Baroclinic Turbulence with Varying Density and Temperature

Doyub Kim, Seung Woo Lee, Oh-young Song, and Hyeong-Seok Ko

Abstract—The explosive or volcanic scenes in motion pictures involve complex turbulent flow as its temperature and density vary in space. To simulate this turbulent flow of an inhomogeneous fluid, we propose a simple and efficient framework. Instead of explicitly computing the complex motion of this fluid dynamical instability, we first approximate the average motion of the fluid. Then, the high-resolution dynamics is computed using our new extended version of the vortex particle method with baroclinity. This baroclinity term makes turbulent effects by generating new vortex particles according to temperature/density distributions. Using our method, we efficiently simulated a complex scene with varying density and temperature.

Index Terms—fluid animation, variable density, turbulent flow, vortex particle method

1 Introduction

MANY of the dynamic scenes in motion pictures involve hot, smoky, and violent fluid phenomena, such as a mushroom cloud from a bomb explosion or an ash cloud from a volcanic eruption. Such phenomena can be described in more physically based terms as inhomogeneous fluid dynamical instability resulting from density/temperature fluid flows. Although this instability has been well studied in fluid dynamics for several decades as a turbulence effect, it has rarely been discussed in the field of computer graphics. In this paper, we define such effects arising from varying density and temperature as baroclinic turbulence, and, from the perspective of visual effects, we propose an efficient way of modeling and implementing such baroclinic turbulence effects.

In meteorology, it is said that complex shapes of stratified clouds are made when density is coupled with temperature as well as pressure; this is called an atmosphere baroclinic system. In the area of fluid dynamics, such baroclinic systems have been modeled as a collection of multiple shearing layers resulting in stratified instability turbulence. Compared to systems with homogeneous fluid, flows with variable density and temperature will have additional driving and stabilizing forces that give rise to new instability mechanisms [1]. In addition, many theories and experiments have established that varying density and temperature are dominant factors in simulating complex motion of fluids in mixed inhomogeneous streams [2], [3], [1]. In the field of combustion, this misalignment of the pressure gradient and thermal gradient is regarded as one of the major vorticity generating factors that lead to fire whirls which are easily observed in mesoscale fires and volcanic eruptions [4], [5]. Especially, under conditions of weak mean horizontal flow, the buoyantly forced vortices are mainly generated through the baroclinic and feature the oscillatory nature of heat sources ranging in scales from candles to pool fire [6], [7]. We focus on this vorticity generating factor and adopt the concept of varying density and temperature as the source of instability.

Recently, simulating turbulence effects has been actively studied in the area of computer graphics [8], [9], [10]. These methods mainly focus on generating subgrid turbulence based on procedural or energy-based syntheses. Such approaches are based on the extrapolation method for generating high-frequency flow fields. Other frameworks, such as vortex particle methods, have also been introduced for simulating complex swirls and successfully generated detailed velocity fields on a fine scale. However, as all of these methods are based on homogeneous fluids, they have limitations in creating turbulence effects according to density distribution.

To simulate these baroclinic turbulence effects in an inhomogeneous fluid, the governing equations should encompass effects associated with the variable density and temperature of these systems. Such complex dynamics has been well studied in the field of computational fluid dynamics for several decades, and various types of instability can be simulated using existing frameworks [11], [12], [13]. However, the turbulent motion and complexity of flows with variable density and temperature make numerical simulation of such flows very challenging. In fact, simulation of inhomogeneous fluids using existing methods is impractical, as it requires the direct computation of

- *Doyub Kim is with Carnegie Mellon University, USA. E-mail: doyubkim@andrew.cmu.edu*
- *Seung Woo Lee is with Youngwoo CNI, Korea*
- *Oh-young Song is with Sejong University, Korea*
- *Hyeong-Seok Ko is with Seoul National University, Korea*

the complicated governing equations. To create visually plausible simulation results using those methods, simulations must be performed with a high-resolution setup using high-order numerical schemes. Thus, from the perspective of creating visual effects, the impracticality of such an approach stems mainly from the computational resources required.

Here, we present a new framework for simulating baroclinic turbulent flows. The framework considers spatially varying density and temperature in an efficient way. As mentioned, the computation of density and temperature effects, which involves solving the governing equations under conditions of varying density and temperature, is the most challenging aspect of simulating such flows. In our framework, we approach this problem by hybridizing the conventional grid-based method with a new, extended version of the vortex particle method. First, we use a simple buoyancy model to simulate the average motion of the fluid, and this approximate model is used to solve the global dynamics on a coarse grid. Next, to implement complex instability effects at subgrid resolution, we propose an extended vortex particle solver including baroclinity, which is the key term for modeling spatially varying density. We also extend the conventional baroclinity computation so that the effects of spatially varying temperature can be integrated. This extended version of baroclinity enables generation of different turbulent shapes, according to density/temperature distributions, that cannot be captured in previous homogeneous fluids. Finally, the solutions of our new vortex particle solver are synthesized to produce the baroclinic turbulence effects. Through these pipelines, we efficiently simulated the complex scene of varying density and temperature.

