Visual Simulation of Flow

Arie Kaufman[1] and Ye Zhao[1]

1 **Department of Computer Science**
Stony Brook University
Stony Brook, NY 11794, USA
{ari,yezhao}@cs.sunysb.edu

Abstract

We have adopted a numerical method from computational fluid dynamics, the Lattice Boltzmann Method (LBM), for real-time simulation and visualization of flow and amorphous phenomena, such as clouds, smoke, fire, haze, dust, radioactive plumes, and air-borne biological or chemical agents. Unlike other approaches, LBM discretizes the micro-physics of local interactions and can handle very complex boundary conditions, such as deep urban canyons, curved walls, indoors, and dynamic boundaries of moving objects. Due to its discrete nature, LBM lends itself to multi-resolution approaches, and its computational pattern, which is similar to cellular automata, is easily parallelizable. We have accelerated LBM on commodity graphics processing units (GPUs), achieving real-time or even accelerated real-time on a single GPU or on a GPU cluster. We have implemented a 3D urban navigation system and applied it in New York City with real-time live sensor data. In addition to a pivotal application in simulation of airborne contaminants in urban environments, this approach will enable the development of other superior prediction simulation capabilities, computer graphics and games, and a novel technology for computational science and engineering.

1998 ACM Subject Classification I.3.1 Hardware Architecture, I.3.8 Applications, J.2 Physical Sciences and Engineering

Keywords and phrases Lattice Boltzmann Method, Amorphous phenomena, GPU Acceleration, Computational Fluid Dynamics, Urban Security

Digital Object Identifier 10.4230/DFU.SciViz.2010.246

1 Introduction

Visually reproducing flow phenomena in all its richness and complexity has been an exciting endeavor in computer graphics and visualization. The purpose can be categorized into two main topics. One is to enrich the virtual environment with photorealistic images and animations of natural scenery. We have seen astounding appearances in the movies and games of streaming water, flaming fire, turbulent smoke, and so on. In these applications, the requirement is to visually convince the observers that they have seen duplicated reality. The other topic is the correct flow simulation, which demands the accuracy and precision besides the visual authenticity. Such simulations have a broader scope of applications in scientific, environmental and security areas with its reasonable behavior replication and believable picture making.

We have now entered a new era in computer graphics with the advent of hardware accelerated programmable rendering and shading. With the programmability of the graphics processing unit (GPU), combined with the increased performance of CPUs, we can now start to simulate flow phenomena at interactive rates. Direct computational fluid dynamics (CFD)

Scientific Visualization: Advanced Concepts.
Editor: Hans Hagen; pp. 246–258
Dagstuhl Publishing
Schloss Dagstuhl – Leibniz Center for Informatics (Germany)

solvers of the complex Navier-Stokes (NS) equations have been introduced to benefit the graphics and visualization applications.

Procedure techniques have been applied for many years to model gaseous phenomena. They [4, 3, 27, 29, 20, 13] use mathematical functions and algorithms to define the appearance of the objects. Particle-based Methods [26, 23, 25, 19, 9, 28] have also been widely used in amorphous phenomena modeling because they are computationally inexpensive, flexible to control the behaviors and easy to fit into the user-interaction paradigm.

Over the past decades, the application of CFD methods for solving the NS equations has led to significant advances in the modeling of fluid phenomena. Foster and Metaxas [10, 11] developed physically-based methods for the realistic animation of fluids by solving the NS equations. Based on a stable fluid solver [30], various realistically looking smoke [8], water [5], flame [21], viscoelastic fluids [12] and flows on surfaces [31] were generated.

A promising and relatively new CFD method, Lattice Boltzmann Method (LBM), has been introduced by our group to the graphics and visualization community in 2002 [33]. Extensive research has been conducted on using the LBM to model various flow phenomena [34, 1, 7, 24, 35, 36, 40, 39]. The LBM solves the fluid dynamics within the framework of statistical mechanics, where the microscopic physics of fluid particles are modelled and the macroscopic averaged properties obey the desired NS equations. Its microscopic kinetic equation provides many advantages, including easy implementation of boundary conditions and fully parallel algorithms. The LBM can handle any arbitrarily-shaped inside objects and even dynamic boundaries of moving objects. The computation of the LBM is inherently local and explicitly parallel, which is easy to accelerate on the modern graphics processing unit (GPU) to achieve great simulation performance. Meanwhile, multi-resolution LBM approaches have been applied to optimize the use of computational resources for large-scale simulations. The LBM has been successfully used in modeling various amorphous phenomena and simulating contaminants dispersion in urban environments. It has the great potential to benefit researchers and end-users in a variety of applications in education, entertainment and scientific simulations.

2 Lattice Boltzmann Method

The LBM is a relatively new approach in CFD, inspired by cellular automata, that models Boltzmann particle dynamics on a lattice. In the case of a fluid, for example, the Boltzmann equation expresses how the average number of flow particles with a given velocity changes between neighboring lattice sites due to inter-particle interactions and ballistic motion. In the LBM, the variables associated with each lattice site are the particle velocity distributions that represent the probability of the presence of a flow particle with a given velocity. The set of velocities in the model is discrete, being defined by the number, orientation, and length of lattice links. Particles stream between neighboring sites synchronously in discrete time steps. Between streaming steps, they undergo collision. The Bhatnager, Gross, Krook (BGK) model [32] is commonly employed to represent the collisions as a statistical redistribution of momentum, which locally drives the system toward equilibrium. As in kinetic theory, the collisions conserve mass and momentum.

