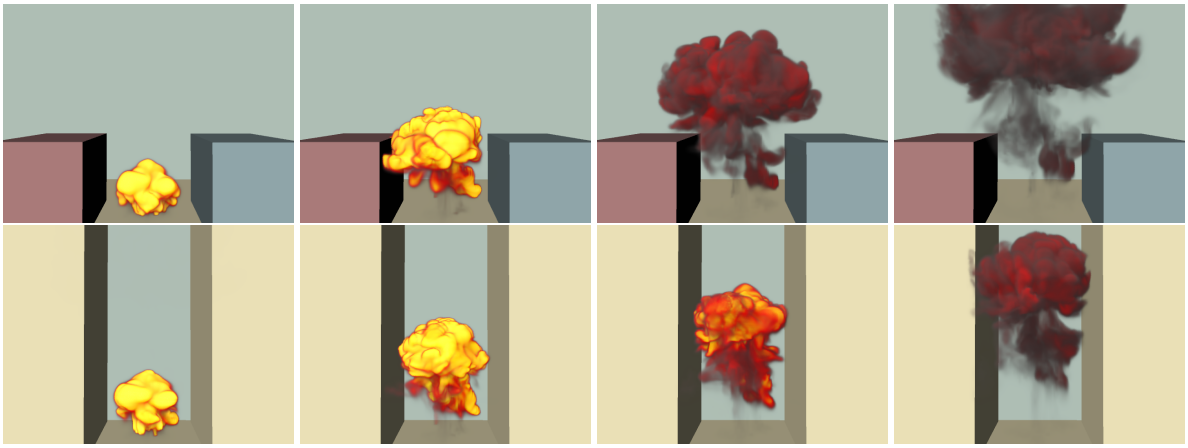

Figure 1: *Practical examples by our method (Top row, plume size is controlled while maintaining physical characteristic of size change.): from the left, 50th, 100th, 150th and 200th frame respectively. (Bottom row, control of swirling motions while maintaining physical characteristics): from the left, 60th, 100th, 120th and 150th frame respectively.*

Abstract

We propose a fluid simulation method with controlling the shape and motion of rising fire and smoke, called plumes*, in the incompressible phase of explosion phenomenon. With our method, plumes are generated based on physical phenomenon called* entrainment*, which strongly characterizes plume behaviors such as rise and circulation. We consider to newly utilize properties characterizing these behavior (*physical property*). Then, control elements of plume such as rising velocity, size, and the magnitude and position of swirling motions are individually adjusted using these physical properties. With this method, each control element is adjusted by the velocity field which represents the corresponding behavior. By combining all velocity fields and applying those fields to grid-based simulation, plumes can be generated. Our method is unique in that it can both generate and control plumes based on one unified physical model, and this type of model is firstly proposed here. Consequently, our method realizes plumes in the incompressible phase which maintain their physical characteristics as much as possible while being controlled by the user.*

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Physically based modeling—I.3.7Animation

1. Introduction

The phenomenon of explosion consists of a compressible phase where the fire generated by the high pressure ignition

DOI: 10.2312/vriphys.20141225

continues to burn and propagates at a high-speed velocity. It also has an incompressible phase where the fire rises due to its buoyancy after the propagation converges. The density of the fire during the incompressible phase becomes lower than that of the surrounding air. This is because the fire expands due to pressure during ignition. After that, the fire becomes incompressible when expansion stops. Here, since the density difference between fire and air generates the buoyancy force, the fire starts to rise. We call both this fire and smoke into *plume*.

Plume in explosion is a relatively important factor in films, etc. While various methods [FSJ01, Miz03, KLSK11, PTG12] have been previously proposed as physical models for plumes, none of these can be currently considered to control the plumes while physical behaviors is maintained.

In contrast to these methods, we propose a physical model based on *entrainment*. Entrainment is the physical phenomenon in which the surrounding air is absorbed by the whole surface of the plume when the plume rises [Pat07]. Specifically, plume absorbs the air which has higher density than the plume, and this causes plume density and temperature to change. For this reason, this entrainment strongly determines the physical behaviors of plume. Characteristically, our method aims to generate plume shape and motion which have a larger scale than details such as turbulences because it is based on behaviors caused by entrainment.

Various researches have also been conducted on the control of fluid simulation [TMPS03, WP10, FL04, SY05, PCS04, ZCM11]. However, most of these methods treat fluid fields independently from the physical behaviors. Our research differs from such researches as our aim is to develop a control method which directly utilizes the entrainment phenomenon. Specifically, our method is able to control the rising velocity and size of plumes, and the magnitude and position of the swirling motions in plumes by defining buoyant rise, expansion, and circulation behaviors which are generated by entrainment.

Consequently, our method is able to handle both the generation and control of plumes based on one unified physical model. Our method has the following contributions:

- The mathematical formulas and physical characteristics representing the various behaviors generated by entrainment are introduced so as to be specialized in the area for computer graphics simulation. By doing this, plumes can be generated using a model expressing the various physical properties of plumes.
- We propose a new method for separately describing the behaviors of plumes such as rise, expansion, and circulation which are generated by entrainment. This enables individual adjustment of control elements corresponding to each behavior such as the rising velocity, size, and the magnitude and position of the swirling motions in plumes. As shown in Figure 1 top row, plume size is controlled so that the plume rises out of a small space. Then, the plume is controlled so as to increase the size without colliding into the walls while maintaining the physical characteristic over the plume size. In addition, as the bottom row indicates, rising plume with swirling motions is generated in the narrow space without colliding into walls.
- We propose a method which combines fluid simulation and the procedural generation of velocity fields to express the buoyant rise, expansion, and circulation. This allows our method to acquire the advantages of the fluid simulation such as ability to handle boundary condition and obstacle interactions, while controlling the shape and motion of plume.