2 Previous Work

After Stam [14] introduced the unconditionally stable fluid solver based on a semi-Lagrangian advection method and implicit-pressure projection, Fedkiw et al. [15] adopted a vorticity confinement method to model the small-scale fluid details that were not captured in a coarse grid. Selle et al. [16] and Park and Kim [17] introduced a Lagrangian vortex particle method, and Bridson et al. [18] introduced a curl noise method for procedural fluid flows. Kim et al. [8] proposed a band-limited wavelet noise method and Schechter and Bridson [9] and Narain et al. [10] used an energy transport model for turbulence effects. Also, Pfaff et al. [19] adopted vortex particle and synthesizing methods using an artificial boundary layer specified by Reynolds modeling. As finite numerical calculations cause vorticity dissipation, high-order advection methods have also been developed, including the back-and-forth error compensation and correction method (BFECC) [20], MacCormack [21] and the constrained interpolation profile [22] method.

From a computational fluid dynamics perspective, the inclusion of a varying density term to model inhomogeneous fluids has been actively studied in other fields of engineering. Thorpe [23] indicated that Kelvin–Helmholtz type instability is one of the main factors causing turbulence. According to Turner [24], a variable density generates vorticity through the baroclinic torque, explained by the curl of the density gradient and the pressure gradient. Brown and Roshko [2] demonstrated this phenomenon experimentally by applying density gradient effects in mixing layers. Corcos and Sherman [25] suggested that the baroclinic secondary Kelvin–Helmholtz-type instability is caused by vorticity braids (large vorticity cores connected by thin vorticity layers). Anderson [26] introduced vortex methods considering both the vorticity and density gradient. Sethian and Ghoniem [12] further developed this method to simulate thermally stratified layers. Reinaud et al. [3] focused on the effects of baroclinity on the two-dimensional shear layer beyond the Boussinesq approximation. Forthofer et al. [5] examined the influence of vortices in generating extreme fire behavior and adopted the baroclinic term as one of main factors to cause these turbulent effects.

3 Coarse-grid Buoyancy Solver

As mentioned in section 1, our framework consists of two major stages. Assuming that an inhomogeneous fluid has spatially varying density and temperature, we first use a buoyancy model to calculate the average motion of the inhomogeneous fluid flow, based on an incompressible flow [15]. In this section, we briefly review this framework.

Our framework assumes a viscous incompressible flow when modeling the inhomogeneous fluid. The incompressible Navier–Stokes equations are given by the momentum conservation equation

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\frac{\nabla p}{\rho} + \nu \nabla^2 \mathbf{u} + \mathbf{B}, \tag{1}$$

and the mass conservation equation

$$\nabla \cdot \mathbf{u} = 0, \tag{2}$$

where $\mathbf{u}$, p, ρ, ν, and $\mathbf{B}$ represent the velocity, pressure, density, viscosity, and body force, respectively. We solve the above equations using the stable fluids framework of Stam [14], which includes solving the pressure Poisson equation

$$\nabla \cdot \left(\frac{\nabla p}{\rho} \right) = \frac{\nabla \cdot \mathbf{u}}{\Delta t}, \tag{3}$$

where Δt is the simulation time-step. The pressure Poisson equation can be solved using an iterative method such as the preconditioned conjugate gradient or multigrid method. However, the above equation involves a variable density term inside the divergence

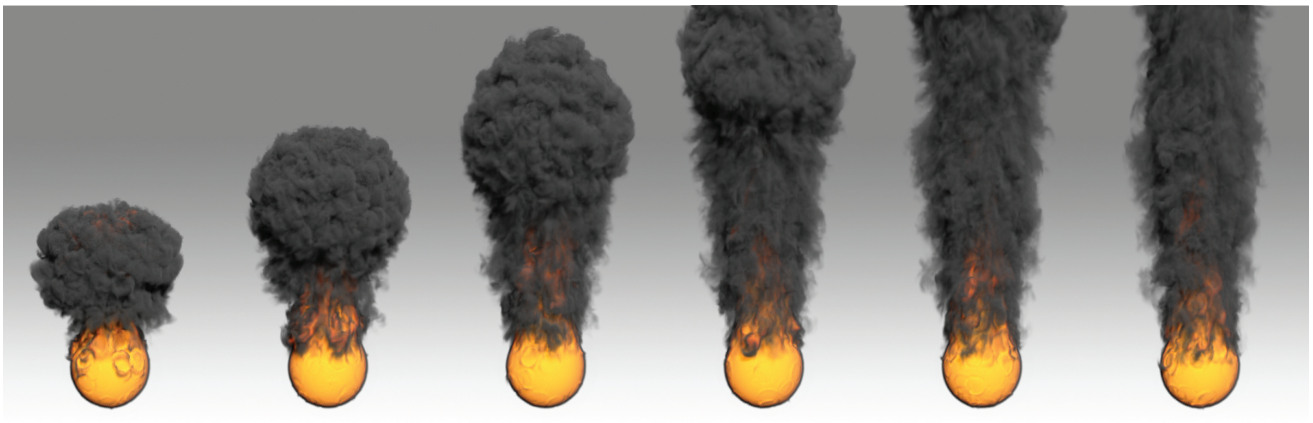


Fig. 1. Simulation of smoke from a hot emitter. The simulation was carried out using a 64×96×64 simulation grid and a 256×384×256 grid was used for the high-resolution synthesis. The simulation took about 9.3 s per time-step.

operator on the left-hand side. This implies that the resulting linear system has a relatively large spectral radius, and hence will require many more iterations to converge. Also, aggregating the thermodynamics terms with the incompressible flow equations is a nontrivial task from the standpoints of both efficiency and consistency. Thus, when solving the dynamics at a coarse level, we adopt the simple buoyancy model of Fedkiw et al. [15],

$$\mathbf{f}_{buoy} = -\alpha\rho_s\mathbf{z} + \beta(T - T_{amb})\mathbf{z}, \tag{4}$$

where α and β are two control parameters, and T and T_{amb} represent the temperature and ambient temperature, respectively. The above equation encompasses the pressure effects of density and temperature. Thus, instead of exactly solving equation 3 with varying density and thermodynamics, we first solve equation 3 with constant density $\tilde{\rho} = 1$,

$$\nabla^2\tilde{p} = \frac{\nabla\cdot\mathbf{u}}{\Delta t}, \tag{5}$$

and solve equation 4 to include varying density and temperature effects. By using the buoyancy model solely, the average motion of the smoke reflects the distribution of density and temperature field in global scale.

The evolution of smoke density and temperature can be computed by solving advection and diffusion equations for each term. The equation for the temperature can be expressed as

$$\frac{\partial T}{\partial t} + (\mathbf{u}\cdot\nabla)T = \mu_T\nabla^2 T, \tag{6}$$

where T and μ_T are the temperature and heat diffusion coefficients, respectively. Similarly, the equation for the smoke density can be written as

$$\frac{\partial \rho_s}{\partial t} + (\mathbf{u}\cdot\nabla)\rho_s = \mu_{\rho_s}\nabla^2\rho_s, \tag{7}$$

where ρ_s and μ_{ρ_s} are the smoke density and diffusion coefficient for smoke, respectively. Here, note that the smoke density ρ_s is *not* the fluid density ρ used in equation 1, but rather is simply an occupation ratio of airborne microscale particles.

4 Baroclinic Vortex Particle Solver

By solving the equations in the previous section, the average motion of a fluid can be simulated. When sophisticated higher-order numerical methods with high-resolution discretization are applied, complex turbulent motion can be generated. However, such an approach is numerically very expensive. An alternative is to use a hybrid method based on a vortex particle solver to generate highly turbulent effects that are unachievable in a coarse grid. As this approach, however, shows limitations when implementing inhomogeneous fluid scenes at the subgrid level, we try to add variable density and temperature effects in the form of baroclinity. Through our new extended version of the vortex particle solver, we can model subgrid dynamics based on the inhomogeneity of the fluid. In this section, we show how we effectively adopt varying density and temperature in terms of baroclinity.

4.1 Solving the Vorticity Equation

Based on the solution computed by the coarse-grid fluid solver, we extend the vortex particle method [16], [27] to solve the dynamics on a fine scale. The vorticity equation describes the evolution of the vorticity $\boldsymbol{\omega} = \nabla\times\mathbf{u}$ of a fluid element as it moves, and can be derived from the conservation of momentum equation 1. In the Lagrangian framework, its general vector form can be expressed as [28]

$$\frac{D\boldsymbol{\omega}}{Dt} = (\boldsymbol{\omega}\cdot\nabla)\mathbf{u} + \frac{1}{\rho^2}\nabla\rho\times\nabla p + \nabla\times\mathbf{B} + \nu\nabla^2\boldsymbol{\omega}. \tag{8}$$

The first term on the right-hand side of the above equation describes the stretching or tilting of the

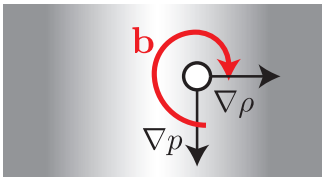


Fig. 2. When fluid motion is dominated by a downward pressure gradient, the baroclinic effect creates vorticity according to equation 9. Here, a brighter color indicates a lower density, and the circle represents a vortex particle.

vorticity resulting from the velocity gradients. The second term on the right-hand side is the baroclinic vector, which we focus on in this section. The body force term $\mathbf{B}$ is usually a gravitational force, and if the gravity is given by a constant number, the third term drops out. The last term describes the diffusion process of the vorticity vector.

4.2 Capturing the Effects of Varying Density and Temperature

In a fluid with variable density, the baroclinic source term appears in the vorticity equation whenever surfaces of constant density and surfaces of constant pressure are not aligned. As shown in figure 2, such misalignments generate a vorticity vector, referred to as the *baroclinic* vector. The baroclinic contribution in equation 8 can be expressed solely as

$$\mathbf{b} = \frac{1}{\rho^2}\nabla\rho \times \nabla p. \tag{9}$$

Instability arising from the baroclinic vector is a fluid dynamical instability of fundamental importance in inhomogeneous fluids with varying density. Unlike the advection, stretching and tilting terms in equation 8, this baroclinic vector can generate vorticity in cases where the pressure gradient and thermal gradient are not parallel. This effect of the baroclinic term has been actively studied in the area of fluid dynamics, including research on the mixing processes of inhomogeneous fluids [3] and extreme fire behavior [5]. From the perspective of modeling the subgrid dynamics of an inhomogeneous fluid, we assume that the baroclinic vector is the major source of the turbulence effects observed in these fluids.

To compute the baroclinic vector, evaluation of the pressure and density gradients is required. However, neither of the terms can be defined explicitly in the approximate buoyancy model since we approximated pressure p to $\tilde{p}$, and density ρ to a constant, for computational efficiency. As the baroclinic vector handles the effects of varying density and pressure, we need to calculate these terms more precisely.

Since, for efficiency, we assumed a constant density when solving the pressure Poisson equation, the resulting pressure at the coarse-grid level is only an imaginary scalar field to make the fluid incompressible. One possible way of computing the physical pressure is to use an equation of state such as $p = k\frac{\rho_0}{\gamma}((\frac{\rho}{\rho_0})^\gamma - 1)$ [29]. Unfortunately, under this approach, the resulting pressure gradient will depend on the density, and taking the cross product with the density gradient will always give a zero baroclinic vector.

Instead, we focus on the approximation process of the buoyancy model we used, and see how the effects of the density and pressure are redistributed. In the original momentum equation without approximation (equation 1), a uniform gravitational force is applied to the fluid field, and the effect of spatially varying density (baroclinity effect) is integrated inside the resulting pressure field. Thus, even with a constant gravitational field, a region with lighter density tends to arise, creating vorticities around it. Meanwhile, in the buoyancy model, the effect of such an ascending current due to gravity is moved to the buoyancy force (equation 4), and the pressure field is only responsible for making the fluid field incompressible, generating vorticities.

Now, we try to reflect the same concept in evaluating the subgrid varying density and pressure terms. From an intuition of the relations between pressure, density, gravity, and buoyancy, we rewrite the second and third terms in equation 8 as

$$\tilde{\mathbf{b}} = \frac{1}{\rho^2}\nabla\rho \times \nabla\tilde{p} + \nabla \times \mathbf{f}_{buoy}. \tag{10}$$

As mentioned earlier, the third term in equation 8 vanishes when assuming a constant gravitational field as the body force. Here, however, we assumed that the effect of a vertical current in the pressure field, resulting from the gravitational effect, is transferred to the buoyancy force, the same as in the coarse-grid solver. Thus, we can consider the body force term as a spatially varying function that needs calculation. As the buoyancy force takes part in the baroclinity effect from the first term of the above equation, we denote these two terms as a pseudo-baroclinity vector $\tilde{\mathbf{b}}$.

In addition to the pressure, the density term should also be evaluated. Here, we define fluid density as a combination of the smoke density and the temperature; this can be written as

$$\rho = a\rho_s + b\frac{1}{T}, \tag{11}$$

where a and b are user-given coefficients. The first term in the above equation indicates that the density is higher when the smoke participation is high. In addition, the spatial distribution of the temperature is reflected by the second term of the above equation; this indicates that a higher temperature gives a lower density. In terms of thermodynamics, density, pressure, and temperature are tightly coupled, requiring delicate computation for evaluation of the density.

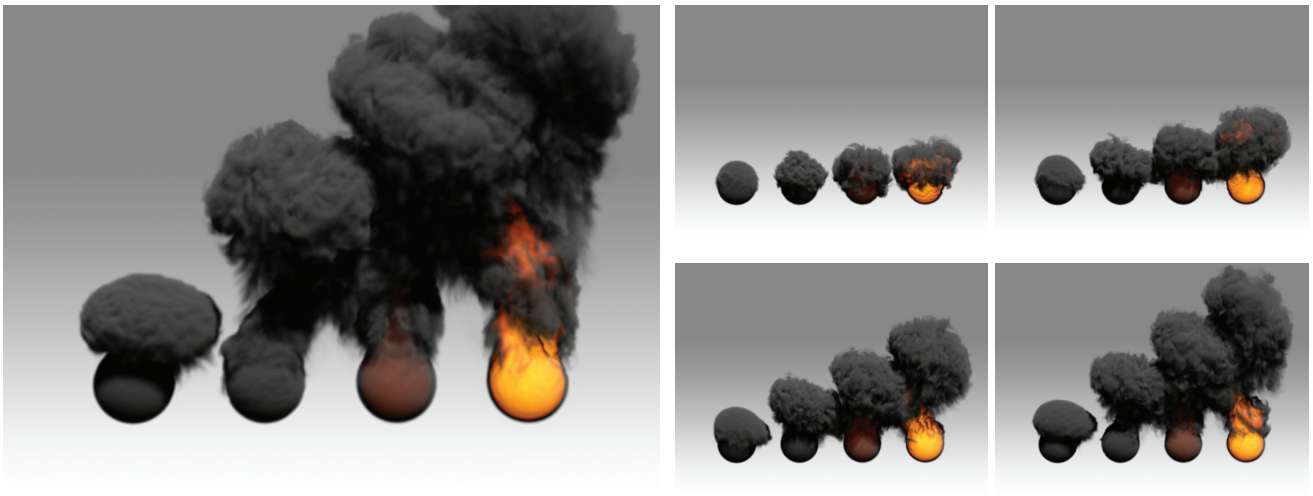


Fig. 3. Simulation of smoke using various heat sources of different strength. The simulation was carried out using a 96×64×96 simulation grid and a 384×256×384 grid was used for the high-resolution synthesis. The simulation took about 15.7 s per time-step.

However, similar to the buoyancy model in Fedkiw et al. [15], for simplicity, we take a phenomenological approach.

As shown in figure 3, our approximated pressure and density can simulate various effects by controlling the heat source. Note that the results with lower temperature show laminar flow, and the results with higher temperature show turbulent flow. Using our method, we effectively realized the baroclinic turbulent effects.

The handling of the boundary conditions for the vortex particle method is not part of our contribution. Sophisticated treatment of turbulence effects near the boundary can be found in [19]. In our simulations, we ignored the turbulence effects caused by fluid–solid interactions, and simply applied a heuristic damping force when vortex particles are near the boundaries.

4.3 Seeding Vortex Particles

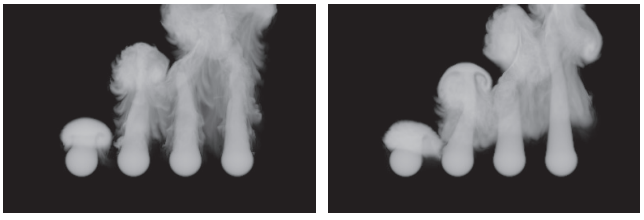
Based on the equations we formulated, the vorticity equation can be computed using equation 8 with the vortex particle method. For the actual simulation, we must define where and how much to seed vortex particles in our simulator.

Instead of seeding vortex particles randomly, we place new particles where the baroclinic contribution is worth capturing. For each grid cell, we first measure the magnitude of the baroclinic vector, and compare it with the user-given criterion c as follows:

$$|\tilde{\mathbf{b}}(\mathbf{x})| > c. \tag{12}$$

If the above condition is met, we perform a Russian roulette,

$$\frac{N_{p,\max}}{N_{grid}}\Delta t > \xi, \tag{13}$$



(A) (B)

Fig. 4. Comparison between our adaptive seeding method (A) with simple random seeding method (B).

to control the birth rate of vortex particles. Here, $N_{p,\max}$ is the maximum number of new particles per second, N_{grid} is the total number of grid cells, and $\xi \in [0,1]$ is a uniform random number. When both criteria are met, we seed a particle with an initial vector of $\Delta t\mathbf{b}$.

Figure 4 compares the results from two different particle seeding strategies. For both simulations, heat sources with various temperatures are used to emphasize the effect of baroclinic vector and the maximum number of vortex particles used was about 4,000. The result from our new seeding policy (A) shows more swirls near the hot region, while the simple random seeding approach (B) gives less turbulent simulation result. Since the simple random seeding approach uniformly distributes the vortex particles even for the region that is not highly important, our adaptive seeding approach shows better result qualitatively.

Definitely, with sufficiently many vortex particles, there's not a huge difference between the both random seeding and adaptive seeding. However, with smaller number of particles, we can see that the result from adaptive seeding is better qualitatively. This can be understood similarly to the sampling methods of Monte-Carlo simulations.

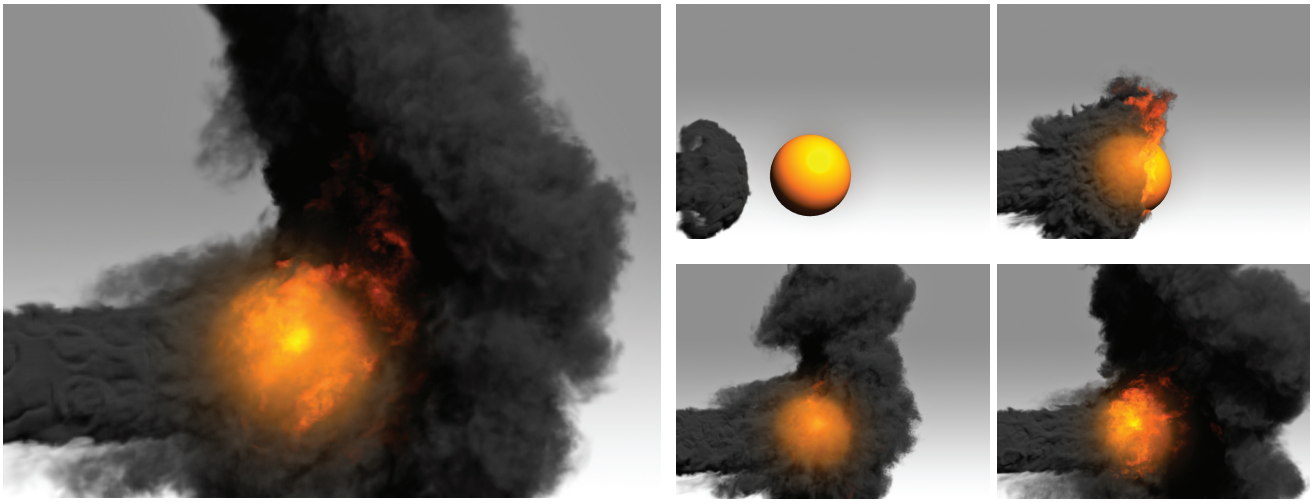


Fig. 5. Simulation of blowing gas to a hot obstacle. The simulation was carried out using a 96×64×96 simulation grid and a 384×256×384 grid was used for the high-resolution synthesis. The simulation took about 16 s per time-step.

5 Velocity Synthesis

After solving the two simulation stages, two sets of solutions should be synthesized. We achieve this using an approach similar to that taken by Yoon et al. [27].

First, we define the vector potential field from the vortex particles. The vector potential can be expressed as

$$\mathbf{\Psi}(\mathbf{x}) = \sum_p w(|\mathbf{x} - \mathbf{x}_p|)\boldsymbol{\omega}_p, \tag{14}$$

which is a direct summation of all of the vortex particles. For choosing w, we use a truncated Gaussian function, as described in Selle et al. [16] and Pfaff et al. [19].

After defining the vector potential field, the high-resolution velocity ($\mathbf{U}$) synthesized from the grid and vortex particles can be expressed as

$$\mathbf{U}(\mathbf{x}) = \mathbf{u}(\mathbf{x}) + \nabla \times \mathbf{\Psi}(\mathbf{x}). \tag{15}$$

As in Kim et al. [8] and Yoon et al. [27], the above $\mathbf{U}$ is then used for computing high-resolution density and for temperature computation.

6 Results

All the experiments reported here were performed on an Intel Core i7 X 980 3.33 GHz processor with 8 GB of memory. We used a uniform regular staggered grid for discretization. The advection term was computed using the BFECC method [20]. For all the experiments in this section, we used $\alpha = 6.250 \times 10^{-4}$, $\beta = 5.000$, $a = 1.000 \times 10^{-5}$, $b = 1.000$, and $c = 0.013$. The vortex particle solver was parallelized using the OpenMP library. We did not use any fire/flame solver in any of the experiments, but the temperature data are rendered with color mapping to emphasize the effects of the heat sources.

Figure 1 shows the results of a simulation of smoke generated from a hot spherical emitter. The simulation was carried out using a 64×96×64 simulation grid, and a 256×384×256 grid for the synthesis. The simulation took about 9.3 s to advance a single time-step, and the maximum number of vortex particles used was 10,000. Note that complex curly structures are generated around the body of smoke. These structures have their own shapes, but are aligned to the center line of the smoke column. The baroclinity is the key term for generating such a pattern in turbulent flow; this is hard to achieve with randomized particles.

Figure 5 shows the results of a simulation of blowing gas to a hot obstacle. The simulation was carried out using a 96×64×96 simulation grid, and a 384×256×384 grid for the synthesis. The simulation took about 16 s to advance a single time-step, and the maximum number of vortex particles used was about 130,000. This example illustrates that gas shows turbulent motions as its temperature increases while passing by the heat obstacle.

Figure 6 shows how our method can create different turbulence effects compared to the other frameworks. The basic configuration is as same as shown in figure 3. The first simulation (A) without the vortex particle method shows uprising smoke without any turbulence effects. The second simulation (B) shows the result from the conventional vortex particle method, which randomly seeds particles and solves vortex equation without baroclinity terms. We can observe that the turbulence effects are generated regardless of the underlying temperature distribution, resulting unrealistic turbulence behavior. The last simulation (C) shows more arranged turbulence effects by seeding and computing the vortex particles according to the spatially-varying density and temperature. In this case, the laminar flow can be observed where the tem-

(A) (B) (C)

Fig. 6. Comparison between our method with other methods under the same configuration. The left-most image (A) shows the result from the simulation without vortex particles. The image in the middle (B) shows the result from the conventional vortex particle method, and the right-most image (C) shows the result from our method.

	No Vortex Particles	Conventional Vortex Particle Method	Our Method
Advection	7.568	7.534	7.474
Projection	0.308	0.313	0.325
Diffusion	0.837	0.841	0.828
Vortex Particle	N/A	5.208	5.879
Total	9.481	15.088	15.641

TABLE 1
Average simulation time (sec) for advancing a single time-step using various methods. Major routines (advection, projection, diffusion, and vortex particle step) are measured seperately.

perature is low, while the turbulent flow is generated at the hot region. The simulation was carried out using a 96×64×96 simulation grid, and a 384×256×384 grid for the synthesis. The simulation took about 15.7 s to advance a single time-step, and the maximum number of vortex particles used was about 40,000 (for both simulations using vortex particles). Table 1 shows the average simulation times for each time-step. Even with the computation of baroclinity effects, the average time for the computation increased negligibly (3.6%).

7 Conclusion

In this paper, we have presented a simple and efficient method for simulating fluid flows with spatially varying density and temperature. To simulate this type of very complex flow, we separated the direct numerical solver into two stages. First, instead of directly computing the contributions of the variations in density and temperature, we adopted a simple incompressible buoyancy solver to compute the average motion of the flow. Then, our method finds the turbulence source in the dynamics by focusing on the varying density and temperature. The turbulent subgrid dynamics of the inhomogeneous fluid was calculated using our new baroclinic vortex particle solver, and, finally, the average motion was synthesized. We would like to emphasize that our framework is a dynamics-driven turbulence model for solving fluids with varying density and temperature that have not been studied enough in computer graphics. Our method can generate more realistic fluid scenes according to density and temperature distributions in a simple and efficient way.

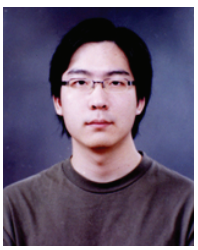
Our simplified approach does, of course, have drawbacks. The main limitation of the proposed approach is that the approximated fluid dynamics does not accurately handle variable density and thermodynamics at the coarse-grid level. However, we believe that, for the purpose of creating visual effects, the practicality of our approach far outweighs the loss in accuracy. Further, for simplicity, our model does not include the variable viscosity of inhomogeneous fluids. However, for buoyancy-driven fluids, such as rising hot smoke, density and temperature variations dominate the motion, hence we assumed that the effects of the variable viscosity could be neglected. Finally, it should be noted that our assumption may not be valid if the density and viscosity contrasts are large (such as at an air–water interface) or if a system contains chemical reactions. For such cases, it would be more suitable to use a multiphase [30] or flame [31] solver. However, we believe that our approach can be applied to such systems to create turbulent motion; this will be the focus of our future work.

ACKNOWLEDGEMENT

This research is supported by the Brain Korea 21 Project in 2011, the IT R&D program of MKE/MCST/IITA (IITA1100090201260001000100100, Development of Digital Clothing Software Technology), ASRI (Automation and Systems Research Institute at Seoul National University), Ministry of Culture, Sports and Tourism (MCST) and Korea Creative Content Agency (KOCCA) in the Culture Technology (CT) Research & Development Program 2011.

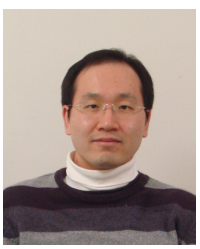
REFERENCES

[1] P. Chassaing, R. A. Antonia, F. Anselmet, L. Joly, and S. Sarkar, *Variable Density Fluid Turbulence*. Kluwer Academic Publishers, 2002.
[2] G. L. Brown and A. Roshko, "On density effects and large structure in turbulent mixing layers," *J. Fluid Mech.*, vol. 64, no. 4, pp. 775–816, 1974.
[3] J. Reinaud, L. Joly, and P. Chassaing, "The baroclinic secondary instability of the two-dimensional shear layer," *Phys. Fluids*, vol. 12, no. 10, pp. 2489–2505, 2000.
[4] J. M. McDonough and A. Loh, "Simulation of vorticity-buoyancy interactions in fire-whirl-like phenomena," in *ASME Conference Proceedings*, vol. 2003, no. 36940. ASME, 2003, pp. 195–201.
[5] J. M. Forthofer and S. L. Goodrick, "Review of vortices in wildland fire," *Journal of Combustion*, vol. 2011, no. 984363, 2011.
[6] S. C. Cheung and G. Yeoh, "A fully-coupled simulation of vortical structures in a large-scale buoyant pool fire," *International Journal of Thermal Sciences*, vol. 48, no. 12, pp. 2187–2202, 2009.
[7] B. M. Cetegen and T. A. Ahmed, "Experiments on the periodic instability of buoyant plumes and pool fires," *Combustion and Flame*, vol. 93, no. 1-2, pp. 157–184, 1993.
[8] T. Kim, N. Thürey, D. James, and M. Gross, "Wavelet turbulence for fluid simulation," *ACM Trans. Graph.*, vol. 27, no. 3, pp. 1–6, 2008.
[9] H. Schechter and R. Bridson, "Evolving sub-grid turbulence for smoke animation," in *Proceedings of the 2008 ACM SIGGRAPH/Eurographics symposium on Computer animation*, 2008.
[10] R. Narain, J. Sewall, M. Carlson, and M. C. Lin, "Fast animation of turbulence using energy transport and procedural synthesis," *ACM Trans. Graph.*, vol. 27, no. 5, pp. 1–8, 2008.
[11] A. J. Chorin, "Numerical study of slightly viscous flow," *J. Fluid Mech.*, vol. 57, no. 4, pp. 785–796, 1973.
[12] J. Sethian and A. F. Ghoniem, "Validation study of vortex methods," *J. Comp. Phys.*, vol. 74, no. 2, pp. 283–317, 1988.
[13] M. C. Soteriou, "Numerical study of turbulent combustion in a shear layer," Ph.D. dissertation, Massachusetts Institute of Technology, 1993.
[14] J. Stam, "Stable fluids," *Computer Graphics (Proc. ACM SIGGRAPH '99)*, vol. 33, no. Annual Conference Series, pp. 121–128, 1999.
[15] R. Fedkiw, J. Stam, and H. W. Jensen, "Visual simulation of smoke," *Computer Graphics (Proc. ACM SIGGRAPH 2001)*, vol. 35, pp. 15–22, 2001.
[16] A. Selle, N. Rasmussen, and R. Fedkiw, "A vortex particle method for smoke, water and explosions," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 910–914, 2005.
[17] S. I. Park and M. J. Kim, "Vortex fluid for gaseous phenomena," in *Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2005, pp. 261–270.
[18] R. Bridson, J. Houriham, and M. Nordenstam, "Curl-noise for procedural fluid flow," *ACM Trans. Graph.*, vol. 26, no. 3, p. 46, 2007.
[19] T. Pfaff, N. Thürey, A. Selle, and M. Gross, "Synthetic turbulence using artificial boundary layers," *ACM Trans. Graph.*, vol. 28, no. 5, pp. 121:1–121:10, 2009.
[20] B. Kim, Y. Liu, I. Llamas, and J. Rossignac, "Advections with significantly reduced dissipation and diffusion," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 1, pp. 135–144, 2007.
[21] A. Selle, R. Fedkiw, B. Kim, Y. Liu, and J. Rossignac, "An unconditionally stable maccormack method," *J. Sci. Comput.*, vol. 35, no. 2-3, pp. 350–371, 2008.
[22] D. Kim, O.-Y. Song, and H.-S. Ko, "A semi-lagrangian cip fluid solver without dimensional splitting," *Comput. Graph. Forum*, vol. 27, no. 2, pp. 467–475, 2008.
[23] S. A. Thorpe, "On standing internal gravity waves of finite amplitude," *J. Fluid Mech.*, vol. 32, no. 3, pp. 489–528, 1968.
[24] J. S. Turner, *Buoyancy Effects in Fluids*. Cambridge University Press, 1973.
[25] G. M. Corcos and F. S. Sherman, "Vorticity concentration and the dynamics of unstable free shear layers," *J. Fluid Mech.*, vol. 73, no. 2, pp. 241–264, 1976.
[26] C. R. Anderson, "Vortex methods for flows of variable density," Ph.D. dissertation, University of California, Berkeley, 1983.
[27] J.-C. Yoon, H. R. Kam, J.-M. Hong, S. J. Kang, and C.-H. Kim, "Procedural synthesis using vortex particle method for fluid simulation," *Comput. Graph. Forum*, vol. 28, no. 7, pp. 1853–1859, 2009.
[28] M. J. Stock, "A regularized invicid vortex sheet method for three dimensional flows with density interfaces," Ph.D. dissertation, The University of Michigan, 2006.
[29] B. Solenthaler and R. Pajarola, "Predictive-corrective incompressible sph," *ACM Trans. Graph.*, vol. 28, no. 3, pp. 1–6, 2009.
[30] F. Losasso, T. Shinar, A. Selle, and R. Fedkiw, "Multiple interacting liquids," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 812–819, 2006.
[31] J.-M. Hong, T. Shinar, and R. Fedkiw, "Wrinkled flames and cellular patterns," *ACM Trans. Graph.*, vol. 26, no. 3, p. 47, 2007.



Doyub Kim received the BS and PhD degrees in the School of Electrical Engineering and Computer Science from Seoul National University (SNU), Korea, in 2005 and 2010, respectively. From 2010-2011, he was a postdoctoral fellow in Automation and Systems Research Institute at SNU. Since 2011, he is a postdoc researcher in Graphics Lab. at Carnegie Mellon University, USA. His primary research interest is in physically-based animation.

Seung Woo Lee received the BS in the School of Electrical Engineering and Computer Science from SNU, Korea, in 2009. From 2009, he has been working as a CAD program developer at Youngwoo CNI, Korea. His primary research interest is in fluid animation and computer vision.



Oh-young Song received the BS, MS, and PhD degrees in the School of Electrical Engineering and Computer Science from SNU, Korea, in 1998, 2000, and 2004, respectively. From 2004-2006, he was a postdoctoral fellow in the School of Electrical Engineering and Computer Science from SNU. Since 2006, he has been an assistant professor in the Department of Digital Contents at Sejong University, Korea. His primary research interests are physics-based animation and character animation.

Hyeong-Seok Ko received the BS and MS degrees in computer science from SNU, Korea, in 1985 and 1987, respectively, and PhD degree in computer science from University of Pennsylvania in 1994. He is a professor in the School of Electrical Engineering and Computer Science at SNU, Korea. He has been working at SNU since 1994. His recent research interest includes digital clothing, fluid animation, hair modeling, and flexible body animation.