Urban Windflow

Investigating the Use of Animation Software for Simulating Windflow around Buildings

Vignesh Kaushik[1], Patrick Janssen[2]
[1]Aedas, Singapore [2]National University of Singapore, Singapore
[1]vigneshkaushik@gmail.com [2]patrick@janssen.name

The animation and visual effects industry is producing advanced software capable of generating realistic behaviours faster than ever by using algorithms that approximate the physics of the real world. There is an opportunity to utilize these software to support performance-based conceptual design for complex simulations such as Computational Fluid Dynamics (CFD). This paper investigates a method of performing windflow simulation using an animation software that implements an Eulerian based smoke solver. These simulations run orders of magnitude faster than the similar simulations in dedicated high-end CFD applications. The paper compares the animated simulation results to a benchmark case with measured wind-tunnel data. The results indicate that at certain points in the animation, the accuracy is very high. However, the challenge lies in predicting best frame at which to stop the animation. The paper ends with a discussion of how this challenge might be tackled.

Keywords: *Computational fluid dynamics, Performance-based design, Smoke solver, Natural ventilation, Concept design*

INTRODUCTION

Performance-based design is an approach that leverages iterative simulation as a way of exploring design options. Simulating natural phenomena like internal and external airflow is becoming increasingly important to architectural design. Computational Fluid Dynamic (CFD) is a branch of fluid mechanics that is primarily used to solve problems involving fluid flows through proven numerical methods. The aim of this paper is to make CFD analysis more accessible to designers during conceptual stages of design to support making performance-based design decisions.

CFD has been used for decades in the automotive and aerospace industries with considerable success. This highlights the compelling possibilities for integrating CFD analysis in architectural and urban design as part of a performance-based design process. The use of CFD for architectural and urban design is a growing research field, with particular focus on understanding windflow patterns and natural ventilation. However, dedicated high-end CFD applications are very complex and require users to have a high level of expertise. Moreover, the simulations are computationally very expensive and may need to be run multiple times in order to achieve convergence. As a result, CFD simulations tend to be very slow, even when using powerful computer hardware. Therefore, CFD simulations are not commonly undertaken, especially in conceptual design, where many critical decisions pertaining to building performance are made (Bogenstatter 2000; Bazjanac at al. 2011).

Canonical Methods

Dedicated high-end CFD applications model fluid flow using a set of partial differential equations based on the Incompressible Navier-Stokes equations. Examples include ANSYS Fluent [1], Cradle scSTREAM [2], and PHOENICS [3]. The Navier-Stokes equations are the canonical model to simulate fluid flow (Stam, 1999) and are used as the basic building block of almost all solvers. A solution of the Navier-Stokes equations is called a velocity field or flow field, which is a description of the way the fluid wants to move at a given point in space and time. The equations are typically solved by splitting them up into several smaller parts and solving these separately. Each part updates the velocity field consecutively during every discrete time step.

The canonical method consists of a preprocessing phase where the simulation is set up, a simulation phase where the Navier-Stokes equations are iteratively solved over a series of time-steps, and post processing phase where the results are analysed.

For the canonical CFD method, two key sets of issues are identified: issues related to meshing of the design geometry during the preprocessing phase and issues relating to the speed of the simulation during the simulation phase. With regards to meshing, a watertight, volumetric mesh of the design geometry is required. The meshing has a significant impact on the speed and accuracy of the simulation, and requires a careful balancing between simplification of geometry and resulting accuracy of the simulation. With regards to the speed of the simulation, the process of solving of the Navier-Stokes equations is computationally expensive and simulations need to be run multiple times to achieve convergence.

The canonical method has proven too slow and cumbersome for designers to adopt, especially when working at the conceptual design stage where time and computational resources are limited. The meshing process is typically laborious and requires a modeller with extensive skills and experience. Running the simulations takes a lot of time, even when powerful hardware is available. In an iterative design process, designers explore multiple options before arriving at a particular building design and as a result these issues are exacerbated.

Alternative Methods

Performance-based analysis adds the most value to the design process only when it provides fast and actionable insight (Hensel and Menges 2008). As a result, designers need to have access to tools that produce adequately informative results within a reasonable time frame without compromising on the fluidity of the design workflow.

