GPU-based Visual Simulation of Dispersion in Urban Environments

Feng Qiu[1] * Zhe Fan[1] Ye Zhao[1] Haik Lorenz[2] † Jin Zhou[1] ‡ Arie Kaufman[1]
Klaus Mueller[1]

[1]Stony Brook University, USA [2]University of Potsdam, Germany

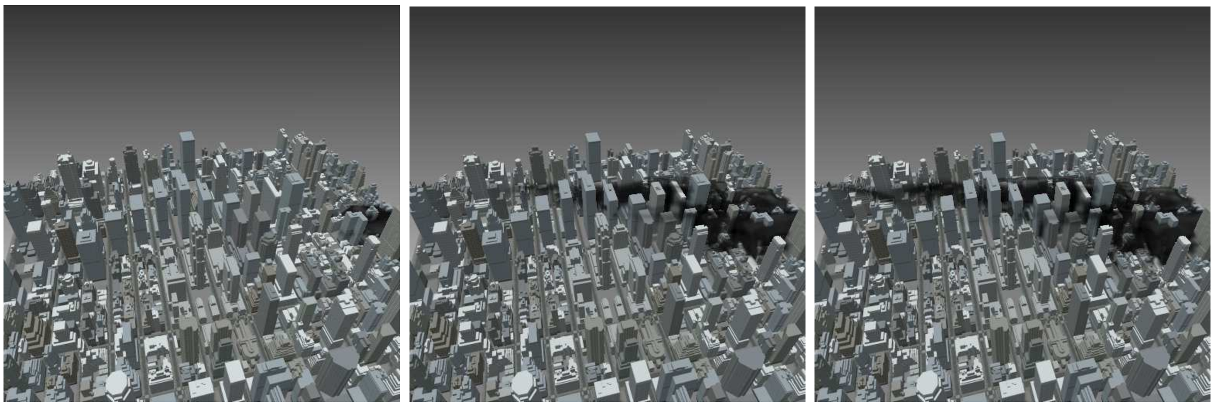

Figure 1: Smoke dispersion simulated in the Times Square area of New York City. The wind is blowing from the right to the left.

Expeditious response to airborne hazardous releases in urban environments requires the availability of the dispersion distribution. We have developed an interactive system of simulating and visualizing dispersion contaminants in dense urban environments on a GPU and GPU cluster.

In practice, the airborne releases are moved by the air flows and propagate in the environment. The flow field of the atmosphere high above the buildings is roughly described by the local meteorological conditions. However, the very coarse meteorological data is hard to be used for estimating the dispersion propagation, because of its low resolution and the complicated hypsography in urban environments. Therefore, we have developed our system based on a computational fluid dynamics (CFD) model - the Lattice Boltzmann Method (LBM). The LBM solves the fluid dynamics by modeling the microscopic behavior of fluid particles, while the fluid dynamics obeys the desired Navier-Stokes (NS) equations. It provides many advantages, including easy handling of complex and dynamic objects and boundaries inside the flow field and fully parallel computations. Therefore, it is ideal for simulating dispersion in urban environments and accelerating it on a modern graphics processing unit (GPU) to achieve great simulation performance. In this work, we have implemented the complicated multi-resolution LBM simulation on a GPU cluster.

For large-scale simulations in urban security applications, it is inefficient to maintain a uniform grid with high resolution spanning the entire simulation domain. Here, we use a multi-resolution LBM to save computational resources and accelerate the simulation. This level-of-detail scheme is implemented by discretizing the computational domain into one coarse grid and one or more fine grids. The air flow in the whole space is modeled by the LBM simulation on the coarse grid with fewer computational resources. In the critical area around the buildings, the air flow is simulated on the fine grids superposed on the coarse one. Boundary conditions of the fine grids are interpolated from the coarse grid computation on their interfaces at each time step. Therefore, the simulation on the fine grids is steered by the large-scale wind properties. Meanwhile, it supplies abundant details and accuracy in the critical regions.

When we partition and distribute the simulation on the cluster nodes, there are two kinds of data communications. The first is the data transfer between the two LBM grids, due to the fact that at every time-step the fine grid obtains its boundary data by transforming the coarse grid information at the same location. The second is the data communication between the different portions of the grid that is partitioned across the cluster nodes. To keep the first kind of communication efficient and simple, we partition the simulation domain in physical space. For the second kind of communication, which is within the same grid and among different portions residing on different cluster nodes, it is inevitable that appropriate data at partition boundaries needs to be read from the GPU memory and sent through the network to neighboring nodes.

Our implementation is based on the ZippyGPU toolkit, which we have developed on top of OpenGL, Cg, and Message Passing Interface (MPI) to help programming of general purpose computation on GPU clusters. The LBM particle distribution functions, densities and velocities, etc. are in 3D arrays, and all simulation grid operations are written in Cg programs.

The density volumes generated in the simulation are distributed on the cluster nodes. Transferring volume data to a single node for rendering is inefficient and unnecessary. Because the simulation is implemented on the GPUs, the volume is rendered on the same GPUs to avoid transferring large amounts of data between the GPU and main memory.

We demonstrate our simulation and visualization system with the Times Square area in New York City, from 8th Avenue to Park Avenue and from 42nd Street to 60th Street. This region is approximately $1.46km \times 1.19km$ and has 75 blocks and more than 800 buildings. The GPU cluster is composed of 16 simulation/rendering nodes, 15 compositing nodes and 1 master node for user interface. The finer grid of the simulation is arranged to tightly enclose the Times Square area and its resolution is $320 \times 100 \times 280$ at the grid spacing of 4.25 meters. The spacing of the coarse grid is twice that of the finer grid and the resolution is $180 \times 100 \times 160$. The simulation takes 242 ms for each step.

*e-mail: qfeng,fzhe,yezhao@cs.sunysb.edu
†e-mail: haik.lorenz@hpi.uni-potsdam.de
‡e-mail: jzhou,ari,mueller@cs.sunysb.edu