The LBM simulation of 3D flow field is generally performed on a 3D lattice where each lattice site has a number of links representing the velocity vectors (including the zero velocity) to its neighbors. As illustrated in Figure 1, this lattice cell is part of a 3D lattice called D3Q13, which includes the center site with zero velocity and the twelve minor-diagonal neighbor links. Stored at each lattice site are 13 particle distributions associated with the

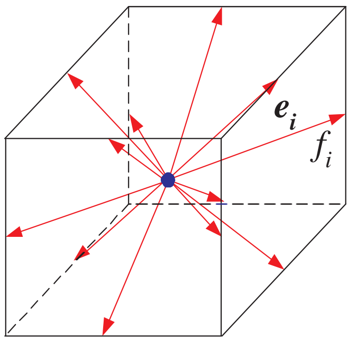


Figure 1 The D3Q13 lattice geometry. The particle distribution f_i is associated with the link corresponding to the $\mathbf{e}_i$ velocity vector.

13 velocity vectors. We denote these as f_i where i identifies a particular velocity vector $\mathbf{e_i}$ among the 13. The macroscopic flow density, ρ, and the macroscopic flow velocity, $\mathbf{u}$, are computed from the particle distributions:

$$\rho(\mathbf{r},t) = \sum_i f_i(\mathbf{r},t) \tag{1}$$

$$\mathbf{u}(\mathbf{r},t) = \frac{1}{\rho(\mathbf{r},t)} \sum_i f_i(\mathbf{r},t)\mathbf{e}_i \tag{2}$$

Using the BGK model, the Boltzmann dynamics can be represented as a two-step process of collision and ballistic streaming:

$$f_i(\mathbf{r},t^+) = f_i(\mathbf{r},t) - \frac{1}{\tau}(f_i(\mathbf{r},t) - f_i^{eq}(\mathbf{r},t)) \tag{3}$$

$$f_i(\mathbf{r}+\mathbf{e}_i,t+1) = f_i(\mathbf{r},t^+) \tag{4}$$

Note that we use the notation t^+ to denote the post-collision particle distribution. Also, f_i^{eq} represents the local equilibrium particle distribution, which is given by

$$f_i^{eq}(\rho,\mathbf{u}) = \rho(A + B(\mathbf{e}_i\cdot\mathbf{u}) + C(\mathbf{e}_i\cdot\mathbf{u})^2 + D\mathbf{u}^2). \tag{5}$$

The constant τ represents the relaxation time scale that determines the viscosity of the flow, while A through D are constant coefficients specific to the chosen lattice geometry. The equilibrium particle distribution comes in as a consequence of the BGK collision model. It is a local particle distribution whose value depends only on conserved quantities - the macroscopic mass ρ and momentum $\rho\mathbf{u}$. Its form may be recognized as the Taylor expansion of the 3D Maxwell velocity distribution to second order. Since the evaluation of Equations 1 through 5 requires only local particle distributions, their acceleration on graphics hardware is efficient (see Section 3). Only one parameter, τ, is used to control the flow behavior in collision (Equation 3). Therefore, this primary LBM is called the single-relaxation-time LBM (SRTLBM). A Smagorinsky subgrid model [36] can be applied to achieve high Reynolds numbers flows without incurring numerical instability, which only involves local particle distribution values and retains the LBM parallelizability.

2.1 Multiple-relaxation-time LBM

Even with the subgrid method, the SRTLBM is prone to unstable numerical computation when used for low viscosity fluids (high turbulent fluids) or incorporated with temperatures or body forces. Multiple-relaxation-time Lattice Boltzmann Model (MRTLBM) is a new general version of LBM developed by d'Humiéres et al. [2]. This collision model abandons SRTLBM to achieve better numerical stability and greater flexibility in selecting the transport coefficients. The essential idea is to make a change of basis from phase space (i.e., the space of the distributions f_i) to the space of hydrodynamic moments (i.e., density, momentum, energy, etc.) and to perform the collision step in the latter space. As in the BGK model, collisions are implemented via a relaxation, but in the moment space each moment is allowed to relax individually. Although the relaxation rates are not all independent, the additional flexibility allows one to maneuver the model into regions of higher stability while decoupling some of the transport coefficients. After relaxation, the inverse transformation is applied to return to phase space where streaming, boundary update rules, and additional micro-physics are implemented as before.

Mathematically, the change of basis from the space of distributions to the space of moments is given by:

$$|m\rangle = M|f\rangle, \qquad |f\rangle = M^{-1}|m\rangle \tag{6}$$

$$|f\rangle = (f_0, f_1, \ldots, f_{12})^T, \tag{7}$$

$$|m\rangle = (m_0, m_1, \ldots, m_{12})^T, \tag{8}$$

where T denotes the transpose. Each of the 13 moments $\{m_i|(i = 0, 1, \ldots, 12)\}$ has a physical meaning. For example, m_0 is the mass density ρ, $m_{1,2,3}$ are the components of the momentum vector $\mathbf{j}$, m_4 is the energy, and the other higher order moments are components of the stress tensor and other high order tensors. The rows of the matrix M relate the distributions to the moments. For example, since $\rho = \sum_i f_i$, the first row of M consists of all ones. Although the values of the distributions and the moments vary over the nodes of the lattice, the matrix M is constant for a given lattice.

