# INTEGRATED OCCUPANT MOTION SENSING WITH REAL-TIME CFD SIMULATION AS A DESIGN ASSESSMENT TOOL

Aman Singhvi[1], Nada Tarkhan[1],
[1]Graduate School of Design, Harvard University
Cambridge, Massachusetts, USA

## Abstract

The research presented is an investigative work on real-time interactive Computational Fluid Dynamics (CFD) simulation where a section of a wind tunnel is computationally simulated using Lattice Boltzmann equations for fluid flows. This research introduces a different approach to dynamic fluid simulation in design, which uses physical parameters from the built environment and creates a user-interactive output interface. The intention of the tool is to a) make early stage airflow analysis in the design process more intuitive to all the stakeholders involved as well as b) invoke future conversations for simulation engines to input live data from physical sensors. The LBM code is built in Microsoft Visual Studio and is connected to Microsoft Kinect V2 to detect the occupant's movement and mirror the resulting disturbance in air inside the simulation. A flow field with initial conditions is generated with parameters such as dynamic viscosity, initial velocity, fluid density and boundary conditions. A *Module* of Kinect sensor and HD projectors is organized in the space to create a sensing and projection network.

Three experiments were setup on a 1:1 scale in spaces with increasing area, height, number of people equipment capacity. The disturbances in airflow caused by people's motion were captured and projected live on a fabric screen for visual assessment. The experiments validated the ability of the method to simulate transient airflow with live occupant states and highly customizable boundary conditions along with furniture input. The built system was observed to output seamlessly at 30fps with occupants interacting with the flow field in the physical world. The system also displayed high quality visuals for an engaging interactive experience.

## Keywords

Real-time CFD, Lattice Boltzmann, Occupant Motion, Spatial Interaction, Wind Tunnel, Early Stage Design, Educational Tool.

## Introduction

Computational fluid dynamics (CFD) is a branch of fluid mechanics that utilizes numerical methods to solve and analyze problems involving fluid flows. The lattice Boltzmann method (LBM) is a parallel and efficient algorithm for simulating single-phase and multiphase fluid flows and for incorporating additional physical complexities. The LBM is especially useful for modeling complicated boundary conditions and multiphase interfaces (Chen and Doolen, 1998). For these reasons it was used in this application.

Real-time CFD using the Lattice Boltzmann Method (LBM) has been explored extensively in the past for the assessment of flow fields. CFD has been used to study ventilation schemes (Asfour and Gadi, 2007) as well as several aspects of building design. The predominant exploration domain of CFD so far has been with steady state models that do not account for movement of occupants or changing locations of obstructions in a flow field. One of the limitations of the development of transient models is the excessive simulation time (Jin et al., 2012). For this particular reason CFD studies are limited when it comes to providing quick exploratory outputs in the conceptual design phase.

Previous studies have integrated the use of supercomputers to accelerate the CFD computation process (Béghein et al., 2005). Yet such approaches demand the use of advanced equipment that may not be readily accessible to architects and building designers for the initial design stages.

The FFD method (Fast Fluid Dynamics) was developed to simulate indoor environments in real time and overcome some of the challenges of lengthy simulation time (Zuo and Chen, 2009; Zuo et al., 2010). It's original intention was to visualize fluid animations in the game industry (Stam, 1999). This method uses the Navier-Stokes equations and is based on the semi-Lagrangian and pressure projection. This approach may compromise in accuracy yet has been shown to run 50 times faster than standard CFD (Zuo and Chen, 2009).

### Design Application

Some of the barriers of entry to the use of such analysis techniques in early stage explorations is lack of sufficient information to construct accurate models and extensive computation time. Existing CFD software also lack user-friendliness for real-time exploration of results (Berger and Cristie, 2015). While the development of simulation software in the engineering world sets out to guarantee accuracy, the world of gaming has seen graphics that have placed speed and visualizations at higher importance. Hence computing and game engine technology can help bridge the gap between engineer and architect (Berger and Cristie, 2015).