2. Related Work

2.1. Physically-based Simulation Method

Fedkiw et al. [FSJ01] expressed rising plumes by computing the buoyancy force from the temperature difference between plume and air. However, this method does not take into account the decrease in temperature by entrainment when determining the buoyancy. In addition, Mizuno et al. [Miz03] generated the volcanic clouds effectively by using physical models to calculate the buoyancy force produced by entrainment. Here, the air density changing with respect to height is also computed to determine the force. With these methods, direct control of the size and swirling motions of the plumes or clouds are not considered while the rising velocity can be changed by using the physical parameters of the buoyancy force.

On the other hand, Feldman et al. [FOA03] described the expansion by artificially adding the divergence modification to the pressure term. Kwatra et al. [KGF10] also expressed explosions in the compressible phase using physical models derived from equations for the law of energy conservation, and took into account the transition to the incompressible phase. However, in fact, all the methods described above do not propose physical models for expressing behaviors in incompressible plumes such as an incompressible flow expansion (expansion), or circulation, except for buoyancy force. Our method aims to express more various physical behaviors of plumes based on physical model taking entrainment into account.

2.2. Control Method for Fluid Simulations

Treuille et al. [TMPS03] computed the velocity fields to generate a state where the smoke density moves at a certain frame as intended by the optimization method. Therefore, smoke is controlled to form a specific shape. Weismann and Pinkall [WP10] interactively controlled smoke behavior with a model for using discrete rings expressing vortex called filaments. There are many other methods such as [FL04, SY05, MTPS04] which effectively control smoke and liquid using velocity fields to express external force to obtain the intended shape and motion. In addition, Kim

et al. [KLSK11] proposed a method which adds details to plumes by considering the baroclinic phenomenon where turbulences are generated because of the inclines of the density or temperature. However, [KLSK11] is intended for relatively small swirling motions of turbulences, and all of these methods mentioned above are not intended for generating vortex which is based on physical characteristics by entrainment.

Pighin et al. [PCS04] realized the control for the plumes by realizing easy modifications of moving density. While this control method gives the users interactive controls, it does not consider the physical behaviors of the plumes. Zhang et al. [ZCM11] efficiently realized the control of fires except explosions using pre-computed databases to express fire motions. Kawada and Kanai [KK11] procedurally generated and controlled explosion's fires by following the paths, based on explosion curves for representing the properties of explosion propagations. However, this method does not take the rising plumes in the incompressible phase into account.

Therefore, since all the methods described above are not intended for realizing the control of shapes and motions of incompressible plumes based on physical characteristics, it becomes difficult to maintain the physical realism of plumes. In contrast, our method aims to control plumes by utilizing physical model based on entrainment while maintaining physical realism as much as possible.

3. Controlling Plumes Based on Entrainment

Our method assumes that fuel during ignition is no longer left in the plume. Therefore, the additional combustion of fuel is not considered. If fuel remains, our method does not consider the effects of the vortex generated by the combustion of this fuel. However, by applying a method such as [KJI07] to express the combustion by vortex particles, this vortex effects can be added. According to the assumption above, since the entrained air is not used for combustion, our method takes that entrainment strongly determines plume behaviors. Our method does not consider the case where plural plumes are generated. This is because the plural blended plumes have complex behaviors, and it becomes difficult to realize control while physical behavior is maintained.

According to the assumption above, we consider that energy is not added, and then the total heat capacity is constant. The whole temperature of the plume thus becomes low, because the plume entrains the air and its temperature is lower than that of the plume. Consequently, since the plume state changes (details will be explained later), the following plume behaviors are generated. The rising behavior of plume (we call this *buoyant rise*) indicates that the buoyancy-based velocity (we call this velocity *rising velocity*) changes. Also, the plume becomes large (we call this *expansion* for convenience), and the flow of the plume generates the *circulation*. All of those behaviors are generated by entrainment.

Here, we consider a new model to express these physical behaviors. In physically-based simulations, it is thought that these behaviors can be obtained by solving the equations for the law of energy conservation and ideal gas law, in addition to incompressible Navier-Stokes equations. However, it is difficult to control each behavior independently using only these equations.

In order to maintain realism by taking physical characteristics into consideration and to individually control the plume behavior at the same time, our method takes the following into account. First, behaviors such as buoyant rise, expansion, and circulation are independent. Here, the property which characterizes each behavior (we call this *physical property*) is used. Without solving the equations described above, our method directly expresses each plume behavior based on the equations described in the study of the plume motion and shape [KS11], or plume's physical characteristics.

Our method uses the velocity fields to express each behavior, based on physical properties corresponding to each behavior. Each behavior is then controlled using each of these velocity fields. Originally, the mutual relationships among these properties need to be considered. For instance, when the rising velocity caused by buoyant rise becomes large, the change in plume volume caused by expansion becomes small. Consequently, by considering each behavior independently, nonlinear complex behaviors generated by mutual relationships can be no longer realized. On the other hand, with our method, each plume behavior is controlled by velocity fields so as to maintain its physical property. We combine these velocity fields and eventually solve grid-based simulation including the pressure term. By doing this, plumes maintaining physical realism as much as possible is generated, while being controlled. Details of physical properties for expressing each behavior are described in following subsections.

3.1. Physical Property of Buoyant Rise

Our method expresses the plume's rising velocity and its change using the physical property of buoyant rise. The plume rises and entrains the air. As a result, the mass fraction of the entrained air in the plume (which we call simply *fraction*) becomes high. As the fraction changes, the behavior of buoyant rise gradually changes.

Our method considers the physical property of buoyant rise based on [KS11] which studied the change of plume motions and shapes per time. The relationship between plume density ρ_{plum} and fraction ξ is described as follows [KS11]:

$$\rho_{plum} = \frac{T_a \rho_a}{(1-\xi)T_0 + \xi T_a}. \quad (1)$$

Equation (1) is based on the ideal gas law. ρ_a is the density of the surrounding air (=1.0), and T_a is the air temperature. Also, T_0 is the temperature of the plume when it just starts to

rise. The buoyancy force on the plume is determined by the density difference between the plume and surrounding air. As the plume density becomes high, the density difference gradually becomes small and the buoyancy force decreases. Therefore, the rising velocity of the plume also decreases.