In MRTLBM, the two step process of collision and streaming becomes:

$$|f(\mathbf{r}, t^+)\rangle = |f(\mathbf{r}, t)\rangle - M^{-1}S[|m(\mathbf{r}, t)\rangle - |m^{eq}(\mathbf{r}, t)\rangle] \tag{9}$$

$$|f(\mathbf{r} + \mathbf{e_i}, t + 1)\rangle = |f(\mathbf{r}, t^+)\rangle \tag{10}$$

The components of the vector $|m^{eq}\rangle$ are the local equilibrium values of the moments. Among them, the mass density and the momentum (m_0 to m_4) are conserved. Expressions for the nonconserved moments depend only on local values of the conserved moments [15]. The matrix S in the collision equation is a diagonal matrix whose elements are the relaxation rates, $\{s_i|(i = 0, 1, \ldots, 12)\}$. Their values are directly related to the kinematic shear and bulk viscosities, ν and ξ, respectively:

$$\nu = \frac{1}{4}(\frac{1}{s_6} - \frac{1}{2}), \tag{11}$$

$$\xi = (\frac{2}{3} - \gamma c_{s0}^2)(\frac{1}{s_5} - \frac{1}{2}), \tag{12}$$

where γ is the specific heat and c_{s0} is the isothermal speed of sound. The user has the freedom to choose the flow parameters to define characteristics of the fluid being modeled. This choice then determines the relaxation rates.

MRTLBM can also accommodate a body force due to gravity, sensor readings or some other external field. This is implemented by adding the force $\mathbf{F}$ to the momentum, $\mathbf{j}' = \mathbf{j} + \mathbf{F}\delta t$

(typically, $\delta t = 1$). In practice, for stability, the force term is executed in two steps, one-half before the relaxation step and one-half after.

To capture thermal effects, temperature is coupled to MRTLBM through the energy moment that the model exposes. For the D3Q13 lattice, the energy equilibrium is modified as follows:

$$(m_4^{eq})' = m_4^{eq} + cT, \tag{13}$$

where the temperature T is coupled with a constant coefficient c. The heat transfer here is modeled separately with a standard diffusion-advection equation, given rise to the LBM:

$$\partial_t T + \mathbf{u} \cdot \nabla T = \kappa \Delta T, \tag{14}$$

where κ is the thermal diffusivity of the fluid.

Note that SRTLBM can be seen as a special case of MRTLBM associated with a specific choice of parameter values in the equilibria of the moments so that only one single relaxation rate, $1/\tau$, remains free.

2.2 Boundary Conditions

Interactions between the LBM flow field and the interacting objects result from the exchange of momentum at their shared boundaries. Generally, there exist two types of boundaries in the LBM: (1) the surrounding walls of the LBM simulating space; (2) internal objects. The treatment of boundary conditions of both the SRTLBM and MRTLBM are handled in the discrete velocity space after a general streaming simulating step.

For the surrounding walls, the boundary conditions are usually treated as periodic, bounce back (forward), or out flow boundaries [32]. For the inside objects, bounce-back rule can be easily applied. For no-slip boundaries that can move, or involve complex geometries, the bounce back rule has been substantially improved [35, 18] to accommodate curved and moving boundaries.

2.3 Multi-resolution LBM

Physically-based flow modeling methods usually employ a uniform grid to discretize the simulation domain, and then apply numerical computations to solve the NS equations. For large-scale simulations, it is inefficient to maintain a uniform grid with high resolution spanning the entire domain. To achieve interactive performance and to optimize the use of resources, we have applied a multi-resolution LBM [38] that offers high resolution computation around areas of interest (for example, near a solid body) and places low resolution grids on other areas or faraway boundaries. Interfaces between the grids with different resolutions are properly treated to satisfy the continuity of mass and momentum.

This level-of-detail scheme is implemented by a 3D block-based grid structure consisting of a coarse grid and one or more fine grids. The global flow behavior in the whole simulation space is roughly modeled by the LBM simulation on the coarse grid with relatively low consumption of resources. For regions of interest, the LBM computation performs on the fine grids superposed on the coarse one. These grids are implemented as separate blocks instead of tree-style recursive structures. The global simulation on the coarse grid determines the flow properties on interfaces and then defines boundary conditions of the fine grids at each time step. Therefore, the simulation on the fine grids obeys the correct global flow behavior. Meanwhile, it supplies rich visual details and accuracy in the regions of interest by utilizing