There have been many workarounds in literature to address the challenges in utilizing CFD to support early stages of architectural design process. For example, Kajima et al. (2013) recently developed a custom tool kit that combines geometry and data from McNeel Rhinoceros [4] and ANSYS Fluent to offer interactive 3D visualizations of simulated physical phenomena. Though it partly solved the problem of enabling designers understand flow field data in relation to architectural geometry, the challenges of mesh generation and speed of CFD remained.

At a more fundamental level, two approaches to tackling the issue of speed are solution approximation and solver approximation. Solution approximation encompasses approaches that extract solutions through intelligent interpolation from large data sets of simulation data of pre-defined set of design variants that only vary in fairly limited and well-defined ways. Since the design variants are simulated using a dedicated high accuracy CFD solver, the validity of the basis data is high. This could be useful in certain specific cases of design exploration and optimization, (Wilkinson & Hanna 2014). However, in general it is not well suited for architecture and urban design, as design options will typically vary significantly from one another. In addition, a key characteristic of solution approximation is the reduction of full 3D spatial field data available through CFD simulation to a 2D plane area of interest. For most designers, it is preferable to be able to see the spatial field data.

Solver approximation encompasses approaches where the focus is on the simplification of the CFD solver itself through simplified meshes or lower-order equations. Though this will inevitably result in greater inaccuracies in results, this level of accuracy may still be high enough for making broad decisions for architecture and urban design. Furthermore, these approaches do not require the reduction of the full 3D spatial field data to a plane. Designers will therefore have access to the spatial field data critical for identifying flow patterns around buildings.

Paper Contents

This paper investigates an alternative method focusing on the simulation of windflow around buildings using animation software. The animation software uses solver approximation to speed up the simulation by orders of magnitude. The next section describes the proposed method. The experiment section presents the results from the animation software and compares them to a benchmark with measured wind-tunnel data. Finally, the discussion section briefly indicates future avenues of research.

PROPOSED METHOD

The animation and visual effects industry is demanding ever more advanced tools capable of generating more realistic effects, faster than ever. To cater to this demand, animation software over the years have been enhancing their various solvers like rigid body solvers, particle solvers, finite element solvers, agent-based solver, smoke solvers etc. The purpose of these solvers is to produce visual effects that look realistic by using algorithms that approximate the physics of the real world. As computing power becomes cheaper, these approximations are becoming ever more accurate and faster to compute. Therefore, there is an opportunity to utilize these tools to support conceptual design enabling the simulation of a wide variety of physical and behavioural phenomena.

Two key issues have been identified: the meshing issue and the speed issue. With regards to meshing, many animation softwares have user-friendly interactive tools to support meshing. These tools allow 3D solids to be easily converted into voxels grids of varying sizes. This allows users to easily experiment with different grid sizes and to visualize the resulting volumetric forms. Furthermore, in addition to volume-based meshing, these software also support surface-based meshing. This further enhances the ease with which meshes can be created from design geometry. The challenge of meshing is therefore mostly overcome by the tools available in these software. With regards to the speed issue, animation software follows the approach of solver approximation.

Solver Approximation

Many alternate approaches have been developed for solver approximation for the purpose of visual effects. Among them the most popular are grid based Eulerian and Lagrangian approaches. The Eulerian approach is based on having stationary sample points that represent the state of the fluid. This can be done by defining a grid that divides the simulation domain into smaller voxels. A voxel stores current information about the fluid in that position. The Lagrangian approach uses small massless placeholder particles, which are moved around during the simulation. Particles can be used to store the necessary data and approximate the values at arbitrary locations in the field.