3.2. Physical Property of Expansion

Our method expresses plume size according to the physical property of expansion. Similar to the property of buoyant rise, we consider the physical property of expansion based on the [KS11]. As time progresses and the plume entrains the air, the plume expands and its size gradually becomes large. The relationship between plume volume ν_{plum} and fraction ξ is described as follows [KS11]:

$$\nu_{plum} = (\frac{T_a}{T_0}\frac{\xi}{1-\xi} + 1)\nu_{source}. \quad (2)$$

Equation (2) is based on the ideal gas law, and ν_{source} is the plume volume at the moment the plume starts to rise (we call this plume source).

3.3. Physical Property of Circulation

While the plume entrains the air over its whole surface, entrainment is caused especially around the lower part of the rising plume (the upper part is defined based on vertically upward direction). At the lower part, air is dragged into the center of plume. This is because the Rayleigh-Taylor instability state [Sha84] is generated by the temperature difference between the plume and air, resulting in a phenomenon where the plume mixes with air. Therefore, the velocity field to drag the air (we call this *absorption velocity field*) occurs at the lower part of the plume. In addition, the Rayleigh-Taylor instability causes the velocity field to radially expand from the center side to the outside of the plume (we call this *divergence velocity field*) at the upper part of the plume. Based on these two velocity fields and the rising velocity at the center side, we consider to express the swirling flows in the plume (we call this *circulation vortex*).

4. Algorithm

Figure 2 shows our whole algorithm. It consists of the Initial Setting, followed by Process 1 to Process 3 which are executed repeatedly per step. This algorithm generates plume controlled in terms of rising velocity, size, and magnitude and position of the circulation vortex.

- In the Initial Setting, the user determines the executing time for each process (buoyant rise, expansion and circulation), and also specifies the initial temperature of the plume. In addition, the density and velocity fields for the plume source is specified (Section 4.1).
- In Process 1, the plume of the current step is detected (Section 4.2.1).

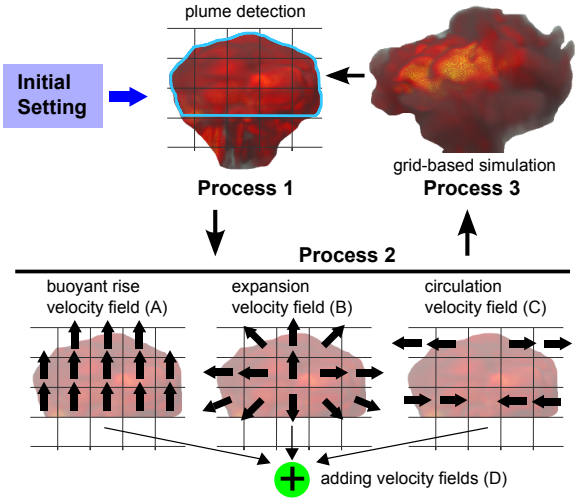

Figure 2: *Our whole algorithm.*

- In Process 2, the velocity fields to represent buoyant rise, expansion, and circulation property terms during the current step are individually computed (Section 4.2.2-4.2.4). Then, these velocity fields are added to the plume in the grid-based simulation domain (Section 4.2.5).
- In Process 3, the whole simulation domain is calculated by executing grid-based simulation (Section 4.2.6). Here, the simulation domain is thought to cover the whole plume until control finishes.

The following subsections describe each process of our algorithm.

4.1. User Input and Fraction Calculation

As the Initial Setting, the input to represent the time to execute the control process for each physical property is specified. In addition, the input to represent the plume's state when the plume starts to rise including the initial temperature is also specified. Based on those inputs, the fraction change per step is calculated.

The time to execute the process for each property, in other words, the duration time is specified respectively as input. The user also specifies the time when the plume starts to rise. The maximum duration time is discretized by the fixed time interval Δt, and the duration time is described as N steps, taking the time to rise to be step 0.

When the step is 0, the plume temperature is the initial temperature T_0, and after the duration time is passed, the plume temperature becomes the air temperature T_a. T_0 and T_a are also specified by the user. The higher T_0 is, the larger is the change in the plume's rising velocity by the buoyant force, and the smaller the size change by the expansion.

Since our method deals with each behavior independently,

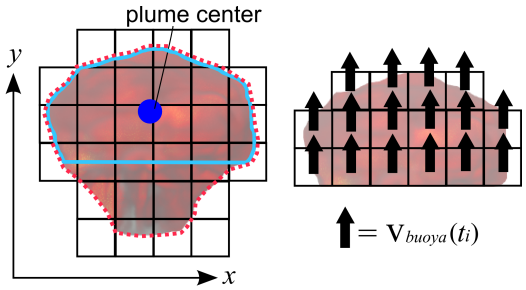

Figure 3: *Left: Plume detection. Right: Velocity field to represent buoyant rise.*

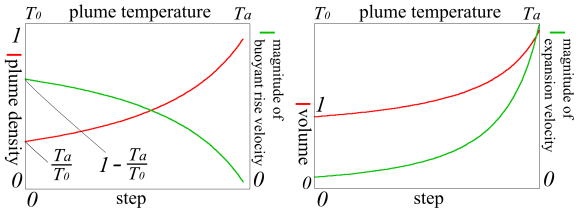

Figure 4: *Curves to represent plume's physical properties. Left: Density and magnitude of buoyant rise velocity. Right: Volume and magnitude of expansion velocity.*

first the case where only buoyant rise behavior exists and that where only expansion behavior exists are reviewed individually. For this reason, the plume temperature and fraction per step are obtained for each behavior respectively. The plume temperature decreases as the plume entrains the air. Since our method assumes that fuel does not burn once the plume starts to rise, thermal energy is not to be added. Consequently, we think that the plume temperature decreases from T_0 to T_a. For simplicity, we especially consider the linear decrease per step. The plume temperature per step T_{plum} for each buoyant rise and expansion is automatically calculated by taking that T_{plum} becomes T_a when the duration time for each behavior is passed. The relationship between plume temperature T_{plum} and fraction ξ is described as follows [KS11]:

$$T_{plum} = \frac{(1-\xi)C_0T_0 + \xi C_aT_a}{(1-\xi)C_0 + \xi C_a}. \quad (3)$$

Here, C_0 and C_a are the ratio of the specific heats of the plume which had just started to rise and air. This relationship indicates the characteristic where fraction increases with time, so that plume temperature deceases gradually and eventually becomes almost the same as air temperature. The fraction per step for each buoyant rise and expansion is calculated backwards by using the value per step of T_{plum} determined above and based on Equation (3). By considering $C_0 = C_a$ as noted in [KS11], the fraction is actually calculated as $\xi = \frac{T_0 - T_{plum}}{T_0 - T_a}$ and $0 \leq \xi \leq 1$. By taking that the plume temperature linearly decreases to T_a, as ξ in Equation (2) becomes close to 1, the value of v_{plum} becomes close to maximum. However, our method does not consider the state when the plume becomes so large that the plume is almost assimilated into the air. Therefore, the expansion process is finished before ξ becomes 1.

On the other hand, only the duration time is used for the circulation property, because our model does not express the circulation by temperature change (Section 4.2.4 for details).

Using the previous method [FOA03] to generate compressible explosion, density and velocity fields for expersssing the plume source are specified by the user.

4.2. Process per Step

In this subsection, the process at step $i (i = 0 \ldots N)$, in other words, when time $t = t_i$, is described. Three processes are executed in order. The plume source starts to rise when $t = 0$ (at step 0), and continues to rise per step. Each process to compute the velocity field for each physical property is executed until the step corresponding to each duration time is reached.

4.2.1. Plume Detection (Process 1)

The plume region is detected during step i. This enables the use of information on the plume position and shape for calculating the velocity fields for expressing each behavior. The area surrounded by blue line in Figure 3 Left indicates the detected plume.

In the simulation domain, we calculate the average velocity magnitude in y direction (vertically upward direction) at all cells (the corresponding cells in the area surrounded by dotted red line in Figure 3) where the density values are larger than a certain threshold. We typically use quite small number such as $0.01\rho_{max}$ (ρ_{max} is the maximum value of the density) for this threshold. Our method assumes that energy is not added and the velocity field itself in the rising flow of the prospective plume advects. Therefore, most of the velocity fields which have the large y component magnitude are concentrated in the upper part compared to the lower part. Consequently, we consider the cells corresponding to dotted red line used to calculate the average mentioned above. The cells having the y component magnitude, which is the multiplication of the constant number by the average are detected as the plume region (the corresponding cells in the area surrounded by blue line in Figure 3). The smaller this constant number, the more will the lower part in the flows be detected as a plume. Finally, the bounding box covering the detected plume region is computed, and the center of this bounding box is obtained as the center of the plume. By setting this constant number as 0, the plume where its density is in the vertically downward area can be detected.

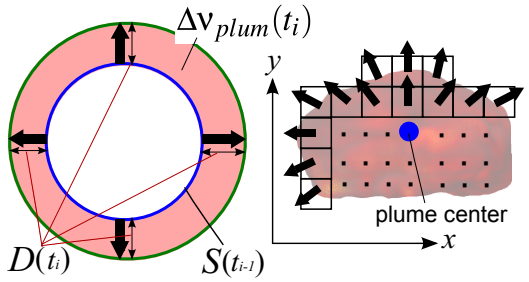

Figure 5: *Left: Expansion on spherical approximation. Right: Velocity field to represent expansion (arrows are partially described for simplification).*

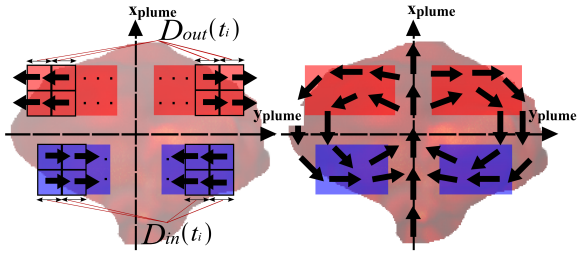

Figure 6: *Divergence and absortion velocity fields. Left: x,y plane (view as the cutting plane of* x_{plume} *and* y_{plume} *in Right). Right: x,z plane (view from positive to negative in axis* y_{plume}*).*

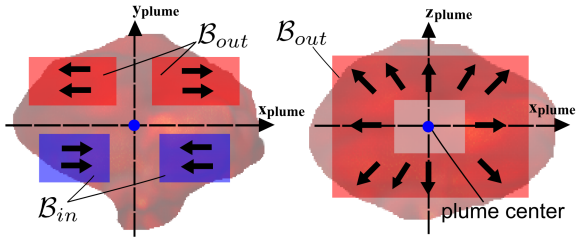

Figure 7: *Left: Velocity field to represent circulation. Right: Circulation vortex.*

4.2.2. Velocity Field Computation for Buoyant Rise (Process 2 (A))

The velocity field during the step i on the physical property of buoyant rise is calculated. Here, the plume's density at step i, $\rho_{plum}(t_i)$ is calculated in Equation (1) by regarding ξ as the fraction for buoyant rise per step $\xi_{buoya}(t_i)$, which is obtained in Section 4.1. $\rho_{plum}(t_i)$ is indicated as the red curve in Figure 4 Left actually obtained by our method. According to [MDN04], the velocity generated by the buoyancy force on the plume, in other words, buoyant rise velocity vector $\mathbf{v}_{buoya}(t_i)$ is computed by the equation as follows:

$$\mathbf{v}_{buoya}(t_i) = \frac{\alpha}{\Delta t}\frac{\rho_a - \rho_{plum}(t_i)}{\rho_a}\mathbf{y}, \tag{4}$$

where α is the coefficient for adjusting the moving distance. This coefficient is used for scaling the velocity magnitude in such a way that the characteristic of the velocity change is maintained. Also, $\mathbf{y}$ is a vertically upward unit vector $(0,1,0)$. The magnitude of buoyant rise velocity $|\mathbf{v}_{buoya}(t_i)|$ is indicated as the green curve in Figure 4 Left actually obtained based on Equation (4). We calcualte the velocity vector $\mathbf{v}_{buoya}(t_i)$ in all cells held by the detected plume region, and these vectors are obtained as the velocity field $\mathcal{U}_{buoya}$ in Figure 3 Right.

4.2.3. Velocity Field Computation for Expansion (Process 2 (B))

The velocity field by the physical property of expansion is calculated during the step i. Here, the plume's volume at step i, $v_{plum}(t_i)$ is calculated in Equation (2) by regarding ξ as the fraction for expansion per step $\xi_{expan}(t_i)$ which is obtained in Section 4.1. $v_{plum}(t_i)$is indicated as the red curve in Figure 4 Right actually obtained by our method. $v_{plum}(t_i)$ per step is indicated as the red curve in Figure 4 Right actually obtained by our method. Here, our method realizes the expansion effects by using the velocity field to express the volume change and moving the density. First of all, in order to obtain the magnitude of this velocity field during the step i, we consider that the plume is approximated as a sphere for simplicity as a blue circle in Figure 5 Left. The increased part of the plume during the step i (thin red region, $\Delta v_{plum}(t_i)$) is calculated as $v_{plum}(t_i) - v_{plum}(t_{i-1})$. $D(t_i)$ is the width in the normal direction from the plume's surface at step $i-1$ (blue circle) to that at step i (green circle). This width is represented as the moving distance of density by the expansion.

By using the relationship between a sphere's volume and surface area, the plume's surface area at step $i-1$, $S(t_{i-1})$ is calculated from $v_{plum}(t_{i-1})$. Next, for simplicity we take that $\Delta v_{plum}(t_i) \approx S(t_{i-1})D(t_i)$, then $D(t_i)$ is computed. Eventually, the magnitude of the velocity field to express the expansion is obtained as $\beta\frac{D(t_i)}{\Delta t}$. Here, β is set as the coefficient introduced to scale the magnitude of the expansion velocity. Our method directly specifies the magnitude of the expansion velocity field by β. Our method differs from methods such as [FOA03] which indirectly specifies the magnitude of velocity through grid-based simulation.

In all the cells held by the detected plume region, we calculate the velocity vector which has the uniform magnitude $\beta\frac{D(t_i)}{\Delta t}$ and the direction from the plume's center to each cell. $\frac{D(t_i)}{\Delta t}$ is indicated as the green curve in Figure 4 Right obtained in the above method. Those vectors are obtained as the expansion velocity field $\mathcal{U}_{expan}$ (arrows in Figure 5 Right).

4.2.4. Velocity Field Computation for Circulation (Process 2 (C))

The velocity field of the physical property of circulation at step i is calculated based on the region information of the detected plume. Figure 6 and Figure 7 indicate the schematic views of the absorption and divergence velocity fields. Also, $\mathbf{x_{plume}}$, $\mathbf{y_{plume}}$, $\mathbf{z_{plume}}$ indicate axes in the local coordinate system having their origin at the plume center here.

To represent the divergence velocity field, we take into account the hollow region $\mathcal{B}_{out}$ (red region in Figure 6) where a region around the plume's center is omitted. At each cell inside $\mathcal{B}_{out}$, we consider the velocity vector (arrows of red regions in Figure 6 Left and Figure 7 Left) which has the opposite direction perpendicular to $\mathbf{y_{plume}}$ and the magnitude $\gamma_{out}\frac{D_{out}(t_i)}{\Delta t}$, where $D_{out}(t_i)$ denotes the scalar value to express the moving distance at step i. This value is the length of edge of each cell (cubic lattice) in $\mathcal{B}_{out}$. In addition, γ_{out} is the adjustment coefficient for the moving distance. Those vectors are the divergence velocity field $\mathcal{U}_{outCirc}$ (red parts of Left or Right in Figure 6).

Next, to represent the absorption velocity field similar to $\mathcal{B}_{out}$, we consider $\mathcal{B}_{in}$ (blue region in Figure 6 Left and Figure 7 Left). At each cell inside $\mathcal{B}_{in}$, we consider the velocity vector (arrows of blue regions in Figure 6 Left and Figure 7 Left) which has the perpendicular line's direction to $\mathbf{y_{plume}}$ and the magnitude $\gamma_{in}\frac{D_{in}(t_i)}{\Delta t}$. $D_{in}(t_i)$ is obtained as the scalar value to express the moving distance similarly to $D_{out}(t_i)$, and γ_{in} is the adjustment coefficient for the moving distance. Those vectors are the absorption velocity field $\mathcal{U}_{inCirc}$ (blue parts in Figure 6 Left). Eventually, the velocity field for circulation is calculated as $\mathcal{U}_{circ} = \mathcal{U}_{InCirc} \cup \mathcal{U}_{OutCirc}$. Our method defines $\cup$ as the sum operation to combine the cells in velocity fields. $\mathcal{B}_{out}$ and $\mathcal{B}_{in}$ are automatically calculated to fit to the bounding box obtained for the plume in Section 4.2.1. The user can specify fine adjustment of the parameters for size and position of $\mathcal{B}_{out}$ and $\mathcal{B}_{in}$ in order to control the region where the circulation is applied.

The velocity calculations described above are based on the methods in [SP00] (or [Sco57]) to express circulation vortex. Consequently, the velocity field indicated in Figure 7 Left generates a circulation vortex shown as the arrows in Figure 7 Right.

4.2.5. Adding and Allocating each Velocity Field (Process 2 (D))

Eventually, the velocity fields for expressing each behavior are added as follows:

$$\mathcal{U}_{entrain} = \mathcal{U}_{buoya} \cup \mathcal{U}_{expan} \cup \mathcal{U}_{circ}, \tag{5}$$

and the sum of such velocity fields $\mathcal{U}_{entrain}$ is obtained. This $\mathcal{U}_{entrain}$ is actually allocated to the corresponding cells in the grid-based simulation domain.

4.2.6. Grid-Based Simulation for Fields (Process 3)

In the final process, by applying grid-based simulation for one step, the density, velocity, and pressure fields are updated for the entire domain. The domain is calculated by executing grid-based simulation [Sta99] using the BFECC (Back and Forth Error Compensation and Correction) method [KLLR05] as the advection term for one step to satisfy the incompressibility condition. As a result, plume behavior is controlled while treating the interaction with obstacles by solving the grid-based simulation.

5. Results and Discussion

The experimental results obtained by using our method are shown in this section. The results are generated using a PC with a 2.8-GHz Intel Core i7-930. Computational cost with $150 \times 200 \times 150$ simulation cells in x, y, z is about 30 seconds per frame (within 1 second for Process 1 and 2) in all experiments. The computation is spent mostly on the grid-based simulation of Process 3. On the other hand, the computational cost for the processes proposed by our method is relatively low. This is because the total computational cost of Process 1 and 2 is at most $O(mN)$ and $m \ll N$, where m is the number of times of going through the entire grid and N is the total number of the cells.

5.1. Comparison between each property

Figure 9 shows the comparisons of plume behaviors for each physical property and the behavior based on the method in [FSJ01]. In addition, the density field with a spherical shape is used for the plume source. The color field is used for rendering. The transition from fire to smoke is expressed by changing the color information as time progresses.

(a) and (f) in Figure 9 show the results using only the buoyancy force with the duration time, 200 frames. (f) shows that the plume continues to rise until the 200th frame. (b) and (g) show the results of applying expansion up to the 200th frame in experiments (a) and (f) respectively. For the plume size, (b) is larger than (a), and (g) is larger than (f).

Next, (c) and (h) demonstrate the results of applying circulation to experiments (a) and (f) respectively. This example shows that the circulation vortex is added to the plume. Here, the magnitude of the absorption and divergence velocity fields are set the same. Our method sets the velocity magnitude of $\mathcal{U}_{circ}$ in Process 2 (C) as 0 every two frames. This prevents the circulation vortex generated by updating the velocity field in the grid-based simulation from becoming quite strong. The following examples are set in the same way.

Adding to that, (d) and (i) show the results including all three behaviors. Here, the magnitude of the absorption and divergence velocity fields are set the same. This example shows that the circulation vortex is added to the plume. Due

to expansion effects, the plume is larger than that in (a) and (f) respectively. (e) and (j) demonstrate the results of directly calculating the buoyancy force according to plume temperature based on the method in [FSJ01]. This example only shows the buoyant force by the temperature change without considering entrainment and the vortex whose magnitude is proportional to the buoyant rise velocity field. This vortex is obtained by updating the velocity which expresses buoyant force in the grid-based simulation.

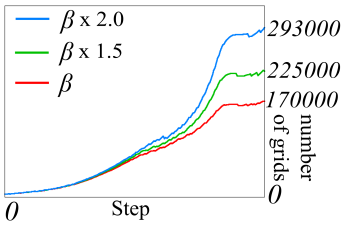

Figure 8: *Relation between β and size (number of cells in plume region).*

5.2. Comparison by Adjusting Each Property's Parameters

Figure 10 demonstrates the results of changing the adjustment coefficient α of the buoyant rise property with the duration time of 100 frames. (a), (b) and (c) show examples at the 100th frame using α (set as a standard for comparion), $\alpha \times 0.5$ and $\alpha \times 2.0$ respectively. These examples show that the rising velocity can be adjusted by using α.

Figure 11 shows the results of changing the adjustment coefficient β of the expansion property with the duration time of 200 frames. (a), (b) and (c) indicate the examples of using β (set as a standard for comparison), $\beta \times 2.0$ and $\beta \times 4.0$ respectively. Moreover (d) and (e) show the displayed results with the expansion duration time of 120 frames for the 120th and 150th frames respectively. (e) shows that the plume no longer becomes large at the 150th frame and its size is similar to that at the 120th frame, because the expansion process has ended. Here, the expansion process ends right before the expansion duration time is exceeded since our method does not consider the case when the plume becomes so large to be assimilated into the air (Section 4.1). The other examples are set in the same way. These examples show that plume size can be adjusted using β.

Next, Figure 12 demonstrates the results of changing the control setting on the circulation from (c) and (h) experiment in Figure 9. The values of γ_{out} and γ_{in} linearly decrease to 0 from the middle in (a) in Figure 12. This example shows that the plume width is smaller especially at the upper part compared to (b) in Figure 12, because the divergence velocity field is lost. In addition, (c) and (d) in Figure 12 demonstrate the result where γ_{out} is twice as large as γ_{in}. This example shows that the plume is quite wide, because the divergence velocity field is stronger than the absorption velocity field. The examples above show that by adjusting γ_{out} and γ_{in}, our method can disable circulation vortex effects from the middle, or the plume width can be controlled. As described in Section 5.1, the method expressing vortex by buoyancy requires rising velocity with large magnitudes in order to obtain strong vortex. In contrast, our method can control the rising velocity and magnitude of circulation vortex independently.

5.3. Other Practical Examples

At the top row in Figure 1, buoyant rise and expansion are used and adjusted in order to control plume behaviors generated between the walls. This example shows that with our control method, the plume rises in the space between the walls for a while after ignition, and later the plume becomes larger than the space without interacting with the walls. In particular, as shown in Figure 4 Right, plume is controlled so that the change in plume size increases with time. The bottom row in Figure 1 demonstrates that buoyant rise and circulation are used and adjusted to control plume behaviors generated in the narrow space between the high walls. This demonstrates that plumes are controlled so as to maintain circulation vortex without changing their size so that the plume dose not collide into the walls in long narrow space. In contrast, for instance, with previous methods such as [FSJ01], the plume could collide into the walls because the plume size is forced to become large when the rising speed increases to maintain vortex.

With our method, first the plume behaviours are determined by adjusting the buoyant rise and expansion respectively, and each adjustment requires several trials and errors. Especially for expansion, by referring to the relationship between β and the plume size as Figure 8 indicates, the intended results can be obtained with a relatively small number of adjustments. After this, circulation behaviour is added to the plume. In this way, the number of trials and errors can be reduced. In the comparisons between (a) and (c), or (b) and (d) in Figure 9, when the magnitudes of the absorption and divergence velocity fields are close, the influence of circulation on plume size and rise velocity is especially reduced.

As the red circles of (i) in Figure 9 and (d) in Figure 12 indicate, the vertically downward area can be partially separated in the case when the strong circulation vortex is generated depending on $D_{in}(t_i)$, or $D_{out}(t_i)$ in the circulation property. However, by setting the constant number for $D_{in}(t_i)$, or $D_{out}(t_i)$ value as the small number which could also detect the density with the small magnitude of vertical vector, the plume's behavior where there is no separation is obtained as Figure 1 indicates.

Plume behaviours can be controlled and obtained almost

with only the computational costs for grid-based simulation, and without huge additional costs for solving the equations for the law of the energy conservation and ideal gas law. This is another advantage of our method.

However, in the future, it is important to consider the method to apply two dimensional simulation results with lower computational cost as close as possible to three dimensional behaviours. Also, we deal with only one plume source in one grid-simulation domain, and multiple domains are used in order to deal with multiple plumes. However, the method to detect multiple plumes and deal with them has to be developed in future work. It is possible to add the new external forces to express the wind and so on to the sum of the velocity field in Process 2 (D) of our method. However, it could be that due to the larger magnitude of the external forces, such as the wind, the plume's physical behavior becomes less dominant.

6. Conclusions and Future Work

In this paper, we proposed a method to control the shapes and motions of plumes, while maintaining the physical realism of plumes as much as possible by utilizing physical properties characterizing the behaviors generated by entrainment such as buoyant rise, expansion, and circulation. Our method detects plume and combines all velocity fields computed independently to express each plume behavior. By doing this, the rising velocity, size, and the magnitude and position of circulation vortex on the plumes can be controlled.

In future work, we hope to apply the proposed method to generate a mushroom cloud as a type of plume. To realize this, for instance, the characteristics of physical behaviors such as the temperature change of a mushroom cloud must be considered. Also the model to deal with non-spherical expansion can be considered to realize the ellipsoid body part of the mushroom cloud. In addition, we would like to develope a method to generate plume behaviors which take more dynamic changes of the temperature into account by considering the additional combustion of the fuel. Taking into account this type of temperature could help improve the realism of rendering qualities.

References

[FL04] FATTAL R., LISCHINSKI D.: Target-driven smoke animation. *ACM Transactions on Graphics 23*, 3 (2004), 441–448. 2

[FOA03] FELDMAN B. E., O'BRIEN J. F., ARIKAN O.: Animating suspended particle explosions. *ACM Transactions on Graphics 22*, 3 (2003), 708–715. 2, 5, 6

[FSJ01] FEDKIW R., STAM J., JENSEN H. W.: Visual simulation of smoke. In *Proc. ACM SIGGRAPH '01* (2001), pp. 15–22. 2, 7, 8, 10

[KGF10] KWATRA N., GRÉTARSSON J. T., FEDKIW R.: Practical animation of compressible flow for shock waves and related phenomena. In *Proc. Symposium on Computer Animation* (2010), pp. 207–215. 2

[KJI07] KANG B., JANG Y., IHM I.: Animation of chemically reactive fluids using a hybrid simulation method. In *Proc. Symposium on Computer Animation* (2007), pp. 199–208. 3

[KK11] KAWADA G., KANAI T.: Procedural fluid modeling of explosion phenomena based on physical properties. In *Proc. Symposium on Computer Animation* (2011), pp. 167–175. 3

[KLLR05] KIM B., LIU Y., LLAMAS I., ROSSIGNAC J.: Flowfixer: Using BFECC for fluid simulation. In *Eurographics Workshop on Natural Phenomena* (2005), pp. 51–56. 7

[KLSK11] KIM D., LEE S. W., SONG O.-Y., KO H.-S.: Baroclinic turbulence with varying density and temperature. *IEEE Transactions of Visualization and Computer Graphics 18*, 9 (2011), 1488–1495. 2, 3

[KS11] KITAMURA S., SUMITA I.: Experiments on a turbulent plume: Shape analyses. *Journal of Geophysical Research B03208* (2011), 1488–1495. 3, 4, 5

[MDN04] MIZUNO R., DOBASHI Y., NISHITA T.: Modeling of volcanic clouds using CML. *Journal of Information Science and Engineering 20*, 2 (2004), 219–232. 6

[Miz03] MIZUNO R.: *Modeling of Volcanic Clouds for Computer Graphics*. Master's thesis, The University of Tokyo, 2003. 2

[MTPS04] MCNAMARA A., TREUILLE A., POPOVIĆ Z., STAM J.: Fluid control using the adjoint method. *ACM Transactions on Graphics 23*, 3 (2004), 449–456. 2

[Pat07] PATRICK M.: Dynamics of strombolian ash plumes from thermal video: Motion, morphology, and air entrainment. *Journal of Geophysical Research 112*, B06202 (2007). 2

[PCS04] PIGHIN F., COHEN J. M., SHAH M.: Modeling and editing flows using advected radial basis functions. In *Proc. Symposium on Computer Animation* (2004), pp. 223–232. 2, 3

[PTG12] PFAFF T., THUEREY N., GROSS M.: Lagrangian vortex sheets for animating fluids. *ACM Transactions on Graphics* (2012). 2

[Sco57] SCORER R.: Experiments on convection of isolated masses of buoyant fluid. *Journal of Fluid Mechanics 2* (1957), 583–594. 7

[Sha84] SHARP D.: An overview of rayleigh-taylor instability. *Physica D: Nonlinear Phenomena 12*, 1–3 (1984). 4

[SP00] SREENIVAS K., PRASAD A.: Vortex-dynamics model for entrainment in jets and plumes. *Physics of Fluids 12*, 8 (2000). 7

[Sta99] STAM J.: Stable fluids. In *Proc. ACM SIGGRAPH '99* (1999), pp. 121–128. 7

[SY05] SHI L., YU Y.: Taming liquids for rapidly changing targets. In *Proc. Symposium on Computer Animation* (2005), pp. 229–236. 2

[TMPS03] TREUILLE A., MCNAMARA A., POPOVIĆ Z., STAM J.: Keyframe control of smoke simulations. *ACM Transactions on Graphics 22*, 3 (2003), 716–723. 2

[WP10] WEISMANN S., PINKALL U.: Filament-based smoke with vortex shedding and variational reconnection. *ACM Transactions on Graphics 29*, 4 (2010). 2

[ZCM11] ZHANG Y., CORREA C. D., MA K.-L.: Graph-based fire synthesis. In *Proc. Symposium on Computer Animation* (2011), pp. 187–194. 2, 3

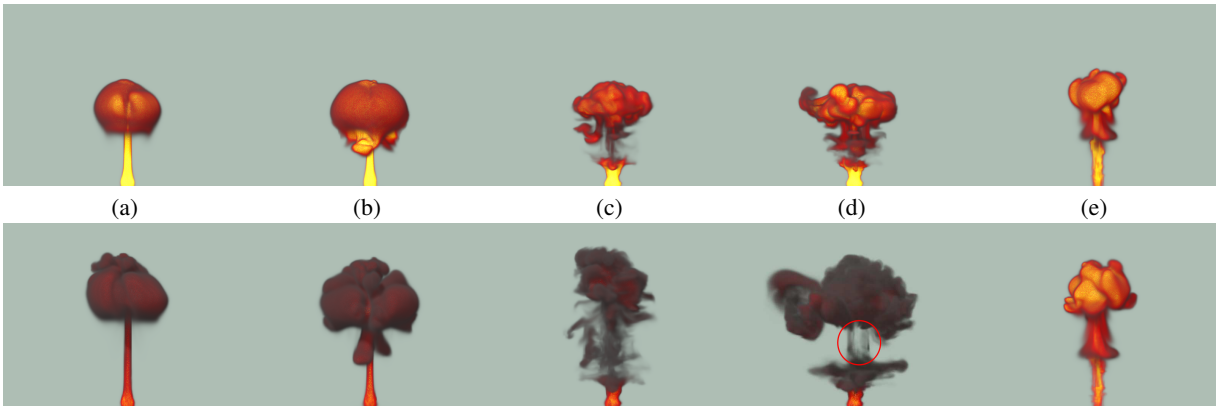

Figure 9: *Comparison between each property (Top row: 150th frame. Bottom row: 200th frame). (a) and (f): buoyant rise. (b) and (g): buoyant rise and expansion. (c) and (h): buoyant rise and circulation. (d) and (i): buoyant rise, expansion and circulation. (e) and (j): based on [FSJ01].*

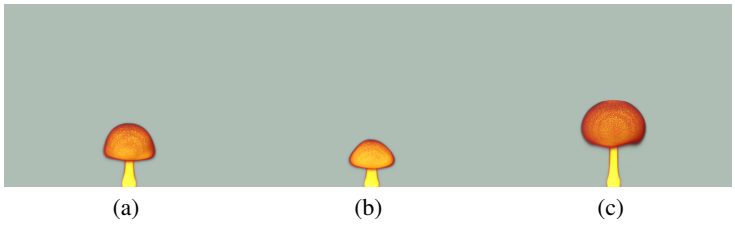

Figure 10: *Comparison by adjusting the parameters of buoyant rise. (a):* α*, (b):* $\alpha \times 0.5$*, (c):* $\alpha \times 2.0$*.*

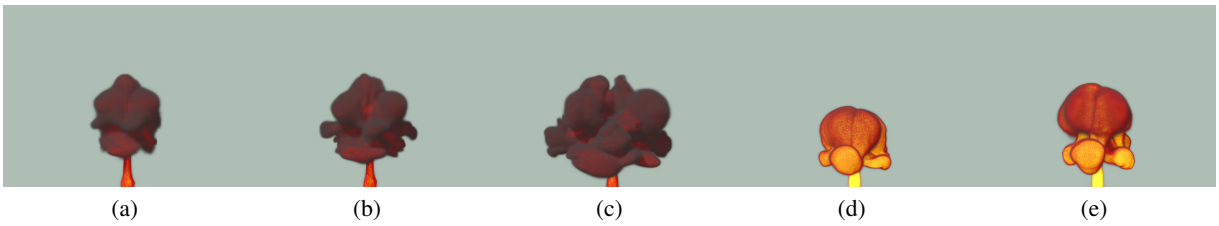

Figure 11: *Comparison by adjusting the parameters of expansion. (a):* β*, (b):* $\beta \times 2.0$*, (c):* $\beta \times 4.0$*, (d) and (e): result for duration time of expansion, 120 frames ((d) for 120th frame, (e) for 150th frame are shown).*

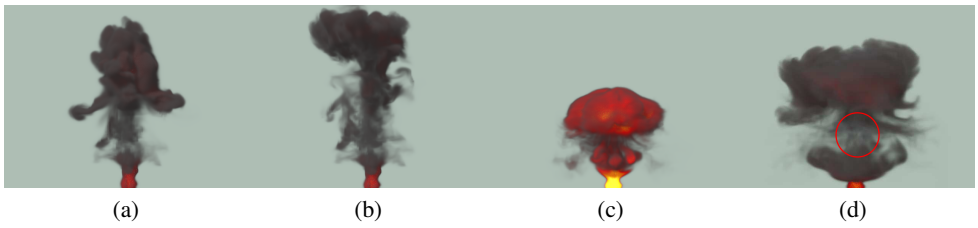

Figure 12: *Comparison by adjusting the parameters of circulation. (a):* γ_{out} *and* γ_{in} *decrease, (b):* γ_{out} *and* γ_{in} *do not decrease, (c) and (d):* γ_{out} *is twice as large as* γ_{in} *((c) for 150th frame, (d) for 200th frame are shown).*