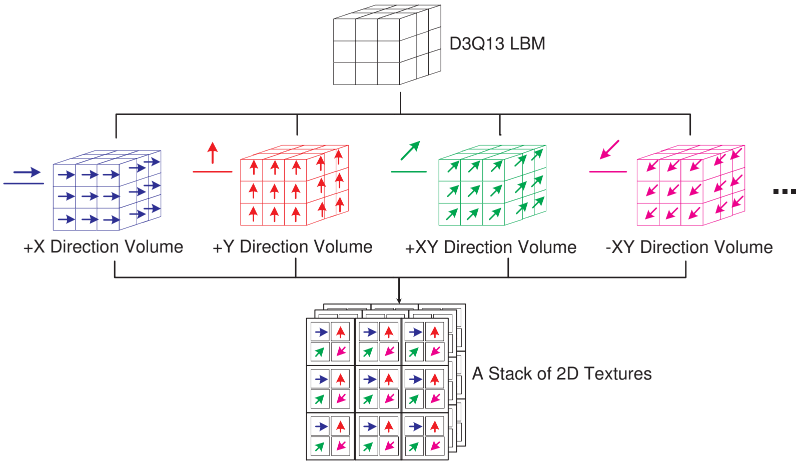


Figure 2 *Each set of particle distributions having the same velocity direction is grouped into a volume and every four volumes are packed into one stack of 2D textures*

small grid spacing, small time intervals, and introducing vorticity confinement. A fine grid is easily initiated and terminated at any time while the global simulation is running. Moreover, a fine grid is able to move along with a moving object, to model small-scale turbulence caused by the object-fluid interactions.

3 GPU Acceleration

As a direct result of recent advances in modern GPU's multiple-pipeline SIMD architecture, stream processing model, and high memory bandwidth, the computational power of the GPU has outpaced that of the CPU. This gap between their computational powers is foreseen to widen, propelled by the booming game industry. As a result, using the GPUs for general-purpose computation has become very attractive. We refer the reader to a further survey on General Purpose computation on the GPU (GPGPU) [22]. However, note that not all computations can take advantage of the GPU computation. Some requirements, such as data parallelism and the locality in memory access, are essential. Therefore, the implicit fluid solvers cannot be naturally plugged into the GPU due to its obligation to solve a linear system from the Poisson equation of the pressure.

An attractive feature of the LBM is that the computation is inherently local and explicitly parallel. This feature allows us to accelerate the LBM computations on the low-cost SIMD processor of contemporary GPUs, and achieve great performance for the flow phenomena simulation. We have accelerated the flow computation on a GPU [16, 17] with moving boundaries, and extended it for large-scale simulations on a GPU cluster [6].

As shown in Figure 2, all the particle distributions f_i associated with the same velocity direction are grouped into a volume and every four volumes are packed into one stack of textures, since a texel has four RGBA components. The equilibrium distributions f_i^{eq} are stored in the same way. The macroscopic density ρ along with the three components of the velocity u, are stored in one stack of textures similarly. Generally, the simulation of the flow field on the GPU is implemented in several steps: (1) Compute the equilibrium distribution values; (2) Compute collision; (3) Apply the boundary conditions at the boundary links; (4) Stream the particle distributions; (5) Compute the macroscopic density and velocity.

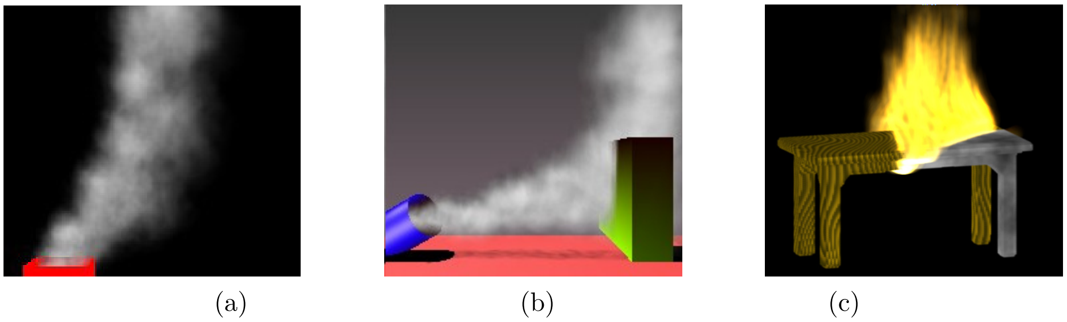


Figure 3 (a) Smoke coming out of a chimney; (b) Smoke interacting with a green moving obstacle. (c) Fire spreading on a table.

Involving only local operations, each step is implemented as a fragment program that calculates the corresponding equation. The fragment programs fetch the input variables from the textures, execute the computation, and render the output results. Comparing the SRTLBM and the MRTLBM, the hardware acceleration is similar, except that for the MRTLBM, extra matrix multiplications are executed. The matrices used are constant and no inversion is required, and therefore, the computation fits well on the GPU. The speedup factor of the GPU acceleration compared with the CPU version depends on the type of the GPU, the optimization of the code, and the resolution of the simulation lattice.

