EGVE Symposium (2008)
B. Mohler and R. van Liere (Editors)

Integrating Particle Dispersion Models into Real-time Virtual Environments

P. Willemsen[†1], A. Norgren[‡1], B. Singh[§2], and E.R. Pardyjak[¶2]

[1]Department of Computer Science, University of Minnesota Duluth, USA
[2]Department of Mechanical Engineering, University of Utah, USA

Abstract

We present a system for interacting with fast response particle dispersion models in a real-time virtual environment. The particle simulation system is used as both an engineering tool for modeling and understanding turbulence in flow fields and as an integral part of an atmospheric, wind display for real-time virtual environments. The focus of this paper is on the use of virtual environment visualization and interaction as a means for providing deeper insight about the underlying turbulence and flow field models in the dispersion simulations. The use of immersive virtual environment displays, as opposed to standard computer display screens, allows engineers to acquire novel view points and interact with the data in novel ways. We present our current visualization and interaction methods. Our goal is to develop a tool for interacting with, controlling, and manipulating the model parameters of fast response particle dispersion systems while immersed within the simulation environment.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Virtual Reality

1. Introduction

This paper presents a virtual environment framework for modeling, simulating, and interacting with fast response, particle dispersion and wind simulations. The term *fast response* describes systems that are designed to quickly provide information about changing conditions, such as where airborne particles are being dispersed in an urban setting. The information obtained from these systems can be used for emergency response training and preparedness, urban planning, and education about environmental concerns. However, current fast response systems for particle dispersion and flow field modeling have not been fast enough to be run at interactive rates. Data is often visualized and analyzed in a post-process phase of the simulation. As such, live interaction with the simulation components is rarely possible and any knowledge created for training or education purposes takes time to gather. Moreover, computational fluids engineers working to develop the underlying equations that govern the mechanics of the wind flow field and particle dispersion in the urban setting do not receive immediate feedback on the model's characteristics.

The objective of this research is to solve these problems by developing a fast response dispersion system that runs at real-time rates, allowing the dispersion simulation data to be visualized and interacted with as the simulation progresses. With this solution, the fast response system transforms from a simulate and analyze approach to one in which simulation and analysis happen simultaneously, making rapid prototyping of the underlying dispersion framework possible. Additionally, emergency responders could use the system to optimize response strategies by predicting the outcome of live situations. Such a system has the potential to assist engineers, urban planners, emergency responders, and students to better understand and refine dispersion models relating to environmental processes in urban settings. More specifically, our current efforts are aimed at creating interaction and visualization techniques that have potential to more effectively provide insight about the complex processes and interactions

---

[†] willemsn@d.umn.edu
[‡] norg0039@d.umn.edu
[§] balwindersingh@gmail.com
[¶] pardyjak@eng.utah.edu

that arise between airborne particles (such as pollutants) and the urban environment.

In summary, our approach implements the particle dispersion simulation on current graphics hardware to achieve real-time simulation and interaction. Visual rendering of the particles extends information to the viewer about flow direction around urban structures. Users can also walk within the simulation to seek out viewpoints that allow for more specific investigation of the resulting dispersion or how well the underlying equations work in specific circumstances. For instance, an engineer may want to know whether particles became trapped between two specific buildings given new equations. We track a user's hand position and orientation as a natural means to precisely investigate the flow of particles, and hence the underlying dispersion model. The visualization and interaction mechanisms work together to provide benefits to the users of this system that do not exist by viewing the simulation on a monitor screen. To our knowledge, this is the only fast response dispersion modeling work executing within a virtual environment setup and utilizing the parallelism afforded by current graphics hardware.

1.1. Motivation

Our particle dispersion simulation is being developed as part of an atmospheric, wind display project for real-time virtual environments. The wind display system, the Treadport Active Wind Tunnel (TPAWT, pronounced *teapot*) is being developed at the University of Utah to generate controlled wind around a user walking on a large treadmill, surrounded by three screens filling 180 degrees of view [HGP05, KDPM06, KMDP07]. The fast response particle dispersion system described in this paper will integrate with the TPAWT's mechanical system that controls the wind speed and direction, providing inputs about the local wind field and dispersion concentrations local to the virtual environment user. Our system will also be used to visualize and interact with the environmental, atmospheric processes.