Many different Lagrangian fluid simulation plugins have been developed for major animation software for visual effects purposes. Maya Fluids [5], Glu3d [6], and RealFlow toolkit [7] are all fluid animation extensions that can directly integrated with Autodesk Maya [8]. These extensions use

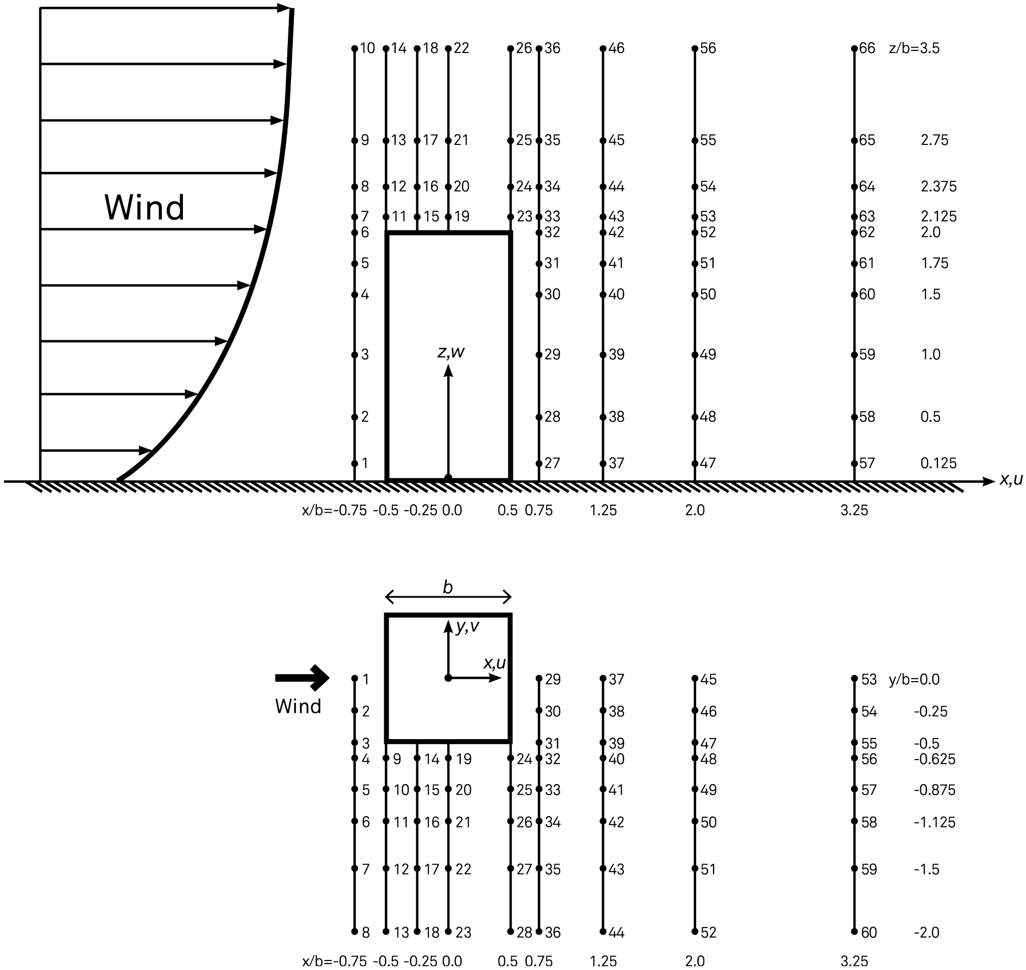

Figure 1
Outline of wind tunnel experiment. (top) 66 points in vertical cross-section (y = 0). (bottom) 60 points each in horizontal plane at z = 0.125b and 1.25b.

Smoothed Particle Hydrodynamics (SPH) methods introduced by Monaghan (1992). Autodesk Softimage [9] also has a particle-based incompressible fluid solver. Thuerey (2006) implemented a Lattice-Boltzman fluid solver for Blender. These methods are all based on Lagrangian approaches and have proven to be fast but physically not very accurate.

There is also the Semi-Lagrangian method called Fast Fluid Dynamics (FFD) introduced by Stam (1999) for the computer graphics and game industry. This method was subsequently validated by Chen and Zuo (2007) for indoor ventilation purposes and later proved to be much faster in comparison to conventional CFD models (Chen and Zuo, 2009). Chronis (2010) and Chronis et al. (2011) proposed the use of this approach in building engineering as a form-finding method and extended it for genetic optimization of façade apertures (Karagkouni et al. 2013) through a custom toolkit developed in Processing.js Environment [10].

All Lagrangian approaches require a lot of particles to achieve reasonable accuracy. Simulating such large numbers of particles would be too slow. Therefore, an Eulerian approach is considered better suited for windflow simulation (Lambright 2013).

Animated CFD

Higher-order grid-based voxel solvers using Euleiran approaches typically outperform Lagrangian approaches in physical realism. The animation software SideFX Houdini [11] has a smoke solver that uses the Euleiran approach. Houdini's node based architecture also allows the user to have more control over fluid simulations.

The proposed method uses Houdini for all three phases of the CFD simulation process. In the preprocessing phase, the Houdini tools are used to mesh the design geometry, to create a voxel grid for the spatial data field, and to set up the parameters related of the smoke solver. In the simulation phase, the animation is run, and the smoke solver is then used to calculate the wind velocity for each voxel at each frame of the animation. Finally, in the postprocessing phase, the wind velocities are visualized as point trails and the data is filtered and analysed. The data can also be exported for further analysis in other tools.

The focus of the next section is the validation Houdini's smoke solver as a performance indicator for wind simulation within the iterative design exploration process. For the validation, the results from the smoke solver are compared to a benchmark case with measured wind-tunnel data.

BENCHMARK EXPERIMENT

The benchmark for the experiment is taken from a set of CFD guidelines proposed by the Working Group of the Architectural Institute of Japan (AIJ) (Yoshiea et al. 2008). These guidelines were developed for the cross-comparison between CFD predictions, wind tunnel test results and field measurements, for seven test cases. The results from the test cases were meant to be used to validate the accuracy of CFD codes used in prediction of the wind environment around buildings.

For the purpose of this paper, the first test case of wind flow around a single high-rise building is simulated using Houdini's smoke solver. The aim of this experiment is to compare the CFD results from Houdini with AIJ's results and analyse the correlation.

Benchmark

A high-rise building model with the scale ratio of 1:1:2 (width:depth:height) was placed within the surface boundary layer as shown in Figure 1. The profile of wind inflow was provided as velocities at 25 points along the z axis. The velocity varied with height, with lower velocities near the ground and higher velocities as the height rises, as shown in Figure 1. The exponent for the power law of the vertical profile of average wind velocity was approximately 0.27. It was confirmed that the results from computation under basic conditions were not influenced by size of computational domain, grid discretization, and upper and lateral boundary conditions (Mochida et al. 2002).

A total of 186 points, 66 points in vertical cross

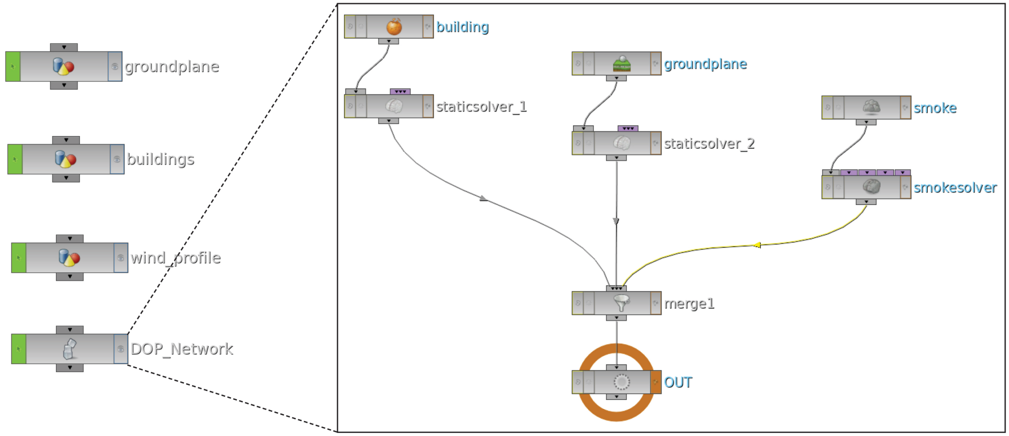

Figure 2
Network in Houdini's dynamics network.

section and 120 points in two horizontal planes, were defined. A split film probe was used to measure wind velocity, and the average wind velocity in each direction of three-dimensional space and the standard deviation of fluctuating wind velocities were determined for the 186 points.

Houdini Setup

Houdini is a procedural node-based modelling software with a number of contexts specialized for specific tasks. For this experiment, two of these contexts are used. The geometry context (called the SOP context) is used for modelling the building, the groundplane, and the wind profile. The dynamics context (called the DOP context) is used for running the smoke simulation. In all these contexts, modelling is performed by wiring together nodes that perform certain actions. After the simulation is complete, the geometry context is again used to visualize and analyse the results of the smoke simulation.

The dynamics context handles all non-particle dynamic simulations within Houdini. The Microsolvers in the dynamics context are powerful mathematical building blocks that can be used to build more complex solvers. Microsolvers are wired together by using a *merge* node, with the order of the inputs defining the order of execution. Although custom solver can be created in this way from scratch, for the purpose of this experiment, Houdini's predefined *smoke* solver is used.

The dynamics network used in this experiment is shown in Figure 2 (right). Data flows through a network from top to bottom for each frame of the animation. The state of the animation at each frame is calculated based on the previous frame.

The simulation in the dynamics network imports data from three geometry networks for the groundplane, the building, and the wind profile. The building is imported into the dynamics context as a rigid body collision object and is automatically meshed. The meshing process is defined by various parameters controlling among other things the size of the mesh. For the experiment, *volume based collision detection* using the high-accuracy *ray intersect* mode was adopted. The overall collision volume was then finely meshed using a voxel size of 0.1.

The *smoke* node in dynamics network defines the general settings for the smoke simulation. It defines the 3D volume for the smoke simulation and discretizes this volume into a grid of cube-shaped voxels, as shown in Figure 3. The location of the (u,v,w) velocity components are located on the minimal faces of each voxel. The density, temperature, and pressure values are stored at the center of the voxel, which in this case are not of use. The *smoke* node uses the wind profile specified in the wind profile geometry network. The *smoke* node is connected to the *smokesolver* node.

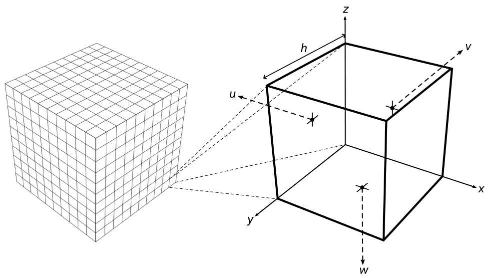

Figure 3
(left) 3D space divided into evenly spaced voxels. (right) Velocity components (u, v, w) in a voxel.

The outputs from the two *staticsolver* nodes and the output from the *smokesolver* node are then all merged, thereby allowing the solvers to interact with one another. Once the animation is started, the velocities of each voxel in the smoke volume will be updated. On completion of the animation, the simulated velocity data for any frame of the animation can then be visualized in various ways, including as points with trails or as coloured volume slices.

In order to compare the results of the simulation with the measured wind-tunnel data, the wind velocity data has to be calculated for the 186 points specified in the benchmark experiment. For this, the velocity data for all voxels in the smoke volume are imported back into the geometry context. The 186 points are then defined and the wind velocities at these points are calculated by using a node that interpolates between the voxels. The final output of the simulation is then the (u, v, w) components of the velocities at the 186 points at any frame of the animation.

Results

Houdini's smoke simulation was run multiple times for 200 frames with various voxel sizes and other simulation settings. Depending on the voxel size, the simulation took around 1-5 minutes to complete 200 frames on a professional workstation. These initial experiments showed that a voxel size of 1.5 units produced the best results. Smaller and larger voxel sizes both resulted in a general but relatively small deterioration in accuracy. For the description of results that follow, the simulation data generated using 1.5 sized voxels is used.

The comparison compares the simulated data to the benchmark measured data. For all the 186 points, the velocity vectors (u, v, w) from the simulated data are compared directly with the velocity vectors in the benchmark measured data. The comparison is performed in two ways: by calculating *average errors* and by calculating *number of bad points*.

For the average error calculation, the length and angle of each velocity vector is calculated, for both the simulated data and the benchmark measured data. The average of the absolute difference between these values for all 186 points is then calculated. This results in two scores: the average absolute error in the magnitude and the average absolute error in angle. These are referred to as the *magnitude error* and the *angle error* respectively. These error values do not take into account the standard deviations for the benchmark measure data.

For the number of bad points calculation, a comparison is made between the (u, v, w) components of the simulated data and the benchmark measured data. In this case, the comparison takes into account the measured standard deviations of the fluctuating velocities. For each point, if the deviation of any of the (u, v, w) components is greater than twice the standard deviation for that component, then the point is marked as a bad point. This results in a single score: *the total number of bad points*.

Figure 4 shows a graph that plots results of the magnitude error versus the angle error for all 200 frames. Depending on the frame number the magnitude error ranges from 0.9 to 1.3 m/s and the angle error ranges from 16 to 23 degrees.

The more significant result is the shape of the curve that is generated over the 200 frames of the

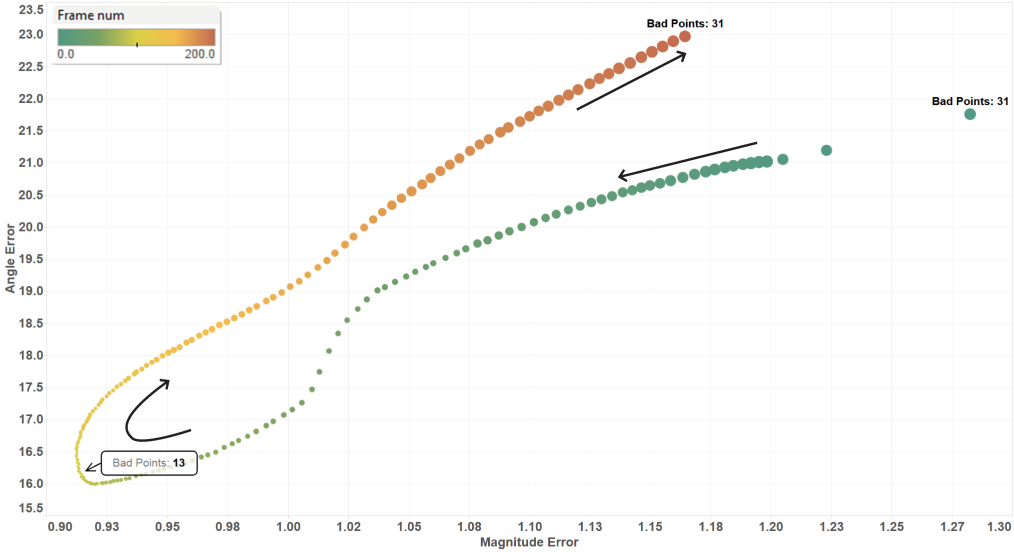

Figure 4
Average absolute errors in velocity with respect to the measured wind-tunnel data for 200 frames of the animation. The size of circle indicates the number of bad points in that frame. The direction of the arrows indicates increasing frame number.

simulation. The errors reduce as the simulation progresses, until about frame 100, beyond which the errors start to increase again. A similar pattern also emerges with respect to the number of bad points. The total number of bad points starts at 31, then reduces to a low of 13 at frame 97, and then increases again to 31 at frame 200. This general pattern of simulation accuracy improving and then deteriorating was found to occur for a wide range of different voxels sizes and simulation settings, with the turning point always falling within about 10 frames of frame 100.

Figure 5 shows the visualization in Houdini of a set of volume slices extracted from the simulation data. Analyzing the bad points in more detail also highlights certain patterns. In particular, the bad points are always those points located closest to the building structure. For the points a little further away from the building, the accuracy improves significantly.

DISCUSSION

In summary, the proposed method seems promising with regards to the two key issues of meshing geometry and simulation speed. With regards to meshing, this can easily be performed using the interactive tools available in Houdini. With regards to speed, the animated CFD was orders of magnitude faster than dedicated high-end CFD applications. The most promising aspect of this method is its potential to give fast and actionable insight to designers during early conceptual design.

The results indicate that if the issue of choosing the best frame can be overcome, then significant improvements in speed can be achieved with a reasonable trade-off in accuracy. At the best frame, the error is within 5-10% from the standard deviation. It is noted that the wind velocity errors for simulations performed with dedicated high-end CFD applications, when compared to the benchmark measured data, are also in the range of 5-10% (Yoshiea et al. 2008). Overall, the values from smoke simulation for the best frame are seen to be consistent with the

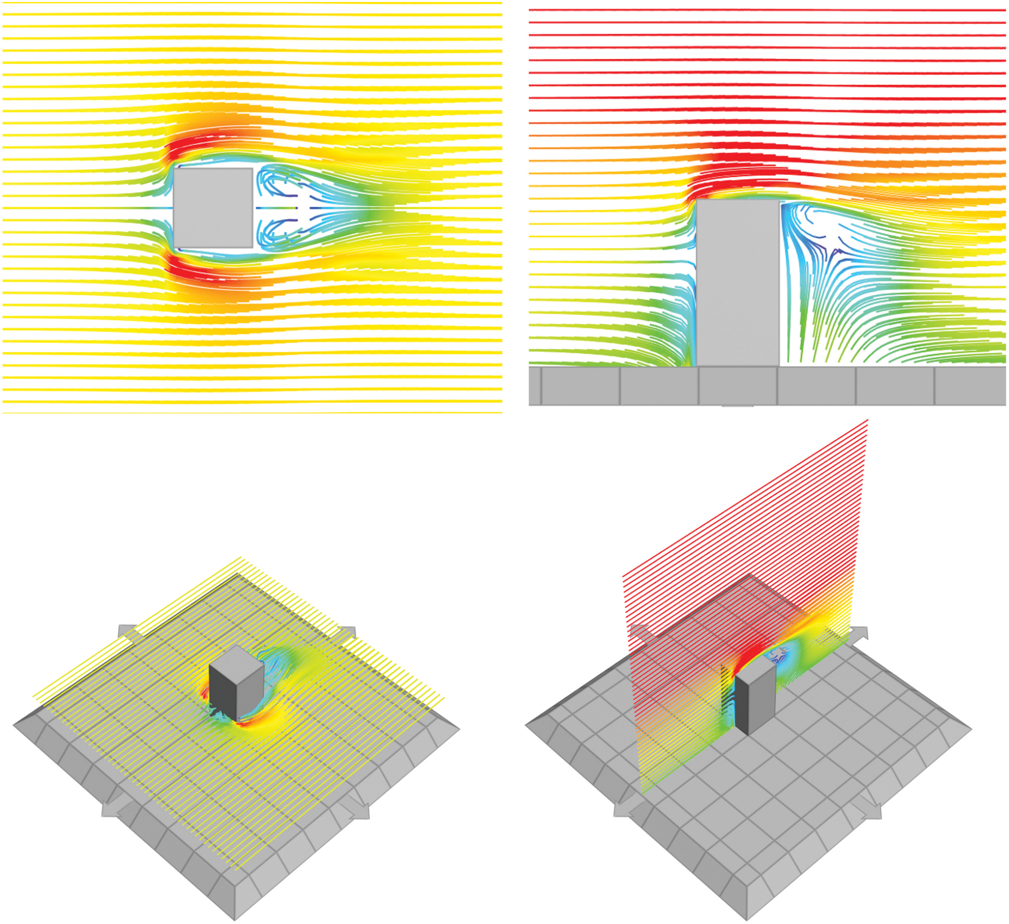

Figure 5
Volume slices extracted from Houdini's smoke simulation.

experimental values. However, they key question is how to identify the best frame.

Future work will compare Houdini's smoke solver to the other test cases specified in the AIJ guidelines, focusing in particular on the issue of how to identify the best frame at which to stop the animation. This will include exploring strategies that make the simulation converge to a state of reasonable accuracy.

REFERENCES

Bazjanac, V, Maile, T, Rose, C, O'Donnell, J and Mrazovic, N 2011 'An assessment of the use of Building Energy Performance Simulation in early design', *Proceedings of Building Simulation*, p. 1579–1585

Bogenstätter, U 2000, 'Prediction and optimization of life-cycle costs in early design', *Building Research & Information*, 28(9), p. 376–386

Chronis, A 2010, *Generative Fluid dynamics, Integration of fast fluid dynamics and genetic algorithms for wind loading optimization of a free form surface*, Master's Thesis, UCL

Chronis, A, Turner, A and Tsigkari, M 2011 'Generative fluid dynamics: integration of fast fluid dynamics and genetic algorithms for wind loading optimization of a free form surface', *Proceedings of the 2011 Symposium on Simulation for Architecture and Urban Design*, pp. 29-36

Foster, N and Metaxas, D 1997 'Modeling the motion of a hot, turbulent gas', *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pp. 181-188

Hensel, M and Menges, A 2008, 'Inclusive performance: Efficiency versus effectiveness. towards a morpho-ecological approach for design', *Architectural Design*, 78(2), pp. 54-63

Kaijima, S, Bouffanais, R and Willcox, K 2013 'Computational Fluid Dynamics for Architectural Design', *Proceedings of the 18th International Conference on Computer-Aided Architectural Design Research in Asia (CAADRIA 2013)*, Singapore, pp. 169-178

Karagkouni, C.S, Fatah, A, Tsigkari, M and Chronis, A 2013 'Façade apertures optimization: integrating cross-ventilation performance analysis in fluid dynamics simulation', *Proceedings of the Symposium on Simulation for Architecture & Urban Design*, pp. 1-9

Lambright, B 2013, *Eulerian Smoke Simulation*, Master's Thesis, Bournemouth University

Mochida, A, Tominaga, Y, Murakami, S, Yoshie, R, Ishihara, T and Ooka, R 2002, 'Comparison of various k-εmodel and DSM applied to flow around a high-rise building - report on AIJ cooperative project for CFD prediction of wind environment', *Wind & Structures*, 5(2-4), pp. 227-244.

Monaghan, JJ 1992, 'Smoothed particle hydrodynamics', *Annual review of astronomy and astrophysics*, 30, p. 543–574

Stam, J 1999 'Stable fluids', *Proceedings of SIGGRAPH 1999*, p. 121–128

Thuerey, N 2006, 'Fluid simulation with blender', *Dr. Dobbs Journal*, 1, p. 1

Wilkinson, S and Hanna, S 2014, 'Approximating Computational Fluid Dynamics for Generative Tall Building Design', *International Journal of Architectural Computing*, 12(2), pp. 155-178

Yoshiea, R, Mochidab, A, Tominagac, Y, Kataokad, H, Harimotoe, K, Nozuf, T and Shirasawag, T 2008, 'Cooperative project for CFD prediction of pedestrian wind environment in the Architectural Institute of Japan', *Journal of Wind Engineering and Industrial Aerodynamics*, 96, pp. 1749-1761

Zuo, W and Chen, Q 2007 'Validation of fast fluid dynamics for room airflow', *10th International IBPSA Conference (Building Simulation 2007)*, pp. 980-983

Zuo, W and Chen, Q 2009, 'Real-time or faster-than-real-time simulation of airflow in buildings', *Indoor Air*, 19(1), pp. 33-44

[1] http://www.ansys.com/Industries/Construction/Ventilation+&+Comfort+Modeling

[2] http://www.cradle-cfd.com/products/scstream/

[3] http://www.cham.co.uk/default.php

[4] https://www.rhino3d.com/

[5] http://download.autodesk.com/global/docs/maya2014/en_us/files/Introducing_Maya_Fluid_Effects_Fluid_Effects_overview.htm

[6] http://3daliens.com/joomla/

[7] http://thevault.realflow.com/docs/rfrk-mentalray-maya.pdf

[8] http://www.autodesk.com/products/maya/overview

[9] http://www.autodesk.com/products/softimage/overview

[10] http://processingjs.org/

[11] https://www.sidefx.com/