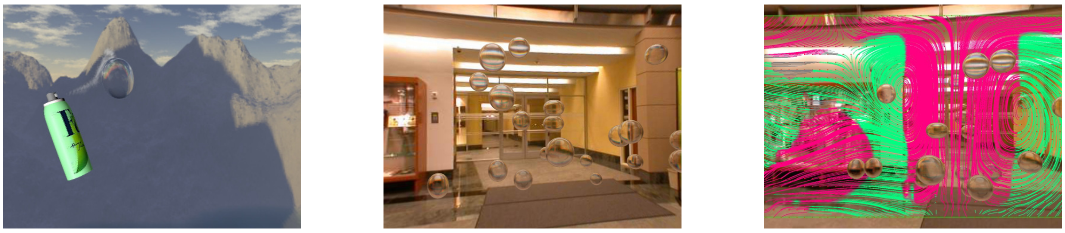
4 Amorphous Phenomena

Natural amorphous phenomena play an important role in graphics and visualization simulations. A good flow model for these phenomena should not only describe the flow, but also model the interaction between the flow and the surrounding environment in a physically correct manner. We have applied the LBM to simulate various flow phenomena, including smoke [36], fire [33, 40], light objects floating in the air [34, 35], solid melting [39], and thermal flow phenomena [37], such as heat shimmering and mirage.

4.1 Smoke and Fire

The movements of smoke and fire are highly complex and have many rotational and turbulent details at various levels. While the LBM simulation models the flow field and takes care of the large-scale interactions of the flow with the scene, our rendering approach can add the small-scale interactions and visual details on-the-fly during the interactive viewing process. A key component of our approach are textured splats [33], which can be efficiently rendered on any commodity hardware board. Textured splats allow us to model both the visual detail of the natural phenomena itself as well as the volumetric shadows cast onto objects in the scene.

Besides the texture splatting method, we have also implemented the volume rendering on the GPU for flow visualization. When a fluid source (for example, a smoke inlet) is positioned and begins to release smoke, the smoke density constructs a scalar volumetric dataset. The evolution of this density volume is modeled by an advection-diffusion equation and computed by a back-tracing algorithm based on the method of characteristics [30]. We use the monotonic cubic interpolation [8] for computing the back-tracing density values at positions not on the regular grid sites. Figure 3 shows the rendering results of fire and smoke.



(a) (b) (c)

Figure 4 (a) Soap bubble deformation and dynamics in response to the spray from the spray can; (b) Multiple bubble blown by a flow; (c) Streamlines show the flow pattern originating from a plane cutting through the most active flow region.

4.2 Floating Objects

Using the LBM with boundary conditions appropriate for moving objects, we have simulated the natural dynamics that emerge from the interaction between a flow field and immersed floating objects [34, 35]. If the boundary is fixed in space and located between two neighboring lattice sites, the no-slip condition is implemented with the bounce-back rule of Section 2.2. To match the velocity at the boundary, in the case of a moving object, the bounced-back particle distribution must also be adjusted to account for the momentum imparted by the boundary. Following the approach of Ladd [14], the essential idea is to enforce the no-slip boundary condition at a moving boundary, while maintaining the conservation of mass and momentum. Based on an SRTLBM simulation, the effect of the boundary velocity is to transfer some momentum across the boundary so that the streamed distribution is:

$$f_i{}'(\mathbf{r},t+1) = f_i(\mathbf{r},t^+) - 2W\frac{3}{c^2}\rho(\mathbf{e_i}\cdot\mathbf{u_b}) \tag{15}$$

where the prime indicates the velocity distribution associated with $-\mathbf{e}_i$, $\mathbf{u_b}$ is the object velocity, and W is a constant associated with the lattice velocity.

We demonstrate our approach using soap bubbles. The soap bubbles in Figure 4 illustrate Fresnel reflection, reveal the dynamics of the unseen flow field in which they travel, and display spherical harmonics in their undulations. Our bubble simulation allows the user to directly interact with the flow field to influence the dynamics in real time.

4.3 Thermal Flow

Various realistic phenomena involve hot objects, dynamic flows and heat transfers, such as melting, dissolving, shimmering and mirage, which are of great interest to computer graphicists and visualization experts. For simulating these phenomena, it is imperative to provide a correct and efficient modeling of the heat transfer as well as the interaction between the objects and the flow. Based on the LBM, we developed a physically-based method that provides a basic framework for modeling these thermal flow phenomena [39, 37].

Our method includes conduction, convection and radiation, which are the three basic types of heat transfer in the real world. Heat sources are defined as any arbitrarily shaped objects interacting with the surrounding air. The temperature distribution on the objects can be calculated from radiators (e.g., the sun), or defined by the user with other physical or nonphysical methods. Such temperature distribution is applied to the surface geometry by a novel mechanism, a *temperature texture*. Therefore, we model the heat transfer from the heat sources to the ambient flow. The different heat exchange behaviors are determined

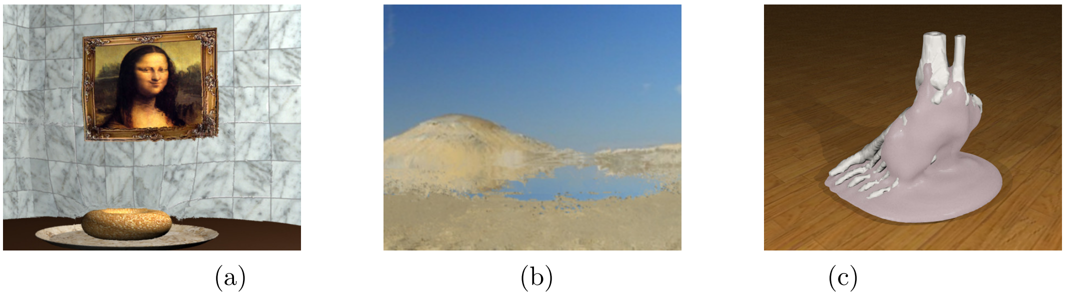


Figure 5 (a) Heat shimmering from a hot bagel; (b) Mirage in a desert; (c) Melting of a volumetric wax foot.

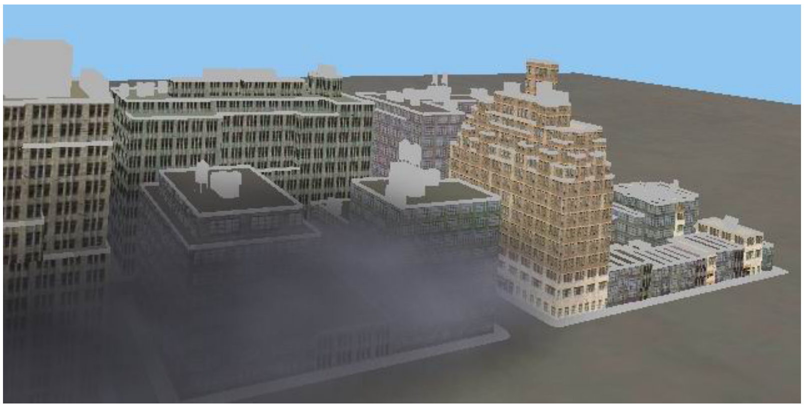


Figure 6 Contaminant propagation in the West Village of New York City.

by material and flow properties that are controlled by physically meaningful parameters, such as thermal conductivity, Prandtl number, and flow velocity. In the flow region, a hybrid thermal LBM, which couples the MRTLBM with a finite difference discretization of a diffusion-advection equation for temperature, is used for modeling the thermal flow dynamics.

Heat shimmering and mirage appear when the heated air has a different refractive index from that of the cooler surrounding air, resulting in an altered light direction through the hot air compared to that of the cooler air. That is, the changes in the index of refraction are attributed to temperature variation. Once the dynamic temperature distribution is computed by our physically-based modeling framework, we apply a nonlinear ray tracing method to render the resulting visual effects [37]. Figure 5a illustrates the shimmering effects from a hot bagel with a distorted background. In Figure 5b, a phantom body of water appears in the desert due to total reflection. We have also presented a method to simulate the melting and flowing phenomena with different materials in multiple phases [39]. In such a multiphase environment, solid objects are melted because of heating and the melted liquid flows while interacting with the ambient air flow. Figure 5c shows the melting effects of a volumetric foot. When the skin and other soft parts are melted as wax and begin to flow downwards, the bones are revealed.

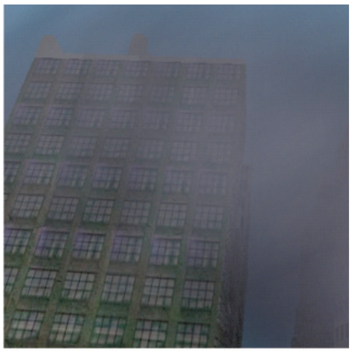


Figure 7 A closeup view of buildings and smoke.

5 Urban Security

The LBM can accurately model air flow and contaminant transport and mixing in geometrically complex environments, such as urban canyons, with the inclusion of thermal effects due to surface heating. Our simulation work [24] is directly relevant to: (1) dispersion models that predict the path and spread of the hazardous agent; (2) interaction with emergency responders who use the information provided by the models. By exploiting the inherent parallelizability of the LBM and implementing the computation on the GPU or a GPU cluster, it is possible to build large scale simulations that span a whole city.

Traditionally, the airborne dispersion of toxic contaminants in open environments is modeled via mesoscale Gaussian plume models which completely disregard the small-scale complex flow dynamics around buildings defining the intricate boundary conditions in deep urban canyons. This may cause the local contamination to be vastly misjudged with devastating consequences for evacuation and mediation efforts. In contrast, our approach uses the MRTLBM to accurately model air flow, contaminant transport and temperature fields within a complex GIS city geometry with a resolution sufficient to accurately simulate its dispersion along the canyons.

To provide an efficient visual interface of the simulation, we first render building facades with textures acquired from real photographs. Because the simulation is executed on the GPU and most of the texture memory is used to store simulation data, there is little space to store textures for buildings. We use noise textures and a smart shader to help texturing the buildings [24]. Then, we render contaminant smoke with self-shadows in real-time with textured splatting method.

To study the behavior of smoke particles, gases, aerosols, and other plumes in a pervasive urban environment, we initially used a 3×3 block area around the Environmental Measurements Laboratory (EML) building in the West Village of New York City. Those sensors record meteorological data (e.g., the wind velocity, temperature, etc.) at a real-time speed. Currently, there are 3 sensors installed in this exercise. Once the live-sensor input is communicated over network links, we adapt the simulation numerical models to accommodate it. The effect of the sensor data is incorporated as a body force [24]. Figure 6 shows the simulation result of a 10-block area rendered by our visualization system. Figure 7 shows the closeup views of the buildings and smoke during the simulation. The LBM model consists of $90 \times 30 \times 60$ lattice sites with the spacing between two neighboring sites less than 5 meters. The building GIS models are at 1 meter resolution. Measured on a computer with a 2.53 GHz Intel Pentium 4 CPU and an NVIDIA GeForce FX 5950 Ultra GPU, our GPU

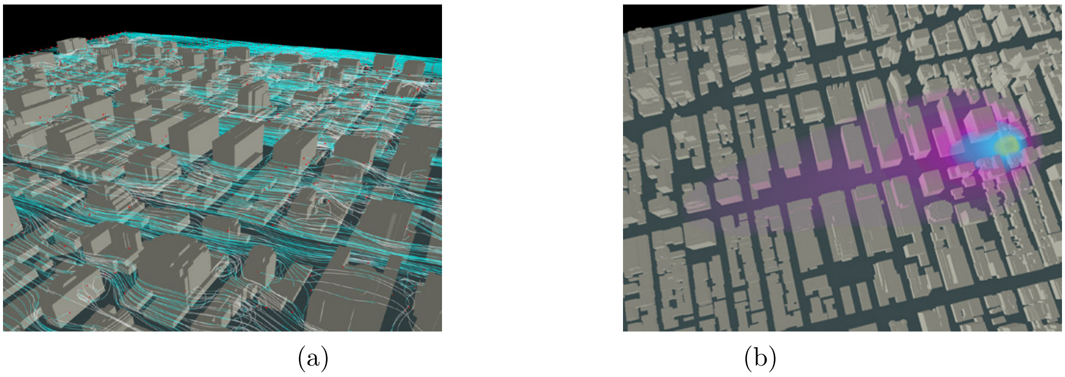


Figure 8 Airborne dispersion in the Time Square area. (a) Wind field streamlines; (b) Contaminant density rendered with colors.

implementation has achieved speedup factor of 8 over a CPU imiplementation and can be run at over 12 steps per second.

For large-scale simulations, we have developed a parallel LBM flow simulation on a GPU cluster and simulated the dispersion of airborne contaminants in the Times Square area of New York City [6]. Using 30 GPU nodes, our simulation can compute a 480 × 400 × 80 LBM (the spacing between two lattice sites is 3.8 meters) in 0.31 second/step, a speed which is 4.6 times faster than that of our CPU cluster implementation on 30 CPU nodes. Figure 8 shows the contaminant dispersion in the Time Square area with wind field streamlines and airborne dispersion distribution.

6 Conclusions

We have proposed a solution for the visual simulation of flow phenomena based on a promising microscopic fluid solver, the Lattice Boltzmann Method. The LBM is powerful and flexible due to its great ability for handling complex geometries and accelerating the computation on parallel machines. Our visual simulation system has great benefits in modeling the complex and non-linear flow behaviors with its convincing visual results and accurate prediction simulation. This approach will enable the development of other superior prediction simulation in computer graphics and games, as well as in computational science and engineering.

Acknowledgement

This work has been partially supported by NSF grant CCR-0306438 and from Department of Homeland Security, Environment Measurement Laboratory.

References

1 N. Chu and C. Tai. Moxi: real-time ink dispersion in absorbent paper. *ACM Trans. Graph.*, 24(3):504–511, 2005.

2 D. d'Humiéres, I. Ginzburg, M. Krafczyk, P. Lallemand, and L. Luo. Multiple-relaxation-time lattice Boltzmann models in three-dimensions. *Philosophical Transactions of Royal Society of London*, 360(1792):437–451, 2002.

3 D. Ebert, F. Musgrave, D Peachey, K. Perlin, and S. Worley. *Texturing and Modeling, A Procedural Approach.* Morgan Kaufmann Press, third edition edition, 2002.

4 D. Ebert and R. Parent. Rendering and animation of gaseous phenomena by combining fast volume and scanline a-buffer techniques. *Proceedings of SIGGRAPH*, pages 357–366, 1990.

5 D. Enright, S. Marschner, and R. Fedkiw. Animation and rendering of complex water surfaces. *Proceedings of SIGGRAPH*, pages 736–744, 2002.

6 Z. Fan, F. Qiu, A. Kaufman, and S. Yoakum-Stover. GPU cluster for high performance computing. *Proceedings of ACM/IEEE Supercomputing Conference*, 2004.

7 Z. Fan, Y. Zhao, A. Kaufman, and Y. He. Adapted unstructured lbm for flow simulation on curved surfaces. *Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 245–254, 2005.

8 R. Fedkiw, J. Stam, and H. Jensen. Visual simulation of smoke. *Proceedings of SIGGRAPH*, pages 15–22, 2001.

9 B. Feldman, J. O'Brien, and O. Arikan. Animating suspended particle explosions. *ACM Trans. Graph.*, 22(3):708–715, 2003.

10 N. Foster and D. Metaxas. Realistic animation of liquids. *Graphical Models and Image Processing*, 58(5):471–483, 1996.

11 N. Foster and D. Metaxas. Modeling the motion of a hot, turbulent gas. *Proceedings of SIGGRAPH*, pages 181–188, 1997.

12 T. Goktekin, A. Bargteil, and J. O'Brien. A method for animating viscoelastic fluids. *ACM Trans. Graph.*, 23(3):463–468, 2004.

13 S. A. King, R. A. Crawfis, and W. Reid. Fast volume rendering and animation of amorphous phenomena. *Proceedings of Volume Graphics*, pages 229–242, 2000.

14 A. J. C. Ladd. Numerical simulations of particulate suspensions via a discretized Boltzmann equation. Part I. Theoretical foundation. *J. Fluid Mech.*, 271:285–309, 1994.

15 P. Lallemand and L. Luo. Theory of the lattice Boltzmann method: Acoustic and thermal properties in two and three dimensions. *Physical Review E*, 68:036706, 2003.

16 W. Li, Z. Fan, X. Wei, and A. Kaufman. *Flow Simulation with Complex Boundaries*, chapter 47, pages 747–764. Addison-Wesley, 2005.

17 W. Li, X. Wei, and A. Kaufman. Implementing lattice Boltzmann computation on graphics hardware. *The Visual Computer*, 19(7-8):444–456, December 2003.

18 R. Mei, L. S. Luo, and W. Shyy. An accurate curved boundary treatment in the lattice Boltzmann method. *Journal of Comp. Phys.*, 155:307–330, June 1999.

19 M. Mueller, D. Charypar, and M. Gross. Particle-based fluid simulation for interactive applications. *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 154–159, 2003.

20 M. Nelson and R. Crawfis. Visualizing wind velocities by advecting cloud textures. *Proceedings of Visualization*, pages 179–184, 1992.

21 D. Nguyen, R. Fedkiw, and H. Jensen. Physically based modeling and animation of fire. *Proceeding of SIGGRAPH 2002*, pages 721–728, 2002.

22 John D. Owens, David Luebke, Naga Govindaraju, Mark Harris, Jens Krüger, Aaron E. Lefohn, and Timothy J. Purcell. A survey of general-purpose computation on graphics hardware. *Proceedings of Eurographics*, pages 21–51, 2005.

23 S. Premoze, T. Tasdizen, J. Bigler, A. Lefohn, and R.Whitaker. Particle-based simulation of fluids. *Proceeding of Eurographics*, pages 401–410, 2003.

24 F. Qiu, Y. Zhao, Z. Fan, X. Wei, H. Lorenz, J. Wang, S. Yoakum-Stover, A. Kaufman, and K. Mueller. Dispersion simulation and visualization for urban security. *Proceedings of Visualization*, pages 553–560, October 2004.

25 N. Rasmussen, D. Enright, D. Nguyen, S. Marino, N. Sumner, W. Geiger, S. Hoon, and R. Fedkiw. Directable photorealistic liquids. *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer animation*, pages 193–202, 2004.

26 W. T. Reeves. Particle system - a technique for modeling a class of fuzzy objects. *Proceedings of SIGGRAPH*, 17(3):359–376, 1983.

27 G. Sakas. Modeling and animating turbulent gaseous phenomena using spectral synthesis. *The Visual Computer*, pages 200–212, 1993.

28 A. Selle, N. Rasmussen, and R. Fedkiw. A vortex particle method for smoke, water and explosions. *ACM Trans. Graph.*, 24(3):910–914, 2005.

29 M. Shinya and A. Fournier. Stochastic motion - motion under the influence of wind. *Computer Graphics Forum*, 11(3):C119–128, September 1992.

30 J. Stam. Stable fluids. *Proceedings of SIGGRAPH*, pages 121–128, 1999.

31 J. Stam. Flows on surfaces of arbitrary topology. *ACM Trans. Graph.*, 22(3):724–731, 2003.

32 S. Succi. *The lattice Boltzmann equation for fluid dynamics and beyond.* Numerical Mathematics and Scientific Computation. Oxford University Press, 2001.

33 X. Wei, W. Li, K. Mueller, and A. Kaufman. Simulating fire with texture splats. *Proceedings of Visualization*, pages 227–237, 2002.

34 X. Wei, Y. Zhao, Z. Fan, W. Li, F. Qiu, S. Yoakum-Stover, and A. Kaufman. Lattice-based flow field modeling. *IEEE Transactions on Visualization and Computer Graphics*, 10(6):719–729, November 2004.

35 X. Wei, Y. Zhao, Z. Fan, W. Li, S. Yoakum-Stover, and A. Kaufman. Blowing in the wind. *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 75–85, July 2003.

36 Xiaoming Wei, Wei Li, Klaus Mueller, and Arie E. Kaufman. The lattice-boltzmann method for simulating gaseous phenomena. *IEEE Trans. on Visualization and Computer Graphics*, 10(3):164–176, March/April 2004.

37 Y. Zhao, Y. Han, Z. Fan, F. Qiu, Y. Kuo, A. Kaufman, and K. Mueller. Visual simulation of heat shimmering and mirage. *Stony Brook CVC Technical Report No. 20051219, Submitted*, 2005.

38 Y. Zhao, F. Qiu, Z. Fan, and A. Kaufman. Interactive fluid simulation with multi-resolution LBM. *Stony Brook CVC Technical Report No. 20060206, Submitted*, 2006.

39 Y. Zhao, L. Wang, F. Qiu, A. Kaufman, and K. Mueller. Melting and flowing in multiphase environment. *Computers & Graphics, Special Issue on Natural Phenomena*, 2006.

40 Y. Zhao, X. Wei, Z. Fan, A. Kaufman, and H. Qin. Voxels on fire. *Proceedings of Visualization*, pages 271–278, October 2003.