The objective of the TPAWT is to generate a physical, tactile wind, directed at a user walking on the treadmill. Users will feel the wind within the environment on their bodies and relate the visual aspects of the wind (leaves, flags, etc...) to their own actions within the virtual environment. The combined physical and visual effects may be important. One example that illustrates the potential for interaction between visual and tactile information in the TPAWT is an urban environment simulation in which the user feels the wind produced in the wake of an autonomous bus driving by them as the user stands on a virtual curb in the environment. Modeling and simulating wind fields within a real-time interactive virtual environment has potential for increasing immersion on several levels. From a visual perspective, accessing simulated wind field data can assist in creating a richer dynamic visual experience with leaves blowing through the environment, trees reacting to the wind, or autonomous pedestrians leaning into strong wind gusts as they walk away from a building. Such visual elements work to create a more dynamic, ambient environment consistent with real environments.

While our work is also motivated by being integrated with the TPAWT virtual environment, our particle dispersion system is emerging as an important engineering simulation tool for modeling, interacting with, and understanding wind turbulence models and plume dispersion using virtual environment technologies. The focus of this paper is on the use of the virtual environment as an engineering modeling tool to increase knowledge for how environmental processes and urban form interact.

2. Related Work

Particle systems have been used for a variety of rendering techniques with relation to computer graphics and visualization, representing effects such as fire, smoke, water, or other objects [Ree83]. More recently, graphics hardware-based particle systems have been used, vastly increasing the numbers of particles and computations associated with each particle in interactive environments [KSW04, KKKW05]. Our dispersion system also utilizes current graphics hardware to accelerate computations.

Visualization of flow fields and particle path traces have also received much research attention. These datasets are often visualized using arrows, streamlines, stream tubes, point sprites, oriented splats, image-based texture approaches [LBS03], stream ribbons, or illuminated stream lines [ZSH96] ([KKKW05] provides a good overview of many of these techniques). Our current implementation utilizes basic path line tracing and point sprites for rendering the particle system. Our visualization would benefit from incorporating many of the possible visualization schemes. However, our efforts have primarily been to develop an interactive and immersive computational engineering tool for fast response dispersion modeling.

The current system utilizes tracking of head and hand positions and orientations to provide interaction with the particle simulation data. We plan to incorporate hand gesture-based techniques to allow for more natural user control of scaling in the simulation space, such as the work done on hands-free scaling [LFKZ01] and multi-scale interaction within virtual environments [KNBP06]. As the dispersion tool progresses, we also expect to incorporate selection techniques [Ste06] and manipulation of objects [MJS97] within the urban environment for controlling dispersion model parameters, urban structure placement, size, and form.

3. GPU-based Fast Response Simulation

Our virtual environment plume dispersion and wind field simulation software is derived from QUIC-Plume (Quick

Urban and Industrial Complex) and QUIC-URB. QUIC-Plume is a fast response transport and dispersion code that models turbulent dispersion of particles in urban settings [WBB*04]. QUIC-URB computes a mass consistent, 3D wind field around buildings and other structures present in urban environments [PB01]. Both packages are considered to be *fast response* models, capable of supplying information about changing wind fields and particle plume concentrations for emergency response applications or urban planning scenarios. While significantly faster than traditional computational fluid dynamics models, neither of these systems are able to run at rates conducive to real-time virtual environments using standard particle release rates and urban domain sizes. It should be noted that these systems do not solve the full Navier-Stokes equations, which describe the motion of gaseous or liquid substances. The Navier-Stokes equations can be very difficult to solve numerically.