In early stages of design, architects and students may value speed and user-friendliness more than precise results. The method undertaken for this research makes the argument that developing an interactive quick-to-use tool helps offset the inertia of software development in this field. A factor of inaccuracy is traded over precision to achieve faster computational speed, which is not detrimental to the tool's intended application as the design is still in early stages. The intention is also to encourage more though about CFD from early stages of design amongst the design and education community. Fast simulations can give an insight into flow properties at a preliminary stage in design without being too deterministic (Kaijima et al., 2013). Relative impact assessments would still be valuable in comparative studies across different design configurations and geometries.

### Lattice Boltzmann Method

The Lattice Boltzmann Method is a class of CFD that solves macroscopic fluid dynamics problems. The equations are derived from Boltzmann transport equations and it take into account energy and momentum transfer between finite sized particles of a fluid. LBM calculates the advection and local collision of a group of particle distribution functions f using the Boltzmann equation with linearized collision operator *Ω,* particle speed ξ located at the point x at time t, and external forces *F* (Lalle-mand et al. 2000).

$$\frac{\partial f}{\partial t} + \vec{\xi}\,\frac{\partial f}{\partial \vec{x}} + \vec{F}\,\frac{\partial f}{\partial \vec{\xi}} = \Omega(f, f) \qquad (1)$$

There are several advantages to using the LBM over the Navier Stokes equations in reference to this application. There are a few key factors that make it easier to implement, as it is explicit and defined only under local operations. Most importantly, data parallelization allows for accelerated computational solving, allowing for real-time results. The LBM method provides a higher degree of numerical stability as well as accuracy. Moreover, geometry and complex boundary processing is enhanced due to the nature of its parallelized data analysis (Meskas et al., 2011).

While there are advantages to adopting this method, a few limitations do exist when comparing to traditional CFD. Due to the explicit nature of its algorithms, it may require more time steps and can be memory intensive (Elhadidi and Khalifa, 2013). Furthermore, macroscopic scale boundary conditions can also be difficult to solve due to the multiple formulation types (Chen and Doolen, 1998).

## Methodology

The methodology is broken down into three discrete steps based on three major functions, each representing a different set of inputs, equipment and intended output. This three-step methodology is translated into a spatial experiment and tested in three different spaces to determine a correlation between the computational speeds, quantity of equipment, installation area and volume.

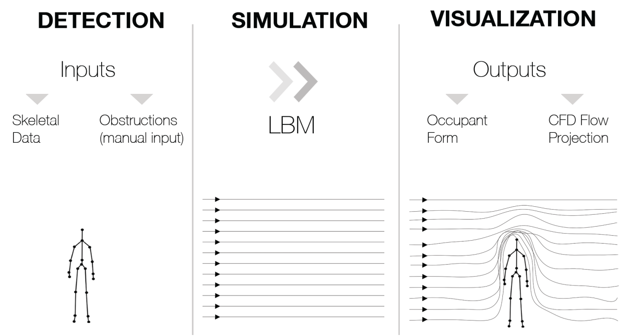



*Fig. 1- Methodology breakdown*

### 1. Detection and Mapping

**Detection:** The detection of occupants and their motion in the physical world is the first discrete step. There are many different options of hardware available today to achieve this but for our purposes Microsoft Kinect V2 was chosen for considerations such as cost and compatibility. The Kinect V2 (also known as Kinect for Xbox One) contains an RGB camera, depth sensor, Infrared (IR) emitters, Infrared receiver, multi-array microphone, as well as supporting cables that allows the unit to output sensor information to an external device via USB 3.0. The IR and Depth sensors in Kinect are used as our primary detection hardware, which can be accessed via the Kinect SDK 2.0. The Kinect sensors can also be custom programmed to recognize 25 joints of a human skeleton (bodies) with accessible data in terms of x, y and z coordinates of each joint in a 3-Dimensional Kinect grid system. These 25 joints are tracked seamlessly at a framerate of 30fps and can be tracked simultaneously for up to 6 bodies in the sensor field (two in full motion).

**Mapping:** The crucial step after the detection of bodies is to accurately map the 3D joints data onto a projection matrix based on a mapping function. By using the *KinectSe*nsor class and the *MultiSourceFrameReader* class from the Kinect SDK 2.0 libraries, the tracked bodies are stored in separate lists including both the frames data and joints information. The RGB image channel can also be superimposed on the frame data using the *CoordinateMapper* class within the KinectSensor instance in order to achieve a more realistic visual output. Since one of the objectives of the tool is interaction in real-time, LBM is used to generate a 2D flow system rather than 3D, representing a slice through the testing space. It is to be noted that the flow simulated in a given 2D slice is independent of the fluid conditions in any of the hypothetical slices in the Z-direction. Running multiple simulations in parallel connected to the same Kinect sensor can output multiple adjacent slices with accurate flows using one single 3D-depth data. A 3D flow field can thus be hypothetically achieved using multiple instances of the program running in parallel. The 2D lattice for LBM in the LBSystem class is constructed on 3D Kinect Coordinate system that is inbuilt in the sensor's software. In Kinect Sensor's cone-of-view, every point in the physical world has an x, y and z coordinate value assigned with reference to Kinect's origin, which is the center of its depth sensor at (0,0,0). The 2D LB lattice is created by defining a unit cell as a square 'container'. Each container contains a numerical definition of a finite quantity of LB Fluid whose behavior is governed by the Lattice-Boltzmann equations. This cell containing LB-Fluid is replicated to fill a 2D slice of physical space (a plane in Kinect's view-cone with constant z coordinates) by mapping the grid dimensions (Nx, Ny) and aspect ratio to that of the viewport. In this way, both the physical space and fluid space uses the same x, y and z coordinates from the physical world. This slice is a user-defined section of a space which acts as the analysis plane for simulation to occur.

This makes it possible to accurately map any occupant, physical object or rate of change of x,y (velocity) onto the simulation as every LB cell containing finite fluid quantity also represents a physical region of space having fluid parameters, visualization parameters as well as adjacent boundary condition value. However, to accurately detect occupants' feet, some position correction is added as offset distance from the ground within Kinect's view cone, which is reflected in the simulation.
The 2D coordinates of each tracked joint of the occupant (x,y) are mapped onto the LB-Lattice coordinates (LBx, LBy) so that the detected bodies are perfectly mapped and scaled on the simulated LB fluid lattice. The visual quality of the bodies (frames with joints) were modified for our experiments in order for the viewers to better recognize a human form within the flow field and also to avoid possible confusion of the joints with the fluid particles. For each tracked body b, three joints are connected consecutively through a quadstrip (Figure 3). Kinect's detection

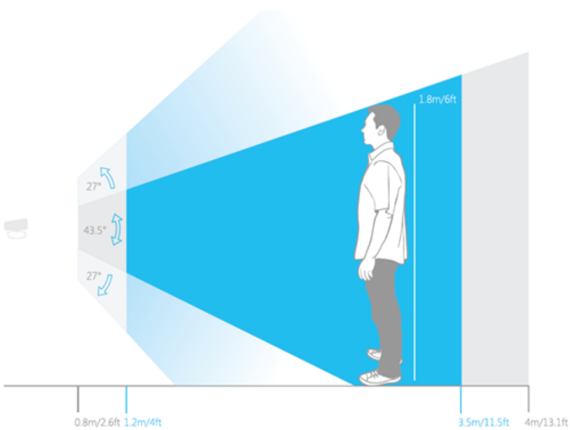



*Fig. 2- Kinect Detection Range*
https://msdn.microsoft.com/en-us/library/hh973074.aspx

range is 0.8m-4.0m horizontally from the center of its depth sensor (which is also the origin of its coordinate system at (0,0,0)) however; the effective 'sensing zone' to track full bodies is between 1.5m-4.0m from the origin (Figure 2). If a user is facing the front of the Kinect, the right hand and the head represents +x and +y axes respectively and the vector between the center (0,0,0) and the person represents +z axis.

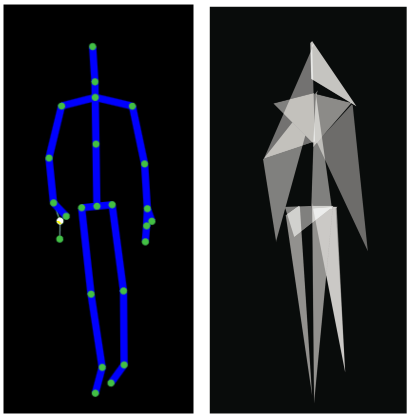


*Fig. 3- Occupant Visualization*

## 2. Simulation and Interaction

**Simulation:** The simulation step involves three tasks: 1) implementing LBM and initializing a simulation of a Newtonian fluid on a 2D lattice grid, assigning this lattice an optimum resolution, frequency of simulation (i.e. 'steps', which is a determinant of speed and fps) and a base color 2) creating *streamlines* on each point of the lattice to simulate a 'wind tunnel' condition and 3) creating solid particles in the fluid to better visualize bulk flow direction and points of fluid concentration/ stagnation. The basic LBM equations are natively coded in Microsoft Visual Studio 2015 Software using C# Programming language. Visual Studio is an Integrated Development Environment (IDE) that can be used to develop programs through the use of different programing languages. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms etc. The LBM implementation is done by creating a base build of Lattice Boltzmann Equations (*LBSystem*) and accessing kinect sensors from within the code to map bodies (joints) in the flow field. An *LBCFD* class is constructed to be the main library definition. This class contains *LBSystem* as sub-class along with a particles subclass (*LBParticles2D)* as well as a method to define boundary conditions (*BoundaryType),* which takes arguments; Solid, Continuous or Periodic. The base fluid's physical and interaction properties such as density range (RhoMin, RhoMax), Interface width, relaxation time and inter-particle interaction potential (G) are defined within the main class LBCFD. The *simulation step* function is defined to compute one step of simulation. The simulation is visualized through openGL. An openGL viewport (size: Nx,Ny) is constructed on a slice from Kinect's view-cone and the

LB lattice is mapped on it as Nx and Ny multiples of the unit cell in the X and Y direction respectively. A base color is also assigned for each cell of this fluid lattice. Using the *BoundaryType* method, the right and left boundaries are set to Periodic and the top and bottom boundaries are set to Solid, similar to a wind tunnel.

In the next layer of simulation, streamlines of finite thickness and given color are drawn starting at regular intervals throughout the simulation on points, which lie on the base fluid lattice. Apart from streamlines, small solid particles are also drawn as openGL objects in the field to add visual clarity to the flow. The fluid particles class, *LBParticle2D* creates a finite fluid particle. Each Particle *p* in the *LBParticle2D* class has the following attributes: Position (x, y), size, color(r, g, b, a) and derived Velocity (ux, uy). The *MoveParticles* method is defined to set initial velocity to all the particles in the simulation.

**Interaction:** The interaction capabilities are also implemented within the LBSystem namespace. These methods and statements consist of events in the physical world, which modifies the LB field. Various 'modifiers' are defined within the fluid properties such as *add/ remove solid block* (also useful to create furniture and physical objects), *change fluid velocity at a given point, add/ remove droplet of higher or lower density and add or remove a zero-density paint/ smoke of different color*. These modifiers are tied to the fluid simulation by certain events such as mouse clicks and motion of bodies detected by the kinect sensor. The simulation step is called at 30fps which is the same as kinect sensor's fps to achieve a real-time, interactive simulation. The inter-particle force implementation is based on the Shan–Chen-type (SC) multiphase lattice Boltzmann model. In the model, any typical equation of state can be incorporated and different contact angles of the gas–liquid interface at a solid wall can be obtained easily through adjusting the 'density of wall'. (Benzi et al. (2006)) This is calculated in the update function along with the inter-particle interaction potential, which is calculated at equilibrium velocity. The kinematic parameters are tied to the fluid physical parameters to accurately simulate a Newtonian fluid.

After the simulation is initialized and the first interaction is captured in the flow field, the 'disturbed' particles move along with the streamlines until equilibrium is achieved again in the system. The time duration after which this equilibrium state will be achieved depends on fluid properties and all the LBM force equations. Combining the quadstrip visualization of bodies (Figure 3) into the flow field, the intended visual quality is achieved. Since the objective of this research was to develop a visual-interactive tool, the focus of the parameters was more on the accuracy rather than precision.

## 3. Visualization

The third and final step developed through this methodology is the visualization. Traditionally, CFD simulations are represented on computer screens through the software interfaces. Users typically manipulate sectional cuts through the building with simulated static conditions. As the aim of this tool is user interaction, education as well as tracking transient conditions, the output of the visualization was projected on a 1:1 scale. This was done by setting up a fabric screen measuring the width of the room.

A wide-angle HD projector was necessary to capture the full width of the space. Two types of projectors were used depending on the scale of the room. For smaller areas a 4,000 lumens projector was used as opposed to two 10,000 lumens for large areas. The output was calibrated to give as close as possible representation of actual human scale. This encouraged the occupants to interact with the projection as their movements were captured in real time and represented before them. Alternatively, the same simulation could also be viewed on a computer screen without a projection component. In some of the simulations, the joints of detected Kinect bodies were set to add smoke of a different color at every instance of the frame. This smoke-like fluid is intended just for visualization purposes and is useful in clarifying the dynamics of the system. This smoke does not alter the physics of the simulation. The streamlines and base lattice visualization properties are made to be highly customizable and based on the visualization requirements of each space, the properties can be adjusted for better interactive experience.

### The Module

A unit *module* of equipment is defined as a way to assess various spaces with respect to the minimum hardware required to complete a detection-simulation-visualization cycle. The module is also helpful to assess the cost of equipment and to measure the scalability of the system. One module consists of three essential hardware components; One quantity each of Microsoft Kinect V2 sensor, HD Projector and a 64-bit Laptop Computer with minimum Intel Core i5 2.4Ghz Processor and an 8GB RAM. The module also includes 10m length each of HDMI, USB 3.0 and power cables. As discussed earlier, each Kinect sensor has an effective sensing zone area of 5.625 sq.m. and a maximum sensing height of 4 meters (Figure 2). The experiments section details how the number of modules employed in a system increases proportionally with the increase in area of the space and number of occupants moving through the sensor zone.

## Experiment

The experiment is designed to test mock-up spaces, evaluate the ability of the algorithm to detect occupants in real time and determine the limitations in both occupant detection and physical detection per given module of equipment. Figure 4 shows a schematic diagram of the sectional simulation study.

### Shoebox Room models

Similar to energy analysis where shoebox models are frequently used to test smaller, representative spaces- a set of room sizes was explored using the chosen methodology. Below is a summary of the testing parameters across the three scenarios.

| | Space 1 | Space 2 | Space 3 |
|---|---|---|---|
| **Area (m2)** | 20 | 54 | 96 |
| **Number of Occupants** | 1-2 | 1-4 | 1-6 |
| **number of Kinect sensors** | 1 | 1 | 2 |
| **number of projectors** | 1 | 1 | 2 |

*Table 1- Summary of experiment Setup parameters*

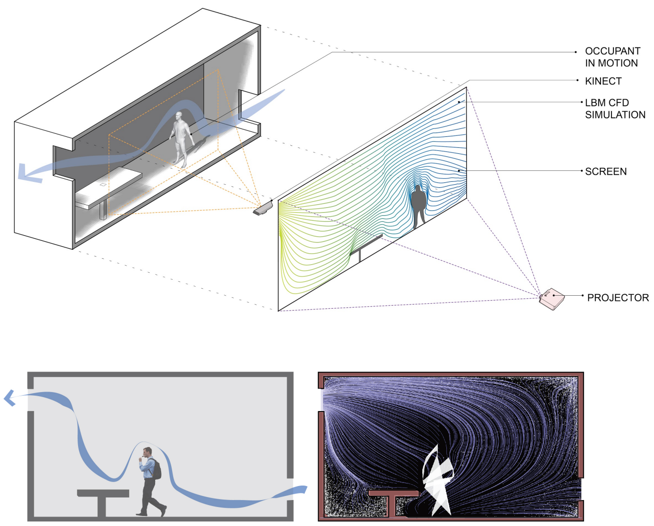



*Fig 4- Spatial Experiment Diagram*

There can be an infinite number of possible boundary conditions and room interior scenarios that can be modeled into the build; attributed to its highly customizable cellular lattice construction. Below are four possible boundary conditions for a given space for air inlet and outlet locations.

Using the inlet-outlet configuration of shoebox B, the spaces for the physical setup were chosen based on the above testing parameters represented in Figure 5. Multiple modules of Kinect sensors, projectors and a fabric screen were utilized to set up three separate experiments in three different spaces with increasing floor area, height and occupant movement per hour. The aim was to visually project simulation of air being disturbed in these spaces by increasing number of occupants moving through the Kinect sensing zone. These experiments were also designed to provide feedback on the real-time interactive capabilities of the system with respect to motion sensing capacity of the Kinect sensors, computational capacity of the build and number of people in the sensing zone in a given duration of time.

### Physical Setup

The experiments were setup in three spaces 1, 2 and 3 with areas 20 sq.m., 54 sq.m. and 96 sq.m respectively (Table 1). The height of projection in each space was 3.2m, 4.5m and 5.5m respectively. Space 1 setup utilized one module where the Kinect and the projector were arranged to be on the opposite sides of the central projection screen. Space 2 utilized one module with a 10K projector and the the Kinect and the projector were placed on the same side of the projection screen. The setup in Space 3 employed two modules with two 10K projectors where the Kinect and Projectors were placed on the same side as well. All three setups were calibrated to maximize motion capture within the

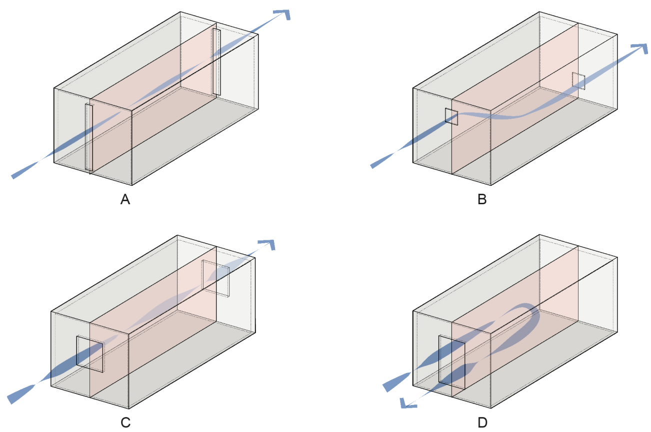



*Fig 5- Wind tunnel shoe-box room boundary Configurations*

Kinect sensor field and to get an unobstructed projection area. The design of the space using the

Fabric screen aimed to facilitate natural circulation of people within the sensing zone and ease their interaction with the CFD simulation.

The virtual experiment of colored smoke dispersion was carried out by inputting a concentration of fluid at the inlet (shown in blue in Figure 7) and monitoring its flow over a specific period of time. The experiments were carried out with the following initial conditions; Lattice size 300x300, initial wind tunnel fluid velocity as 10 m/s in the -y direction, number of LB-particles as 1000. Figure 7 shows experimentations on Space 1 with standardized boundary configuration based on shoebox B- one inlet and one adjacent outlet, representing two windows. The below three variations were simulated:

**Study 1:** One occupant, One table

**Study 2:** Two Occupants, One table

**Study 3:** One Occupant, Two tables, smoke at inlet.

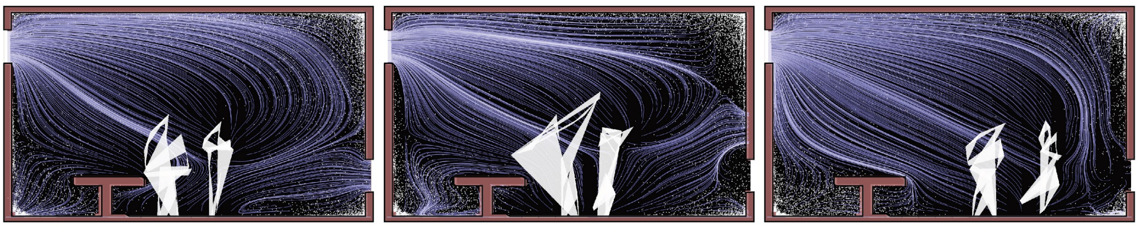
## Results and Discussion

The simulations were assessed on multiple parameters to examine their efficacy with respect to the intended outputs. Following were some of the observations from the initial testing, shoebox testing and the physical experiments: A) The build was successful in capturing occupant movement and corresponding fluid disturbances in real time. B) The system was found to have no lag before and after Kinect sensor was initiated however, a lag of about 2 seconds in the initialization of Kinect sensor was observed on startup. C) In the detection stage the interactive simulation was observed to run in real-time for up to three people in the sensor zone. After four people, computational lags were seen and the simulation steps were noticeable (discussed further in limitations). Using additional equipment modules in the space can mitigate this. D) The inputting of internal furniture (as fluid of very high density) allowed for multiple spatial configurations to be inserted in the flow field without affecting the speed of the simulation. E) The added mouse interaction component enabled operators to manually edit obstructions and boundaries in real time, adding to the educational and interactive component of the installation. F) The addition of multiple obstructions (number of tables, fixtures, and partition walls) did not affect the simulation speed. G) The simulation maintained a seamless 30fps frame rate with additional solid objects for up to 3 people. Thus, the experiments also validated the ability of the code to have as inputs, large number of possible boundary conditions and interior obstructions without loss in speed. H) In the initial shoebox testing, the fluid flow was found to vary across the four different shoe box models as per expected fluid behavior, this validated the usefulness of the system as an educational tool in design related CFD fields. I) The responsiveness of the visual components (streamlines and fluid particles) was found to be high considering their dynamic interaction with the detected Kinect joints for up to three bodies. J) The added colored smoke visualization was found to be useful to further understand the transient behaviour of smoke in the flow field.

Finally, the results verified the scalability of the model and confirmed that additional equipment modules can be connected in parallel to proportionally cover increasing areas. In Space 3 two Kinects and 2 projectors were used which allowed for the detection of more number of people. The main limitation of the occupant detection capacity is discussed in the next section.

Time t = 0.0 sec t = 2.0 sec t = 4.0 sec

Study 1: Effect of air-redistribution from the movement of one occupant (walking). Room with furniture, low inlet and high outlet.



Time t = 0.0 sec t = 2.0 sec t = 4.0 sec

Study 2: Effect of air-redistribution from the movement of two occupant. Room with furniture, low inlet and high outlet.

Time t = 0.0 sec t = 4.0 sec t = 8.0 sec

Study 3: Visualizing air velocity by placing colored smoke of set properties. Room with one occupant, two tables, low inlet and high outlet.

*Fig 6- Spatial Experiment Outputs at different time intervals*

**Limitations of Method**

Occupant detection capacity is reached when more than 3 people cross a single kinect sensing zone. This was indicated by a lag in the output and distorted occupant recognition within the sensing zone.

*Fig 7- Photograph of the installation; people interacting with the flow field. (Credits: Tina Tian Photography)*

Another limitation of this method is the lack of a dynamic thermal input model; even though movement is picked up within the simulated space and corresponding fluid disturbances are represented, thermal data from the occupants and surroundings is not detected. This would require the setup of thermal algorithms within the LBM code or possible integration of thermal sensor data (refer Future Work). In light of the objective of creating a real-time CFD simulation the thermal model was not implemented since it requires different spatial calibrations and a more complex model leading to an increase in computational time. Figure 6 shows a 2-D section of a space with furniture and 'freeze-frame' snapshots at regular time intervals of the disturbance of flow.

## Conclusion

This method demonstrates that a real time, transient CFD analysis is possible at an early stage of design when few variants are known. The method allows for quick testing of shoe-box wind tunnel simulations with various spatial configurations. It is successful in detecting occupant movement in real time. The dynamic creation and deletion of solid boundaries showed that it is possible to adjust room enclosures to reflect exact openings in a 2-D plane of a representative space. Repeating this process allows for a full spatial analysis. In addition, it could provide quick feedback on locations of windows as well as mechanical outlets. Integrating this type of exploration on design projects in both academic and professional settings could help bridge the gap between engineer and architect. This is however dependent on the further testing of the model.

The smoke dispersion-tracking feature makes use of the multiphase fluid simulation capabilities of the Lattice Boltzmann Method and can be further modeled to affect the physics of the simulation rather than pure visualization purposes. The current system is capable to incorporate such multi-phase simulation features as it is modeled using Lattice Boltzmann Equations, which supports multi-phase simulations. This would require further development and calibration in order to represent accurate particle concentrations and atmospheric conditions. Alternatively, this method could also be used to track $CO_2$ concentrations and diffusion as well as pollen dispersion from the exterior environment entering a space. The representational study shown in Figure 7 is designed to explore the multiphase simulation potentials of the algorithm to carry out this type of analysis.

## Future Work

Additional findings show that there are several opportunities for advancing this body of work. These are as follows:

**Performance Boosts:**

Recent testing has shown that significant performance boost can be achieved by using C++ as the programing language. This can elevate the potential of simulating multiphase fluid interaction in real-time or close to real-time. This can also help to achieve better visual output for a finer resolution of the base lattice without compromising the speed of simulation.

**Additional sensing capabilities:**

The current system can be customized to read dynamic data from additional physical sensors in space such as thermocouples, humidity sensors, Flir thermal camera etc. to add precision to the air flow simulation, however, this might affect the simulation speed. The coupling of thermal data on the base LB system could be especially beneficial to also simulate buoyancy driven flows within a space. This can assist in later stages of design to test specific thermal design ideas and placement of openings.

**Dynamic Object Detection and Tracking**

Using Computer Vision libraries such as OpenCV and EmguCV, physical objects with complex shapes can be detected and their motion can be tracked. In the existing system, this could be done by making use of the Kinect's RGB camera. Stand alone object tracking using a 640 x 480 resolution VGA camera has shown to detect complex objects in real-time (30fps).

**Three-dimensional Spatial Analysis:**

The current analysis only evaluates a two-dimensional plane of a room. A three dimensional simulation can be achieved by implementing the system in a different software with VR (Virtual Reality) capabilities or as an

Augmented Reality (AR) overlay on physical space through the use of Microsoft Hololens.
In addition to these technological upgrades, HTC Vive VR headset along with its interaction remote could provide an alternative to motion sensing using Kinect, as well as, provide an upgrade in accuracy and processing speed. In order to move this method forward in the direction of design assessment- other parameters need to be included such as room dimensions, depth of space, type of openings and window fixture types. The inclusion of these parameters would help refine the model and input more detailed information in the fluid model. Building types where this assessment may provide value could include office ventilation schemes or lab facilities where smoke detection and exhaust is required.

## Acknowledgements

We would like to thank Prof. Panagiotis Michalatos at Harvard Graduate School of Design, under whose valuable guidance the research is carried out.

## References

Asfour, Omar S., and Mohamed B. Gadi (2007). Using CFD to Investigate Ventilation Characteristics of Domes as Wind-Inducing Devices in Buildings. International Journal of Green Energy 4.6: 571-88.

Beghein, C., Y. Jiang, and Q. Y. Chen (2005). Using Large Eddy Simulation to Study Particle Motions in a Room. Indoor Air 15.4: 281-90.

Benzi R, