Eurographics/ ACM SIGGRAPH Symposium on Computer Animation (2011)
A. Bargteil and M. van de Panne (Editors)

Procedural Fluid Modeling of Explosion Phenomena Based on Physical Properties

Genichi Kawada and Takashi Kanai

University of Tokyo, Graduate School of Arts and Sciences, Japan

Abstract

We propose a method to procedurally model the fluid flows of explosion phenomena by taking physical properties into account. Explosion flows are always quite difficult to control, because they easily disturb each other and change rapidly. With this method, the target flows are described by control paths, and the propagation flows are controlled by following these paths. We consider the physical properties, which are the propagations of the pressure generated by the ignition, the detonation state caused by the pressure and the fuel combustions. Velocity, density, temperature and pressure fields are generated procedurally, and the fluid flows are computed from these four fields based on grid-based fluid simulations. Using this method, we can achieve a fluid motion that closely resembles one generated solely through simulation. This method realizes the modeling of flows controlled frame by frame and follows the flow's physical properties.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation; I.6.8 [Simulation and Modeling]: Types of Simulation—Animation

1. Introduction

Explosion phenomena are defined as the propagation of fluid, the flowing density of flame with compressible expansion behavior caused by the chemical process consisting of low- and high-speed fluid aspects (see [Bak73] and [HS94]). The high-speed aspect is where ignition occurs and the fluids propagate rapidly by expansion. The low-speed aspect is where the fluids continue to propagate with a gradual loss of velocity, and then converge at the destination. Explosion flows show properties such as whirling and streaming motions throughout the entire process. Ignition causes the propagation of pressure from the explosion source, and this strong pressure is called a shock wave. This pressure determines the physical explosion properties. We focus on a model of the explosion phase prior to the flow reaching its destination, while previously-proposed approaches have dealt with what occurs after the flow reaches its destination.

Explosion behaviors are extremely complex, in particular high-speed flows and large turbulence have to be considered when attempting to express such behaviors. Therefore, explosion flows are difficult to define using simple geometric features or only physical parameter settings at the ignition. Adding to that, a grid size for the simulation typically has to be small enough to realize the whirling and streaming flows. However, with the small grid size it is difficult to realize high-speed flows, because the time steps need to be small according to a CFL condition and that leads to huge computational costs.

In related work, Kim et al. proposed a method to guide flows by generating velocity fields along a path to control smoke as a general fluid [KMD06]. Simulations for explosions are, however, different from simulations for other fluid phenomena, and it is difficult to lead the behaviors with large vortices and high-speed flows using only the method of Kim et al.. Dobashi et al. [DSYA09] proposed a control method for explosion by directly adjusting the velocity fields in grid-based simulation. It is, however, quite difficult for this method to realize the detailed control for high-speed flows as a unique explosion property, or to generate the highly curved flows with occlusions from the source. We believe that explosions must be characterized step by step based on the propagation behaviors of the flows during the simulation. This is because the model should have enough information about the flows during each step to describe the whole propagation in a relatively short time before the flow reaches its destination.

We propose a procedural method to model explosion phenomena by considering the physical properties. In addition

to controlling the explosion flows based on the paths specified by the user, the flow fields are procedurally generated and the grid-based simulations are applied to the fields. In this way, a smooth transition from high-speed flows to low-speed flows can be realized. We generate density, velocity, temperature and pressure fields based on explosion curves derived from the physical explosion models, and realize the complex explosion behaviors including large vortices and pressure change in flows. Our method also takes into consideration the detonation phenomena to cause the drastic change of pressure and the fuel combustion to cause large vortices.

We believe our method has the following advantages:

- While our mathematical equations are directly and theoretically derived from the physics, we first present practical ways to apply those equations to compressible and explosive fluids for computer graphics simulation.
- Although the flow behaviors are less physically accurate than full grid-based simulation, each flow following a control path exhibits the appropriate physical properties.
- The user inputs are the specifications of non-uniform rational B-spline (NURBS) curves and the parameters of the explosion curve. The explosion flows are automatically calculated once the inputs are complete. Therefore, our method realizes modeling of the flows that is free, easy to predict and intuitive for the user.
- In our method, the transition between compressible and incompressible states as a unique explosion property is consistently dealt with by grid-based simulation. After the transition to low-speed flows, various previously-proposed approaches for grid-based simulations can be applied.

The rest of the paper is organized as follows. In Section 2 we briefly review the relevant related work. Section 3 presents the details of our model and the algorithm. In Section 4 we show our experimental results and discuss the advantages and limitations of our approach. Section 5 presents conclusions and suggestions for future work.

2. Related Work

Modeling explosions. Yngve et al. [YOH00] proposed a model in which compressible Navier-Stokes equations with small time steps are used to generate blast waves of explosions. Sewall et al. [SGTL09] introduced a finite volume method for resolving shock waves that requires small time steps to describe the compressible phenomena. In a later study, a divergence modification was added to [Sta99] to handle the reactive ignition described in [FOA03]. While [FOA03] can realize the drastic effects of the ignition, the high-speed flows generated by this method are hard to predict.

Takeshita et al. [TOT*03] modeled flames of explosions using the Lagrangian method. Ihm et al. [IKC04] and Kang et al. [KJI07] extended the method of Takeshita et al. to create a model for explosive fluid by using chemical reaction processes. Gao et al. [GLHB09] proposed a hybrid solver of Lagrangian and Eulerian methods to deal effectively with high- and low-speed fluids in one model. Kwatra et al. [KGF10] proposed a method to physically calculate an explosion's compressible-to-incompressible transitions with larger time steps in a semi-implicit method. Since those previous works determine the explosion behaviors only by the physical parameters, it is difficult to control the complex explosion behaviors.

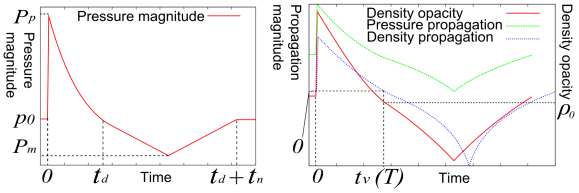

Figure 1: *Explosion curves. Left: Pressure magnitude curve. Right: Density opacity curve, pressure propagation curve and density propagation curve.*

Control for fluid simulations. Treuille et al. [TMPS03] proposed the density keyframing algorithm whereby the driving forces to generate specific keyframes are optimized. To follow up this research, Fattal and Lischeiski [FL04] modeled the algorithm to avoid optimization. To get smoke fluids to follow specified paths, Kim et al. [KMD06] proposed a path-based control method by generating velocity fields. Shi et al. [SY05] controlled target liquids, which change their shapes rapidly by external forces. Thuerey et al. [TKRP09] proposed a control method by decomposing velocity fields into low- and high-frequency components with various control forces.

Pighin et al. [PCS04] edited buoyancy-driven fluids by specifying the density at various keyframes. Beaudoin et al. [BPP01] and Lamorlette et al. [LF02] realized the behaviors of fire by using primitives or particle elements based on physical properties without numerical simulations.

Although various types of fluid control models have been proposed, unfortunately only a few methods for modeling explosion fluids have been developed. The step size needs to be small enough for high-speed explosions, as in the method of Shi et al. [SY05]. This leads to huge computational costs not suitable for user turnaround. Dobashi et al. [DSYA09] first proposed a fluid control model specialized for explosion. However, this method's primary purpose is to effectively generate the end flow as it reaches the destination, rather than to simulate the entire propagation of the flow frame by frame.

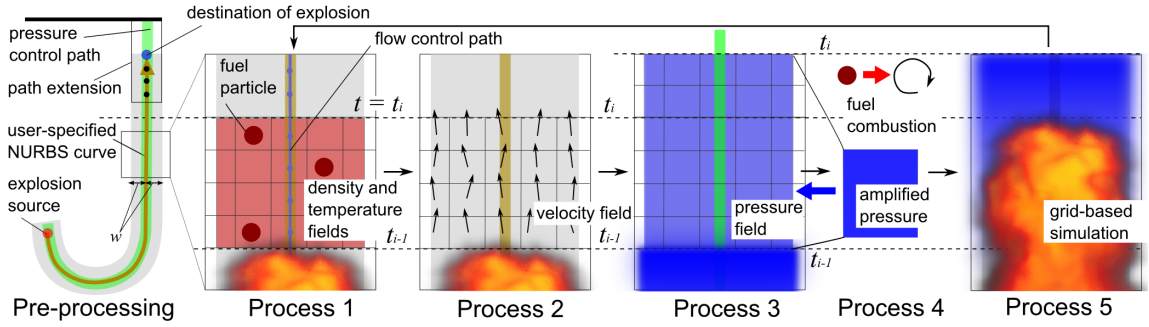

Figure 2: *Our algorithm. Pre-processing: flow control path converted from user-specified NURBS curve; Process 1 and Process 2: density and temperature fields and velocity fields calculated based on density propagation and opacity curves, respectively; Process 3: pressure field determined based on the pressure propagation curve; Process 4: detonation state for drastic pressure change and fuel combustion for large vortices; Process 5: grid-based simulation of the field calculations.*

3. Model and Algorithm

3.1. Model

We define a model based on physical properties to generate the behaviors of flows in explosion phenomena. We consider a one-dimensional model as the physical model. This one-dimensional model represents the propagation properties over time in a certain direction from the source to the destination. Explosion flows are described by applying this one-dimensional model to each of the control paths specified by the user.

For the model to calculate the flows along the control paths, we utilize the *pressure magnitude curve*, *pressure propagation curve*, *density propagation curve*, and *density opacity curve* defined by [Bak73] and [HS94] (or see [GD77]) (Figure 1). The vertical axes are different for each curve, while the horizontal axes of all those curves describe time. The pressure magnitude curve exhibits an important property, and the other curves are derived from this curve. In other words, the pressure obtained by the pressure magnitude curve determines other physical properties (the density opacity, the velocity of density, the magnitude of pressure, and the velocity of pressure).

The magnitude of propagating pressure front p at the explosion can be approximated by the Friedlander equation [Bak73] as described below. We define p as the *standard pressure*. The pressure magnitude curve (Figure 1 left) formularizes the following equation:

$$p(t) = \begin{cases} p_0 + P_p(1-\frac{t}{t_d})e^{-\frac{bt}{t_d}}, & if\ t \le t_d, \\ p_0 - \frac{2P_m}{t_n}(t-t_d), & if\ t > t_d \\ & \&\ t \le t_d + \frac{t_n}{2}, \\ p_0 - \frac{2P_m}{t_n}(t_d+t_n-t), & if\ t > t_d + \frac{t_n}{2} \\ & \&\ t \le t_d + t_n, \\ p_0, & if\ t > t_d + t_n, \end{cases} \quad (1)$$

where p_0 is the ambient pressure, P_p is the peak over pressure, P_m is the minimum negative pressure, t_d is the time to reach p_0, t_n is the time to reach p_0 from t_d, and b is the decreasing coefficient. This curve indicates that a huge pressure arises around the ignition, and that pressure gradually decreases and converges before the flow reaches its destination. On the other hand, the density propagation curve describes the propagation velocity of flow (density), and the density opacity curve describes the opacity (density value) of flow. The pressure propagation curve describes the propagation velocity of standard pressure. The latter three curves are formulated in later sections.

In our method, each explosion flow propagates on each control path until the destination time based on the propagation velocity magnitude of the flow is reached. This velocity magnitude is determined by the explosion curves represented by the one-dimensional model described above. The distance from the ignition source to the destination, that is, the length of each control path, is different, while we also consider all flows to reach the ending points of the paths at the same time. To this end, the propagation velocity magnitude of each flow is scaled based on the actual length of each path. Therefore, the longer the control path is, the larger the actual velocity magnitude becomes, while the relative velocity magnitude on each path between times is maintained.

In contrast, the explosion behaviors for the entire scene domain are realized by the grid-based simulation. Namely, our method combines the grid-based simulation with the generation of velocity, density, temperature and pressure fields from multiple control paths. Therefore, consistency for all explosion behaviors is maintained, while each explosion flow is controlled.

3.2. Algorithm

Figure 2 shows our algorithm. There is one pre-processing in our algorithm. There are also five modeling processes per step. The five processes are executed until the flows reach

the destination, while only Process 5 is executed after the flows reach the destination.

- As Pre-processing, NURBS curves representing the target flows are specified by the user. These curves are automatically converted into flow control paths and pressure control paths (Section 3.3).
- As Process 1, to realize the compressibility of the flow, the density opacity during the current step is calculated. The density and temperature fields at the region along the flow control paths are also generated (Section 3.4.1). Furthermore, fuel particles are randomly generated based on the temperature (Section 3.4.6).
- As Process 2, the propagation velocity magnitude of the density during the current step is calculated, and velocity fields at the region along the paths are generated. Vortex particles are also randomly generated (Section 3.4.2).
- As Process 3, the velocity and the magnitude of the pressure front during the current step are calculated, and the pressure fields at the region along the pressure control paths are generated (Section 3.4.3).
- As Process 4, by detecting the detonation state that will cause the drastic change in the pressure, the pressure fields are amplified (Section 3.4.5). Large vortices are generated from fuel particles by considering the combustion (Section 3.4.6).
- As Process 5, the whole grid domain is calculated by executing grid-based simulation [Sta99] for one step (Section 3.4.4).

Details of each process in our algorithm are described in the following sections.

3.3. User Input and Control Path

The parameters for the pressure magnitude curve (Equation (1)) are determined by the user. Also, NURBS curves (brown curves in Figure 2) are specified to prescribe the directions of the desired target flows. The beginning locations of those NURBS curves describe the explosion sources, and the end locations describe the destination points. Our method considers that the density to represent the flow proceeds along this curve of width w. The propagation time of the flow from the source to the destination is T, which is the actual time corresponding to the time t_v when the propagation magnitude becomes 0 at the density propagation curve (Figure 1 right). Here, t_v is automatically determined once t_d is obtained, and $t_d < t_v$ is always true. The user specifies w and T.

The pressure magnitude curve in Equation (1) determines three curves in Equation (2), (3), and (6) described below. The horizontal axis, t, is scaled by considering that the time when density propagation magnitude becomes 0 in Equation (2) is T.

Actual calculation is executed in the grids covering the entire domain to describe the target explosion. Therefore, each

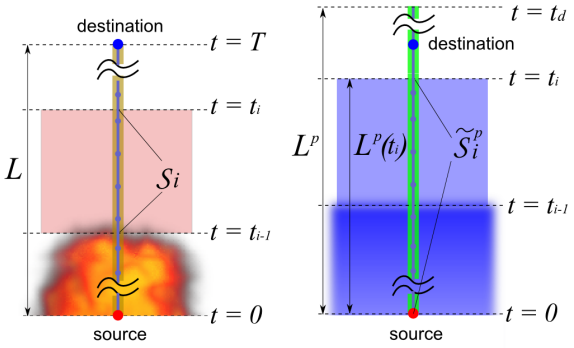

Figure 3: *Left: Density Field. Right: Pressure field.*

NURBS curve is automatically converted into a *flow control path* (blue line in Figure 2) defined as a discretized curve composed of a sequence of center points on the grid.

The pressure propagates faster than the flow, and the propagation distance of the pressure is also longer than that of the flow. Therefore, the *pressure control path* (green line in Figure 2) is generated by extending the curve along the averaged unit tangent vector of the curve's last several points (black points in path extension in Figure 2).

T is discretized by n sets of time interval Δt, and the computation for T is described as n steps. The density flows start from the explosion source at $t = 0$, propagate along the specified curves in each step, and reach the ends of the curves at the n th step (time $t = T$).

3.4. 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. Five processes per step, which are the density and temperature generation, velocity generation, pressure generation, pressure amplification, fuel combustion, and grid-based simulation, are executed in order.

3.4.1. Density and Temperature Field Generation

First of all, the density fields along the flow control paths during step i are generated (Process 1 in Figure 2). Those fields are calculated based on both the density propagation curve to describe the density propagation velocity and the density opacity curve to describe the density value.

The velocity magnitude, $v(t)$, at the density propagation curve is described [HS94] as follows:

$$v(t) = \frac{cp(t)}{\gamma p_0}\left(1 + \frac{\gamma+1}{2\gamma}\frac{p(t)}{p_0}\right)^{-\frac{1}{2}}, \qquad (2)$$

where c is the speed of sound in the air, and γ is the ratio of the specific heats (1.4 at the ambient pressure).

The front-most location of the flow on the path during the current step i is calculated based on this velocity magnitude,

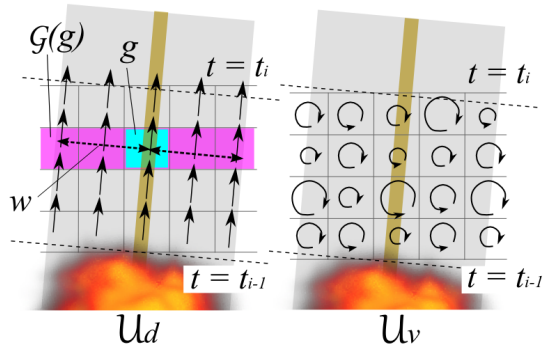

Figure 4: *Velocity field.*

which describes only the relative value. Therefore, this velocity magnitude needs to be scaled to become the magnitude along each actual control path. Namely, we consider that the flow propagates based on the velocity magnitude described by Equation (2), and then the propagation distance from the source until time t_i is $l(t_i) = \sum_{k=1}^{i} v(t_k)\Delta t$. Therefore, the propagation distance until time T is described as $l(T)$. We also let the corresponding length of a control path be L, and then the propagation velocity magnitude on the actual control path becomes $V(t_i) = \frac{v(t_i)L}{l(T)}$.

The density value ρ is determined based on the density opacity curve and ρ at this curve is described [HS94] as follows:

$$\rho(t) = \rho_0 \frac{2\gamma p_0 + (\gamma+1)p(t)}{2\gamma p_0 + (\gamma-1)p(t)}, \tag{3}$$

where ρ_0 is the ambient density specified by the user.

The propagation distance of the flow during step i is $V(t_i)\Delta t$. We define the control path segment corresponding to this distance as $\mathcal{S}_i$, and then the density fields are generated on the grids corresponding to the sweep region made by the width w that surrounds $\mathcal{S}_i$ (Figure 3 left, red region). Indeed, the uniform value $\rho(t_i)$ is allocated at each grid inside this sweep region.

The temperature fields are also generated in the same way as the density fields. The uniform temperature value (specified by the user) is allocated at each grid inside the same sweep region. This temperature value decreases at a constant rate as time passes after the ignition based on the principle of thermal radiation. Those temperature fields computed procedurally are mainly utilized for generating large vortices by combustion (described in Section 3.4.6).

3.4.2. Velocity Field Generation

The velocity field is computed by combining direction velocity field $\mathcal{U}_d$ and vortex velocity field $\mathcal{U}_v$ as follows (Process 2 in Figure 2):

$$\mathcal{U} = \mathcal{U}_d + \mathcal{U}_v. \tag{4}$$

The direction velocity field (Figure 4 left) during the step i is generated at each grid inside the sweep region along $\mathcal{S}_i$ in the same way as the density field. Namely, the direction velocity $\mathbf{u}_d$ is allocated at the each grid, and the velocity is described as:

$$\mathbf{u}_d(\mathcal{G}(g)) = V(t_i)\mathbf{t}(g), \tag{5}$$

where g (Figure 4 left, light blue grid) is a grid on $\mathcal{S}_i$, $\mathcal{G}(g)$ is a set of grids (Figure 4 left, pink region) inside the circle region with width w surrounding g, and $\mathbf{t}(g)$ is the unit tangent vector derived from g on the flow control path. $V(t_i)$ is the same value for all grids inside the sweep region.

The vortex velocity field $\mathcal{U}_v$ (Figure 4 right) is an important explosion property. We apply the vortex particle method by [SRF05] to add fine details of whirling motions. The vortex particles with uniform magnitude are randomly generated at the grids inside the sweep region $\mathcal{S}_i$ in the same way as the direction velocity field. Namely, the vortex velocity field whose center is a particle is added to this region. Particles also advect along the field as the step proceeds. Also, vortex vectors of the field are rotated per step without changing its magnitude [SRF05]. This prevents the total magnitudes on the partial region from increasing as the step proceeds. More dynamic and complex behaviors of explosion phenomena are realized by such vortex velocity fields.

3.4.3. Pressure Field Generation

The pressure field propagating along the control path is calculated according to the propagation velocity determined by the pressure propagation curve (Process 3 in Figure 2). As described in Section 3.3, since the pressure propagates faster than the density flow, the pressure field propagates more forward than the density field. We consider the pressure to propagate until time t_d. The velocity magnitude, $v^p(t)$, at the pressure propagation curve is described [HS94] as follows:

$$v^p(t) = c\left(1 + \frac{\gamma+1}{2\gamma}\frac{p(t)}{p_0}\right)^{\frac{1}{2}}. \tag{6}$$

The actual velocity magnitude on the pressure control path, $V^p(t_i)$, is calculated in the same way as $V(t_i)$, the actual velocity magnitude on the flow control path (Figure 3 right). Namely, the scale adjustment is applied to v^p based on both the control path length and the propagation distance at the pressure propagation curve.

It is very important to note that the method to calculate the pressure field is totally different from the methods to calculate the density and velocity fields, as described previously. The density and velocity fields are generated in the region where the flow propagates only at the current step based on the corresponding explosion curves. In contrast, the pressure distribution during the current step between the pressure front and the pressure end (the source) is independent of the previous frame's distribution, which is calculated based on the previous step. In other words, each frame's distribution

is calculated based only on that frame's information. Therefore, the pressure fields need to be calculated step by step by considering the entire distribution between the front and the end of the current pressure.

Our method to calculate the pressure field is described as follows [WBB99]. First of all, we define $\tilde{\mathcal{S}}_i^p$, the pressure control path segment ranging from its top to its end, as shown in Figure 3 right. The distance for $\tilde{\mathcal{S}}_i^p$ is then $L^p(t_i) = \sum_{k=1}^{i} V^p(t_k)\Delta t$. The pressure value $0.4P(t_i)$ is generated at each grid inside the sweep region along the segment ranging from the end to the point corresponding to $0.5L^p(t_i)$. $P(t_i)$ is the adjusted value determined by considering the standard pressure, $p(t_i)$, to be linearly† proportional to the velocity magnitude of the pressure. The scaled pressure magnitude, $P(t_i)$, is then obtained. For the rest of the segment $\tilde{\mathcal{S}}_i^p$, we consider $0.4P(t_i)$ for the location of $0.5L^p(t_i)$ and $P(t_i)$ for the front. In this case the pressure field values inside the corresponding region are linearly interpolated.

3.4.4. Grid-Based Simulation for Fields

The density, temperature, velocity, and pressure are determined around the control paths as our algorithm proceeds to Process 3. The grid fields for the entire domain are, however, not determined yet. Therefore, by applying grid-based simulation [Sta99] for one step, the density, temperature, velocity, and pressure fields are updated for the entire domain.

Our method updates each field by two-stage calculations, as follows. For the first stage, advection, pressure (projection), and diffusion terms in grid-based simulation [Sta99] are solved for one step without utilizing the pressure fields calculated in Section 3.4.3. The reason is that $p(t)$ is included in the right term of Equation (2) and (3) for the density and velocity fields in Section 3.4.1 and 3.4.2. Therefore, the density and velocity fields take the effects by the pressure calculated in Section 3.4.3 into account.

For the second stage, first the pressure fields are calculated as described in Section 3.4.3. Those calculated fields are used to determine the detonation state in Section 3.4.5. The pressure field for the whole domain is generated by solving the pressure term of the grid-based simulation. This field is mainly utilized for the visualization of pressure (shock wave).

In our method, Process 5 continues after time T while all other processes are terminated. All fields are stably and consistently solved from procedurally determined pieces of

† According to Equation (6), the pressure magnitude needs to be proportional to a square of the velocity magnitude of the pressure. However, we found by our experiments that the pressure magnitude difference between the paths becomes quite huge, so we apply this linear relation equation.

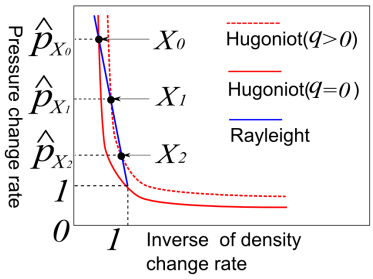

Figure 5: *Rankine-Hugoniot Relation Equations.*

fields, especially for advection, and pressure terms. Therefore, the pressure calculated at the compressible state is not added to the incompressible solver as a huge numerical burden. This is the advantage of our method in terms of preventing the velocity field from diverging, as discussed in [KGF10].

3.4.5. Considering Drastic Change of Pressure

The pressure is changed drastically in the domain where the shock wave pressure propagates. The state that yields the drastic pressure change is called the *detonation state*. In this subsection, we explain our method to realize the pressure change caused by a strong detonation state. By considering such a change in state, large vortices and changes over the entire flows are realized.

We use the Rankine-Hugoniot relation equations [Law06] (see graphs in Figure 5) to express the drastic pressure change, as follows:

$$(\hat{p}+\frac{\gamma-1}{\gamma+1})(\hat{\eta}-\frac{\gamma-1}{\gamma+1}) = \frac{4\gamma}{(\gamma+1)^2}+2q(\frac{\gamma-1}{\gamma+1}), \quad (7)$$

$$M^2 = -\frac{\hat{p}-1}{\gamma(\hat{\eta}-1)}, \quad (8)$$

where $\hat{p}$ denotes the pressure change rate between before and after the pressure propagation, and $\hat{\eta}$ denotes the change rate for the inverse of the density ($= 1/\rho$). Both of these equations describe the relationship between $\hat{p}$ and $\hat{\eta}$. Here, q denotes the amount of heat release per unit mass flux, and M denotes the Mach number ($= v^p(t)/c$). Equation (7) represents two different curves (each is the Hugoniot curve) depending on the q value. Equation (8) is also called the Rayleight curve.

The detonation state when $q = 0$ is the strongest detonation state that yields the largest pressure change between before and after the propagation. We consider the intersection point, (X_0), of the Rayleight curve and the Hugoniot curve when $q = 0$ (solid red line in Figure 5).

In contrast, the detonation state when $q > 0$ yields less pressure change, because the energy is partially used for heat generation. This state is considered to be the state of a usual explosion. There are two intersection points (X_1, X_2) for the Rayleight curve and the Hugoniot curve when $q > 0$ (dotted

red line in Figure 5). In general, the change rate for the pressure on X_0 ($\hat{p}_{X_0}$) has a larger value than the change rates on X_1, X_2 ($\hat{p}_{X_1}, \hat{p}_{X_2}$).

In our method, the strong detonation state is realized by the amplification of the pressure as follows: the pressure change rate $r_{\hat{p}} = \frac{2\hat{p}_{X_0}}{\hat{p}_{X_1}+\hat{p}_{X_2}}$ is multiplied to $P(t_i)$ if $M > 1$ (the condition to become the detonation state by considering the Mach number). $P(t_i)r_{\hat{p}}$ is replaced by the new pressure value in Process 3 (Section 3.4.3). This routine is added as Process 4, as shown in Figure 2.

For the temperature field, the temperature value is amplified by using the same pressure change rate, $r_{\hat{p}}$.

3.4.6. Fuel and Combustion

In this subsection, we consider the relationship between fuel and combustion and realize large vortices, including the flows called turbulence. As an actual routine, fuel particles are generated depending on fuel quantity. We define fuel particles as the seeds to generate the large vortices with the duration time determined by the temperature. In addition, those particles advect themselves with the density flows.

We consider two sphere-shaped explosions, A and B. Let the ignition fuel quantity for A and B be W_A and W_B, respectively, and the distance in a certain direction from the source to the destination for A and B be d_A and d_B, respectively. The relationship $\frac{W_A}{W_B} = \left(\frac{d_A}{d_B}\right)^{\frac{1}{3}}$ exists in those values [HS94]. We now regard W as the generation probability of fuel particles, in other words, the quantity to determine the percentage of particles for a certain region. Therefore, W is determined according to the control path length, d, by utilizing the above relationship. Our method for generating fuel particles is described as follows. In the density field generation in Section 3.4.1, fuel particles are at the same time generated at the probability rate of W, which is much smaller than the rate for the vortex particles here.

Next, the duration time for the vortex is determined based on the temperature of the grid where each fuel particle is generated. According to the Arrehenius equation, the energy, K ($= \beta_1 e^{-\frac{\beta_2}{T_{em}}}$), is calculated, where T_{em} denotes the temperature of the grid, β_1 denotes the frequency factor coefficient, and β_2 denotes the energy coefficient. The calculated energy determines the magnitude of the vortex. This vortex decreases at a constant rate as the time passes, and the duration time is the time when the vortex magnitude becomes 0 from the generation of the vortex.

Similarly this algorithm to determine the duration time can also be applied to the vortex particles described in Section 3.4.2. We also simply realize the density transition of flame to smoke at each grid by considering the temperature below a certain threshold specified by the user. To generate smoke more specifically, density values can be added to the grid with the temperature below the threshold.

Figure 6: *Our method (one path). First row (without the drastic pressure change or combustion). Second row (with only the drastic pressure change). Third row (with only the combustion). Fourth row (with both the drastic pressure change and combustion).*

4. Results and Discussion

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.

Figure 6 shows the comparison with and without considering the drastic pressure change and combustion by using one vertical control path specified from the bottom to the top of the domain (100^3 grids, Left: 10th frame (destination time), Right: 20th frame). The example shows that the quantity of the flow movements is increased only with the drastic pressure change (second rows). Also, the flow disturbances by large vortices can only be seen in the example with the combustion (third rows). Since we apply a 1D model to a 3D domain, as Figure 6 shows, the velocity fields include some discontinuous artifacts, especially in the case of one path compared to the case of multiple paths.

Figure 7 shows the comparison between the previous method [FOA03] that uses only the grid-based method with vortex particles [SRF05] and our method (100^3 grids). As

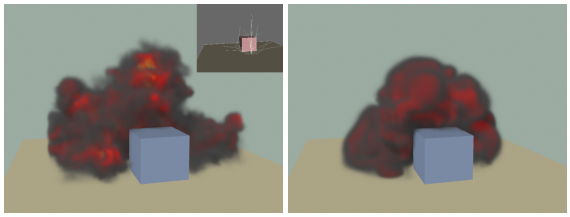

Figure 7: *Comparison of our method with the previous method (ground explosion No. 1 at the 50th frame, which is after the destination). Left: our method (input curves). Right: previous method [FOA03].*

the figure shows, our method realizes similar propagation behaviors from the source as those described using the previous method, while it is difficult to compare the methods under exactly the same conditions. In addition, the example shows that our method models the explosions along the control paths. One of our main contributions is high controllability in flow directions and specific target shapes, which purely grid-based simulation cannot realize.

Figure 9 demonstrates the results using two different sets of control paths by our method. Both results are generated with 200^3 grids. The first two rows in the figure are examples with the paths specified to all directions uniformly, to represent a hemisphere-shaped explosion, while last two rows are the examples with the paths specified to only several specific directions. All examples show our method to freely model the explosions along the paths. The generation of explosions along such path directions have not been addressed in a previous method, and this is the main advantage of our method.

Figure 8 shows a comparison on a camel model with and without the drastic pressure change. Both results are generated with 200^3 grids and the same control paths. Regarding the control paths, the NURBS curves to a head, front legs, and back legs are specified by the user, while the curves to vertices of the body are automatically generated. The top row in the figure shows the example with the drastic pressure change, and the bottom row in the figure shows the example without the drastic pressure change. The largely curved paths in the top row show that the increased quantity of the flow makes the flows larger than the target shape, especially around the neck. For these kinds of shapes, the algorithm to consider geometric features of the paths has to be developed as a future work.

The computational cost per frame is about 60 seconds in both Figure 8 and Figure 9 with 200^3 grids (10 seconds on average for Process 1 - 4). Approximately 30 seconds have to be added when the pressure term in the grid-based simulation is solved for the purpose of visualizing the pressure (shock wave). The computational cost to complete the processes of our method is quite low, while the computation is spent mostly on the pressure term for the visualization.

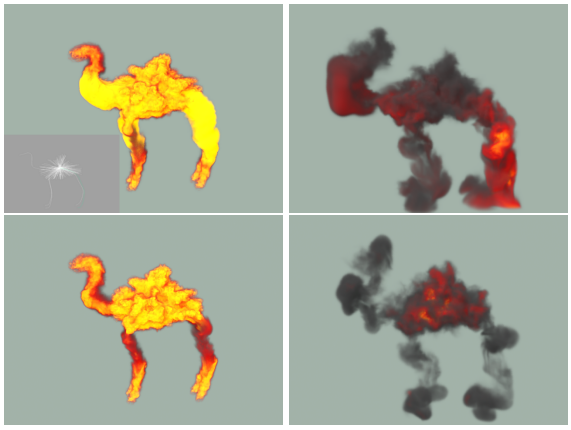

Figure 8: *Our method (camel). Top row (with the drastic pressure change) left: 10th frame (destination time) and input curves. Top row right: 55th frame. Bottom row (without the drastic pressure change) left: 10th frame. Bottom row right: 55th frame.*

5. Conclusions and Future Work

In this paper, we proposed a method to procedurally model explosion phenomena by considering physical properties along the control paths specified by the user. The user's intension can be taken into account by our method, and at the same time explosion flows with complex behaviors can be realized by considering the propagations of the pressure and density flow, the fuel combustion and the detonation state to represent the drastic pressure change.

As our future work, more refined algorithms for generating control paths will be considered. For example, the path widths can be automatically calculated based on the relationship between the target explosion shape and the specified curves, and tree-structural paths can also be considered to describe the flow details. Automatic systems to optimize the number of paths can also be developed by considering the velocity fields among the paths. In addition, interactions with objects have to be taken into account by considering the pressure reflection effects during the interactions. Finally, we would like to reduce computational time by developing a more interactive system for users.

References

[Bak73] BAKER W.: *Explosions in Air*. University of Texas Press, 1973. 1, 3

[BPP01] BEAUDOIN P., PAQUET S., POULIN P.: Realistic and controllable fire simulation. In *Proc. Graphics Interface* (2001), pp. 159–166. 2

[DSYA09] DOBASHI Y., SATO S., YAMAMOTO T., ANJYO K.: Controlling explosion simulation. In *SIGGRAPH ASIA 2009 Sketches* (Yokohama, Japan, December 2009). 1, 2

[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, 7, 8

[GD77] GLASSTONE S., DOLAN P. J.: *The Effects of Nuclear Weapons*. United States Department of Defense and the Energy Research and Development Administration, 1977. 3

[GLHB09] GAO Y., LI C.-F., HU S.-M., BARSKY B. A.: Simulating gaseous fluids with low and high speeds. *Computer Graphics Forum (Proc. Pacific Graphics) 28*, 7 (2009), 1845–1852. 2

[HS94] HETHERINGTON J. G., SMITH P. D.: *Blast and Ballistic Loading of Structures*. Butterworth and Heinemann Ltd., Oxford, 1994. 1, 3, 4, 5, 7

[IKC04] IHM I., KANG B., CHA D.: Animation of reactive gaseous fluids through chemical kinetics. In *Proc. Symposium on Computer Animation* (2004), pp. 203–212. 2

[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, 6

[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. 2

[KMD06] KIM Y., MACHIRAJU R., DAVID T.: Path-based control of smoke simulations. In *Proc. Symposium on Computer Animation* (2006), pp. 33–42. 1, 2

[Law06] LAW C.: *Comubustion Physics*. Cambridge University Press, 2006. 6

[LF02] LAMORLETTE A., FOSTER N.: Structural modeling of flames for a production environment. *ACM Transactions on Graphics 21*, 3 (2002), 729–735. 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

[SGTL09] SEWALL J., GALOPPO N., TSANKOV G., LIN M.: Visual simulation of shockwaves. *Graphical Models 71*, 4 (2009), 126–138. 2

[SRF05] SELLE A., RASMUSSEN N., FEDKIW R.: A vortex particle method for smoke, water and explosions. *ACM Transactions on Graphics 24*, 3 (2005), 910–914. 5, 7

[Sta99] STAM J.: Stable fluids. In *Proc. ACM SIGGRAPH '99* (1999), pp. 121–128. 2, 4, 6

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

[TKRP09] THÜREY N., KEISER R., RÜDE U., PAULY M.: Detail-preserving fluid control. *Graphical Models 71*, 6 (2009), 221–228. 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

[TOT*03] TAKESHITA D., OTA S., TAMURA M., FUJIMOTO T., MURAOKA K., CHIBA N.: Particle-based visual simulation of explosive flames. In *Proc. Pacific Graphics* (2003), pp. 482–486. 2

[WBB99] WINGERDEN K., BJERKETVEDT D., BAKKE J. R.: Detonations in pipes and in the open. In *Proc. the Petro-Chemical Congress* (1999). 6

[YOH00] YNGVE G. D., O'BRIEN J. F., HODGINS J. K.: Animating explosions. In *Proc. ACM SIGGRAPH 2000* (2000), pp. 29–36. 2

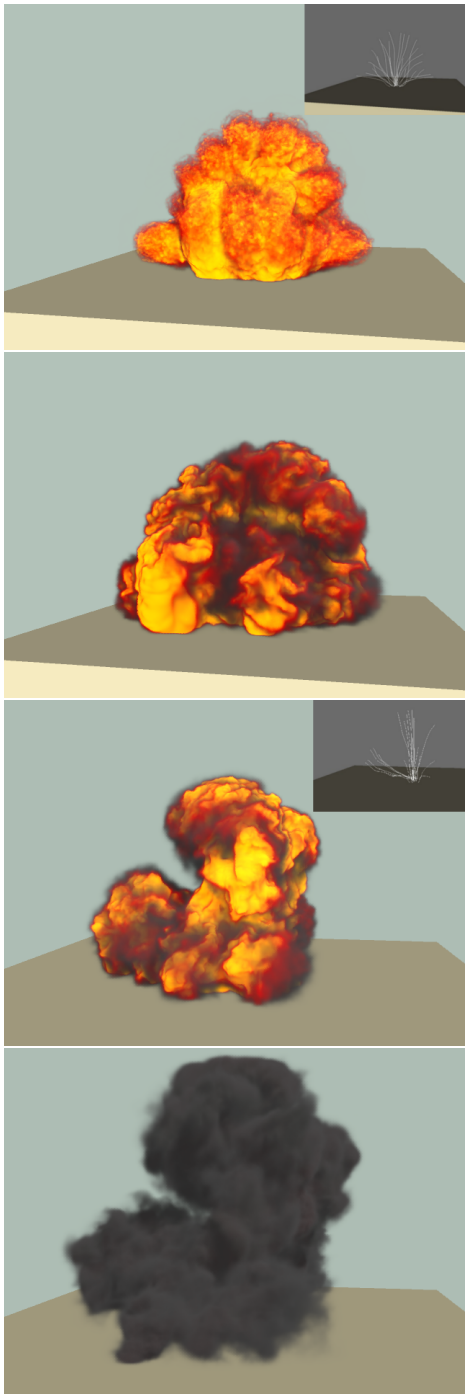

Figure 9: *Our method with both the drastic pressure change and combustion. First row: ground explosion No 2, 10th frame (destination time) and input curves. Second row: ground explosion No 2, 30th frame. Third row: ground explosion No 3, 25th frame (10th frame is destination time) and input curves. Fourth row: ground explosion No 3, 70th frame.*