To facilitate integration with the TPAWT, we have developed a particle dispersion code akin to QUIC-Plume that executes entirely on graphics hardware, taking advantage of the data parallelism in modern graphics processing units (GPUs). Raw performance of GPUs has surpassed performance of CPUs for certain applications, and as a result, GPUs have become useful computational tools. Examples are numerous in the literature as GPUs provide inexpensive highly parallel execution to accelerate a wide range of scientific and simulation applications [SCC05, HBSL03]. Particle simulations do benefit from the parallelism inherent on GPUs, which have been shown to be quite effective for simulating large particle simulations [KSW04, QZF*04]. In particular, work by Krüger et al. use a GPU-based solution to provide interactive visualization of 3D flows [KKKW05].

Our implementation uses a Lagrangian dispersion model that moves (or *advects*) particles during each simulation step with the mean wind field and a turbulent perturbation. We acquire the mean wind field for an urban environment using QUIC-URB. However, work is underway to calculate the changing wind field on the GPU, which will remove the reliance on QUIC-URB. Our dispersion model represents an unsteady random-walk model that factors in building locations and dimensions. Our methodology is an application of the Langevin equations given by Williams et al. [WBP04] and shown in the equations below (following [Rod96]) where the flow has been decomposed into mean and fluctuating velocity quantities (i.e., $u_i = U_i + u'_i$). Our advection step modifies the position of a particle, x_i, based on the particle's previous position, $x_{p,i}$ plus the velocity fluctuations at the current and previous time (u'_i and $u'_{p,i}$) and the mean velocity at the current time.

$$x_i = x_{p,i} + U_i \triangle t + \frac{u'_{p,i} + u'_i}{2} \triangle t \tag{1}$$

$$u'_i = u'_{p,i} + du'_i \tag{2}$$

$$\begin{aligned} du'_1 = {} & -(\frac{C_0\varepsilon}{2})(\lambda_{11}u'_{p,1} + \lambda_{13}u'_{p,3})dt + \\ & \left(\frac{\partial U}{\partial z}u'_{p,3} + \frac{1}{2}\frac{\partial \tau_{13}}{\partial z} + \frac{u'_{p,3}}{2}(\frac{\partial \tau_{11}}{\partial z}(\lambda_{11}u'_{p,1} + \lambda_{13}u'_{p,3}) + \right. \\ & \left. \frac{\partial \tau_{13}}{\partial z}(\lambda_{13}u'_{p,1} + \lambda_{33}u'_{p,3}))\right)dt + (C_0\varepsilon dt)^{\frac{1}{2}} d\xi_1(t) \end{aligned} \tag{3}$$

$$\begin{aligned} du'_2 = {} & -\frac{C_0\varepsilon}{2}(\lambda_{22}u'_{p,2})dt + (\frac{1}{2}\frac{\partial \tau_{22}}{\partial z}\lambda_{22}u'_{p,2}u'_{p,3})dt + \\ & (C_0\varepsilon dt)^{\frac{1}{2}} d\xi_2(t) \end{aligned} \tag{4}$$

$$\begin{aligned} du'_3 = {} & -(\frac{C_0\varepsilon}{2})(\lambda_{13}u'_{p,1} + \lambda_{33}u'_{p,3})dt + \\ & \left(\frac{1}{2}\frac{\partial \tau_{33}}{\partial z} + \frac{u'_{p,3}}{2}\frac{\partial \tau_{13}}{\partial z}(\lambda_{11}u'_{p,1} + \lambda_{13}u'_{p,3}) + \right. \\ & \left. \frac{\partial \tau_{33}}{\partial z}(\lambda_{13}u'_{p,1} + \lambda_{33}u'_{p,3})\right)dt + (C_0\varepsilon dt)^{\frac{1}{2}} d\xi_1(t) \end{aligned} \tag{5}$$

In Equations 1-5, u_i represents the instantaneous velocity (m/s), U_i the mean velocity (m/s), C_0 is a constant that scales the turbulent kinetic energy dissipation rate (approximately 5.7), ε is the mean turbulent kinetic energy dissipation rate ($\frac{m^2}{s^2}$), τ_{ij} is the Reynolds shear stress (amount of turbulence at a location) divided by density, λ_{ij} is the matrix inverse of τ_{ij}, and $d\xi_i$ are uncorrelated, normally distributed random variables with means of zero and standard deviations of one.

At each simulation step, Equations 1-5 are evaluated on the GPU to perform particle advection. All data necessary for these computations are preloaded into textures or computed on the GPU during the simulation. As particle positions are updated on the GPU, the reflections off of urban structures such as buildings and the ground are also performed. After the advection and reflection step, particles are rendered to the display.

This model and implementation have been evaluated using a continuous point source release in a uniform flow field and within an urban domain. In these situations, the GPU simulation produced similar results to the original implementation and outperformed the CPU simulations by nearly three orders of magnitude in the uniform flow field condition and by over an order of magnitude in the urbanized domain with buildings. More importantly for the engineers that use these tools is the fact that the GPU code can advect millions of particles at real-time rates [WNSP07, PSNW07], unlike the CPU implementation of QUIC-Plume, providing increased statistical significance to the collected data.

The performance difference in our current approach allows us to utilize our dispersion system at real-time rates, making the virtual environment a convenient simulation tool for more accurate plume dispersion and turbulence models within the fast response framework. Because the particle data already resides on the GPU, we are immediately able to view and interact with the particle dispersion field and the

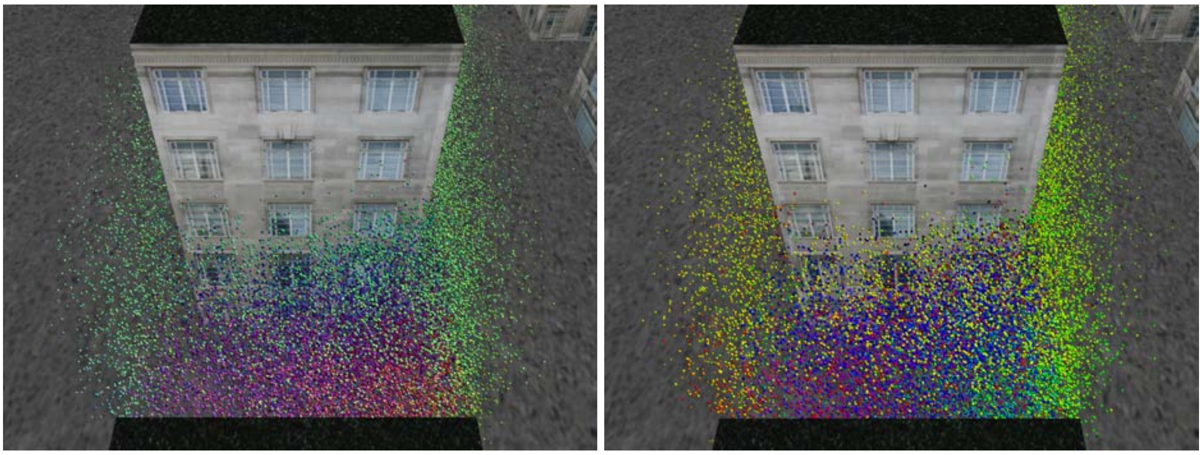

Figure 1: *A simple mapping from the particle motion direction to RGB is shown on the left. Using the opponent color space mapping, oRGB, shown on the right, provides intuitive visual access to direction of particle motion.*

parameters that underly the model. Having live, real-time visual information about the state of the dispersion model combined with the ability to interact with that data has been a huge benefit to the engineers involved in this project. Visual debugging of the model leads to better insight on how to correct the model. Interaction within the domain, such as viewing the turbulence parameters intuitively and emitting particles in key locations also provide useful information. Our motivation for the visualization and interaction components of this work stems from creating a virtual environment that can serve as a laboratory for developing effective models for turbulence, dispersion, and urban wind fields.

4. Visualization Elements

Our main challenge for creating an effective dispersion modeling tool has been in exploring visualization and interaction methods that assist the computational fluid dynamics engineers that work on this project. In particular, these engineers have found that real-time visualization and interaction with their flow models (something they have not been able to do until now) has been extremely beneficial for debugging and understanding their turbulence models. As such, we have focused on novel and unique means to visualize and interact with the components of the flow model that are not common in fast response systems. The engineers have also found that being *in* the data has provided useful vantages. By this, we mean that users (the engineers in this case) are able to position themselves at locations within the urban domain to inspect how the underlying turbulence model operates. For instance, engineers might position their viewpoint to focus on an abutment between a building and the ground plane.

First and foremost, we are simulating particle dispersion in urban environments so by default we show particles advecting in real-time through the domain. Particles are advected on the GPU using the Lagrangian random-walk model that moves particles according to the mean wind field with perturbations that mimic the stochastic turbulence found in urban domains. The general flow and motion produced by large numbers of advecting particles provides tremendous visual effect and information. However, with millions of particles, particle motion can be difficult to interpret at times, especially in the wakes of buildings and through urban street canyons. In these areas, turbulence parameters vary dramatically from the parameters in more open spaces. Moreover, particles reflect off the ground and off structures in the environment, so their motion is complex. Interpretation of motion can become more difficult as visual clutter increases as visual aids are added to the scene to augment the visual flow information (*i.e.* drawing small motion vectors for every particle).

Our goal is to provide simple, yet effective information to users about the motion of the dispersion field. To achieve this, we have explored color mapping particle motion onto each particle sprite as a way to understand the overall dispersion while still being able to pick out individual particle motions when necessary. Our main method for coloring particles uses the instantaneous velocity vector associated with each particle as the basis for the color mapping. One simple mapping uses the particle velocity and normalizes it to be in the unit sphere. This normalized value is then transformed into the RGB cube. Axis aligned motions produce colorings in the center of the RGB cube's planes. The resulting colors are not common primary colors, and as such, tend to be difficult for users to quickly learn. As particle velocity diverges from the axis-aligned directions, immediately differentiating particle motions relative to the main axes can be quite difficult.

Instead, we use a coloring scheme based on opponent

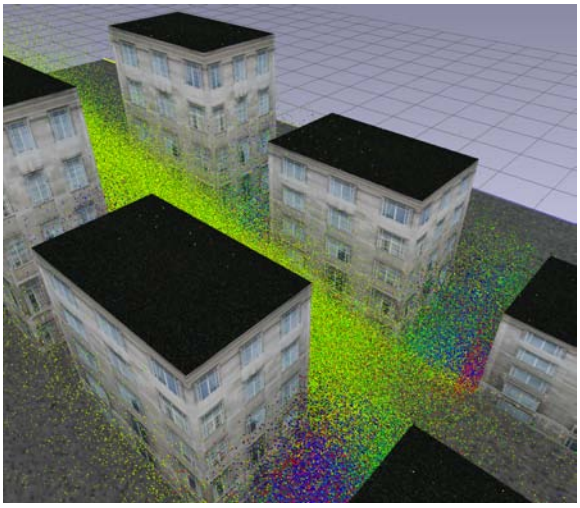

Figure 2: *An image from the dispersion simulation showing the oRGB color mapping on a larger scale.*

color space that produces nearly immediate visual cues for the particle direction. The colors in this scheme map to readily known, named colors, and provide more intuitive correlation with the axis aligned directions of motion (*e.g.*+X, +Y, etc...). This color mapping is derived from the *oRGB* color space developed by Bratkova et. al [BBSar]. Opponent color space is based on Hering's theory of opponent processes for color perception, mapping colors onto three axes: yellow/blue, green/red, and black/white. The *luma* channel (as defined by Bratkova et. al) is the black/white axis, with the yellow/blue and green/red axes representing the chromaticity. For velocity, we map height in Z to the luma channel and set the two chroma channels to the X and Y components of the velocity vector. This will produce a coloring such that motion in the +X direction will be more yellow, motion in -X more blue, motion in +Y more red, motion in -Y more green, motion in +Z lighter, and motion in -Z darker. Figure 1 compares the simple color mapping (left image) to our *oRGB* color mapping (right image) showing particles moving downstream (*i.e.* toward the top of this printed page) along the building being more yellow but influenced by the leftward (reddish) or rightward (greenish) motion of the particles. Blueish particles are swirling in the vortice, opposing the movement of the yellow particles. While one could learn a variety of color mappings for velocity, our experience has been that the simplicity of the *oRGB* mapping, as illustrated by its use of primary, well-known colors, has made visualizing particle motion more intuitive. Figure 2 illustrates a larger scale of the oRGB mapping being used.

5. Interaction with the Simulation

Direct interaction with the simulation elements has been of benefit to the users of this system. In particular, our aim is to leverage the technology available with virtual environments to create natural investigations of simulation data and underlying model parameters. In addition to the TPAWT facility at Utah, our lab in Minnesota consists of a 21ft x 33ft space tracked with a WorldVis PPTx4 position tracking system. Intersense InertiaCubes are used for sensing the orientation of the user's hands and the orientation of an NVIS nVisor SX head-mounted display worn by the user. Users are able to walk around within the dispersion simulation viewing it through their own locomotion and actions. Interactive, immersive environments provide advantages over interactive desktop viewing of the dispersion simulation. First and foremost, the user is within the same spatial framework as the data. Users can walk around and interact with the data. With our current experience, this alone has provided access to viewpoints in the urban domain that are not easy to understand when using mouse and keyboards for control. Second, users can utilize proprioception and spatial updating to interact with the environment. These types of natural control are not available through keyboard and mouse controls.

One of the interactive visualization tools we use is to overlay the turbulence model parameters onto the domain by rendering transparent slices of the turbulence field. When applicable, we utilize our *oRGB* coloring. The turbulence overlay provides the engineers with the ability to simultaneously visualize the components of the turbulence model and the resulting particle behavior. Many of these dispersion model parameters are 3D and fill the simulation domain. In these situations, it makes sense to visualize the data as general slices through the domain. A polygon is created from the intersection of the generalized slice with the dispersion domain. Texture is dynamically applied to the polygon depending on which dispersion parameters are being visualized. The orientation of the slice, in terms of local yaw, pitch, and roll can be controlled through mouse clicks and related motions. However, this control is cumbersome when users view the simulation through either a head-mounted display or through the surrounding screens of the TPAWT.

We simplify control of these data slices by binding the orientation and position of the generalized slice to the user's hand orientation and relative position. Hand position is tracked using the PPTX system, while hand orientation data comes from the Intersense Wireless InertiaCubes. This rather simple connection creates intuitive control over the data and allows the slice to be rotated and translated naturally. The left image of Figure 3 shows a view of the turbulence model data being visually overlaid and then controlled by a user. Viewing of this data has been particularly important for quickly locating errors in flow models around buildings.

Another important investigation tool for understanding

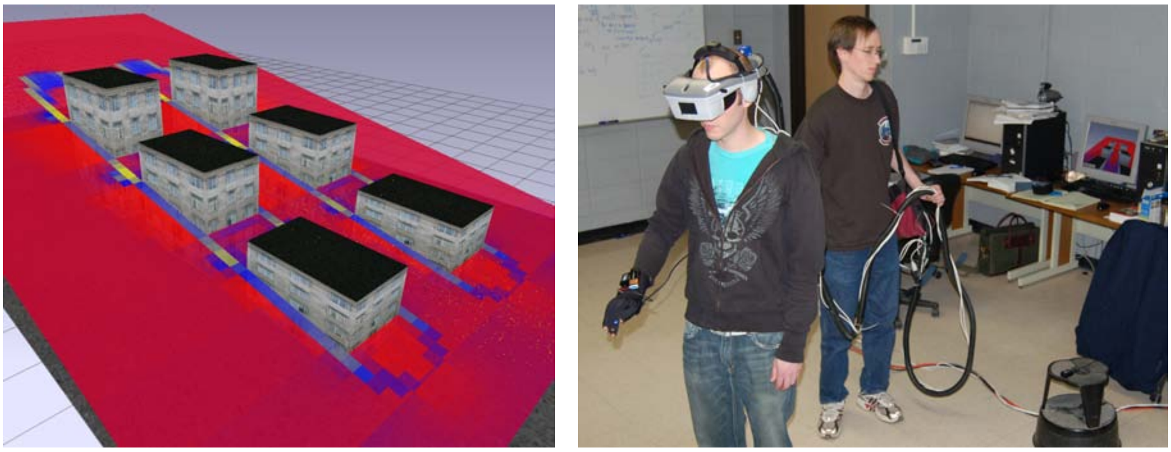

Figure 3: *Visualization of 3D data slices and streak line particle emitting are controlled by user hand orientation and position. In the these images, 3D turbulence parameters are visualized using the user's hand orientation and position.*

fluid flow is by emitting particles at key locations in the domain. In real-world experiments, these actions form *streak lines* resulting from the continuous marking of fluid elements being emitted from a point. In essence, this is akin to walking around with an incense stick and watching how the smoke disperses. We create streak lines in the virtual environment by tracking the user's hand (or a wand that they hold) and letting particles emit from that point in space. The hand-tracked location can be extended from the user, affording for greater reach from the station point of the user. Changing the translational gain on the hand-tracked emitter and mapping it to the domain size allows for large-scale placement of streak lines. In spaces where turbulence varies greatly, binding the emitting location to hand movements results in smooth and precise placement of the streak line, which in turn, provides more effective visual reinforcement for how the model is working. The right side of Figure 3 shows a user in our lab using the system.

6. Discussion and Conclusion

The system supports the wind display of the TPAWT virtual environment, but has emerged into a novel engineering tool for fast-response urban dispersion simulations. The use of the real-time virtual environment for both dispersion modeling and simulation is the key contribution of this work.

We continue to develop this system to be both practical and intuitive. For instance, stereo-viewing of large particle systems is problematic. In scenarios in which the user is situated within the urban domain viewing thousands to millions of particles, the snowstorm-like effect of particles moving toward or away easily causes problems for stereo fusion. For the TPAWT, this is not yet a problem since it does not currently support stereoscopic viewing. However, for head-mounted displays, this is a problem. One way we are working to resolve this issue is to sample particles entering a specific region and only draw a subset of path lines from the total set of particles that enter into the area. This affords engineers a means to focus on specific flow and turbulence while still allowing the simulation to execute around them.

Along with visual aids, we are extending our head-mounted display environment to provide intuitive gesture-based interaction for our dispersion simulations. We plan to use these interaction mechanisms to specify additional visualization properties and control the simulation. We believe that the resulting virtual environment will be effective at developing more accurate models for particle dispersion, and also for educating people about the relationship between environment, wind, and urban form.

7. Acknowledgements

This material is based upon work supported by the National Science Foundation under grant 0428856.

References

[BBSar] Bratkova M., Boulos S., Shirley P.: orgb: A practical opponent color space for computer graphics. *IEEE CG&A* ((to appear)).

[HBSL03] Harris M. J., Baxter W. V. I., Scheuermann T., Lastra A.: Simulation of cloud dynamics on graphics hardware. In *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware* (2003), pp. 92–101.

[HGP05] Hollerbach J., Grow D., Parker C.: Developments in locomotion interfaces. In *IEEE 9th International Conference on Rehabilitation Robotics* (2005), pp. 522 – 525.

[KDPM06] KIRKMAN R., DEAVER M., PARDYJAK E., METZGER M.: Sensitivity analysis of a three-dimensional wind tunnel design. In *Joint U.S.-European Fluids Engineering Summer Meeting, FEDSM 2006* (2006).

[KKKW05] KRÜGER J., KIPFER P., KONDRATIEVA P., WESTERMANN R.: A particle system for interactive visualization of 3d flows. *IEEE Transactions on Visualization and Computer Graphics 11*, 6 (2005), 744–756.

[KMDP07] KULKARNI S., MINOR M., DEAVER M., PARDYJAK E.: Output feedback control of wind display in a virtual environment. In *2007 IEEE International Conference on Robotics and Automation* (Apr. 2007), pp. 832–839.

[KNBP06] KOPPER R., NI T., BOWMAN D., PINHO M.: Design and evaluation of navigation techniques for multiscale virtual environments. In *IEEE Virtual Reality Conference, 2006* (2006), pp. 175–182.

[KSW04] KIPFER P., SEGAL M., WESTERMANN R.: Uberflow: a gpu-based particle engine. In *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware* (2004), pp. 115–122.

[LBS03] LI G., BORDOLOI U., SHEN H.: Chameleon: An interactive texture-based rendering framework for visualizing three-dimensional vector fields. In *IEEE Visualization 2003* (2003), pp. 241–248.

[LFKZ01] LAVIOLA J., FELIZ D., KEEFE D., ZELEZNIK R.: Hands-free multi-scale navigation in virtual environments. In *Symposium on Interactive 3D Graphics* (2001), pp. 9–15.

[MJS97] MINE M., JR. F. P. B., SEQUIN C.: Moving objects in space: Exploiting proprioception in virtual-environment interaction. In *Proceedings of SIGGRAPH 97* (1997), pp. 19–26.

[PB01] PARDYJAK E., BROWN M.: Evaluation of a fast-response urban wind model comparison to single-building wind-tunnel data. In *2001 International Symposium on Environmental Hydraulics, Tempe, AZ* (2001).

[PSNW07] PARDYJAK E., SINGH B., NORGREN A., WILLEMSEN P.: Using video gaming tech. to achieve low-cost speed up of emergency response urban dispersion simulation. In *AMETSOC Conf Coastal Atmospheric Prediction* (Sept. 2007).

[QZF*04] QIU F., ZHAO Y., FAN Z., WEI X., LORENZ H., WANG J., YOAKUM-STOVER S., KAUFMAN A., MUELLER K.: Dispersion simulation and visualization for urban security. In *IEEE Visualization '04* (2004), pp. 553–560.

[Ree83] REEVES W.: Particle systems - a technique for modeling a class of fuzzy objects. In *Computer Graphics* (1983), vol. 17, pp. 359–376.

[Rod96] RODEAN H.: *Stochastic Lagrangian models of turbulent diffusion.* American Meteorological Society, 1996.

[SCC05] SCHEIDEGGER C., COMBA J., CUNHA R.: Practical cfd simulations on the gpu using smac. *Computer Graphics Forum 24*, 4 (2005), 715–728.

[Ste06] STEED A.: Towards a general model for selection in virtual environments. In *IEEE Symposium on 3D User Interfaces* (2006), pp. 103–110.

[WBB*04] WILLIAMS M. D., BROWN M. J., BOSWELL D., SINGH B., PARDYJAK E.: Testing of the quic-plume model with wind-tunnel measurements for a high-rise building. In *Fifth AMS Symposium on the Urban Environment* (2004).

[WBP04] WILLIAMS M., BROWN M., PARDYJAK E.: Development and testing of a dispersion model for flow around buildings. In *4th AMS Symposium on the Urban Environment* (Norfolk, VA, May 2004).

[WNSP07] WILLEMSEN P., NORGREN A., SINGH B., PARDYJAK E.: Development of a new methodology for improving urban fast response lagrangian dispersion simulation via parallelism on the graphics processing unit. In *11th Intl Conf on Harmonisation within Atmospheric Dispersion Modeling* (July 2007).

[ZSH96] ZÖCKLER M., STALLING D., HEGE H.: Interactive visualization of 3d vector fields using illuminated stream lines. In *Proceedings of Visualization '96* (1996), pp. 107–113.