

# User-Controllable GPGPU-Based Target-Driven Smoke Simulation

Jihyun Ryu<sup>1</sup> and Sanghun Park<sup>2,\*</sup>

<sup>1</sup> Dept. of Applied Mathematics, Sejong University, Seoul 143-747, Republic of Korea  
ajhryu@sogang.ac.kr

<sup>2</sup> Dept. of Multimedia, Dongguk University, Seoul 100-715, Republic of Korea  
mshpark@dongguk.edu

**Abstract.** The simulation of fluid phenomena, such as smoke, water, fire, has developed rapidly in computer games, special effects, and animation. The various physics-based methods can result in high quality images. However, the simulation speed is also important issue for consideration in applications. This paper describes an efficient method for controlling smoke simulation, running entirely on the GPU. Our interest is in how to reach given user-defined target smoke states in real time. Given an initial smoke state, we propose to simulate the smoke towards a special target state. This control is made by adding special external force terms to the standard flow equation.

**Keywords:** GPGPU, Navier-Stokes equations, interactive simulation.

## 1 Introduction

The modeling of natural phenomena such as smoke, fire, and liquid has received considerable attention from the computer graphics industry. This is especially true for visual smoke models, which have many applications in the creation of special effects and interactive games. It is important to produce highly realistic results as well as simulate effects in real time. This becomes a more challenging if the produced animation can be controlled by users. Recently computer graphics researchers have created simulations for controlling fluids. Treuille et al. [2] introduced a method to control fluid flows to obtain the target shapes. However, this method is too slow in shape controlled flow simulation. In fact, in real-time applications such as computer games, the simulation speed is more important than image quality. This paper presents a method of controlling interactive fluids in real time using the GPU (Graphics Processing Unit). It is based on the results of Fattal et al. [1], and our goal is to perform all the steps on the GPU. This technique can interactively create target shapes using computer-generated smoke.

## 2 Mathematical Background

To simulate the behavior of fluid, we must have a mathematical representation of the state of the fluid at any given time. The greatest quantity to represent

---

\* Corresponding author.

is the velocity of the fluid. But the fluid’s velocity varies in both time and space, so we represent this as a vector field. The key to fluid simulation is to take steps in time and determine the velocity field at each time step. We can achieve this by solving special equations. In physics, we assume an incompressible and homogeneous fluid for fluid simulation. This means that density is constant in both time and space. Under these assumptions, the state of the fluid over time can be described using the Navier-Stokes equations for incompressible flow:  $\frac{\partial u}{\partial t} = -u \cdot \nabla u - \nabla p + \nu \nabla^2 u + F$ ,  $\nabla \cdot u = 0$  where  $u(x, t)$  is the velocity vector of the position vector  $x$  at time  $t$ ,  $p(x, t)$  is the pressure,  $\nu$  is the kinematic viscosity, and  $F$  represents any external forces that act on the fluid. In our case, we may simulate the non-viscous fluid and therefore solve the Euler Equation with  $\nu = 0$ ;

$$\frac{\partial u}{\partial t} = -u \cdot \nabla u - \nabla p + F, \quad \nabla \cdot u = 0 \quad (1)$$

Let  $\rho = \rho(x, t)$  be the density scalar field at position  $x$  and time  $t$ . In order to describe the transport of the smoke along the fluid’s velocity fields, we solve the additional equation;

$$\frac{\partial \rho}{\partial t} = -u \cdot \nabla \rho \quad (2)$$

Moreover, the external forces term provides an important means of control over the smoke simulation. In the result of Fattal et al. [1], the special external term  $F(\rho, \rho^*)$  depends on the smoke density  $\rho$  with the target density  $\rho^*$ . This result has the same direction as the gradient vector of  $\rho^*$ . In addition, a “normalized” gradient can be used by  $F(\rho, \rho^*) \propto \frac{\nabla \rho^*}{\rho^*}$ . The blurring filter of  $\rho^*$  must have sufficiently large support, since the target density  $\rho^*$  is constant and  $\nabla \rho^* = 0$ . In order to ensure  $\nabla \rho^* \neq 0$ , the blurred version of  $\rho^*$ , denoted by  $\tilde{\rho}^*$ , can be used. The force is  $F(\rho, \rho^*) = \tilde{\rho} \frac{\nabla \tilde{\rho}^*}{\rho^*} \equiv F_{df}$  where  $F_{df}$  is named the “driving force”. In summary, we have two modified equations for the controlled fluid simulations. The first is the advection equation for density; the second is the momentum equation using the driving force.

$$\frac{\partial \rho}{\partial t} = -u \cdot \nabla \rho, \quad \frac{\partial u}{\partial t} = -u \cdot \nabla u + \nabla p + F_{df} \quad (3)$$

In addition, another external force, denoted by  $F_{ui}$ , for user interaction, can be applied to fluid by clicking and dragging with the mouse. The force is computed from the direction and length of the mouse drag.

### 3 Implementation Details

The goal of this research has been to improve user interaction with computer-generated smoke. We implemented GPGPU (General-Purpose computation on GPUs) based techniques to simulate dynamic smoke that can be described by PDEs. Algorithm 1 shows a method to process the user-controllable interactive target-driven simulation. Fattal et al. [1] introduced offline target-driven

**Algorithm 1.** User-controllable interactive target-driven simulation algorithm

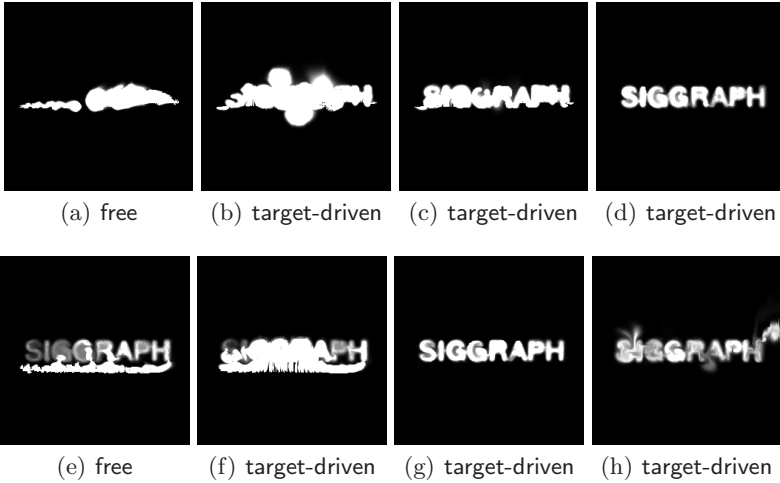
---

```

1: Load a target key-frame, initialize target density field  $\rho^*$ 
2: while (1) do
3:   Setup external force  $F_{ui}$  and density field  $\rho_{ui}$  // from user interaction
4:    $\rho \leftarrow \tilde{\rho}$ ,  $u \leftarrow u + F_{ui}$  // apply gaussian filter and add external force
5:   if (target-driven) then
6:      $F_{df} \leftarrow \frac{\nabla \tilde{\rho}^*}{\rho^*} \rho$ ,  $u \leftarrow u + F_{df}$  // add driving force
7:   end if
8:    $v \leftarrow \nabla \times u$ ,  $u \leftarrow u + v\Delta t$  // vorticity
9:    $\frac{\partial u}{\partial t} = -u \cdot \nabla u$  // advection of velocity
10:   $\nabla^2 p = \nabla \cdot u$ ,  $u \leftarrow u - \nabla p$  // projection
11:   $\rho \leftarrow \rho + \rho_{ui}$  // add density
12:   $\frac{\partial \rho}{\partial t} = -u \cdot \nabla \rho$  // advection of density
13:  draw( $\rho$ )
14: end while

```

---

**Fig. 1.** User-controllable interactive simulation snapshots

animation on CPUs, our algorithm is based on this technique. In free simulation mode, users can generate external force  $F_{ui}$  and density field  $\rho_{ui}$  from mouse movement. The users can change the current simulation mode to target-driven by pressing a keyboard button, and the driving force  $F_{df}$  is applied to current velocity field  $u$ . This results in driving the current density field  $\rho$  to a pre-defined target image  $\rho^*$ . Gaussian filtering for the blurring operation can be implemented in a fragment shader, since recent shading languages support nested loops in shader programs. All of the operations in the algorithm are implemented efficiently as fragment shaders to maximize the degree of rendering speed enhancement; it is possible for users to control smoke simulation interactively.

**Table 1.** Simulation speeds (frames per sec) where resolution of display windows is  $256 \times 256$  and  $n_{ji}$  is the number of Jacobian iteration. The results were evaluated on a general purpose PC, equipped with a 3.0 GHz Intel Pentium D processor, 1 GB of main memory, and a graphics card with an NVIDIA GeForce 7900 GT processor and 256 MB of memory.

$n_{ji}$	Grid of simulation fields		
	$128 \times 128$	$256 \times 256$	$512 \times 512$
50	59.64	15.03	4.60
150	58.01	14.99	3.99
300	29.62	11.99	3.32

To verify the effectiveness of our proposed GPGPU-based system, in smoke simulation, we present timing performances for solving the Navier-Stokes equations on different sized grids. Fig. 1 shows the snapshots of the smoke simulation sequence. The resolution of each image is  $256 \times 256$ , and starting simulation mode is **free**. The system allows users to change the current simulation mode to either **target-driven** or **free**. In (a), users can interactively generate smoke by calculating external force  $F_{ui}$  and density fields  $\rho_{ui}$  through mouse movement in **free** mode. In selecting **target-driven** mode, current smoke starts moving to a pre-defined target key-frame (in this example, “SIGGRAPH” image was used for the target key-frame) shown from (b) to (d). When users choose **free** mode again, the driving force  $F_{df}$  no longer affects simulation operation and users are able to apply their interaction to the current simulation field shown in (e). The status of **target-driven** mode re-activated without any user interaction is shown in (f) and (g). We can see that the image quality of (g) is almost identical to that of (d). When the mode is switched to **target-driven** after running in **free** mode for a few of seconds, the situation where smoke gathers around the target key-frame is shown in (h). Table 1 shows the simulation speeds on different simulation grids where  $256 \times 256$  of fixed display window resolution is used. The system can run smoke simulation at about  $12 \sim 15$  frames per second, where both the grid of simulation fields and the resolution of display windows are  $256 \times 256$ .

## 4 Concluding Remarks

Interactive applications, such as computer games, demand realism, but cannot afford to sacrifice speed to achieve this requirement. We developed user-controllable smoke simulation techniques based on GPGPU computations with these requirements as consideration. This permits the user to select and change the behavior of features as the simulation progresses. Users can create and modify density and flow elements of Navier-Stokes simulation, through a graphic user interface. This system also provides an interactive approach to simulate the smoke towards a special target state.

## Acknowledgements

This work was supported by the Korea Research Foundation Grant funded by the Korean Government (MOEHRD) (KRF-2006-331-D00497) and Seoul R&BD Program (10672).

## References

1. Fattal, R., Lischinski, D.: Target-driven smoke animation. *Transactions on Graphics*, Vol. 23, No. 3. ACM (2004) 441–448
2. Treuille, A., McNamara, A., Popovic, Z., Stam, J.: Keyframe control of smoke simulation. *Transactions on Graphics*, Vol. 22, No. 3. ACM (2003) 716–723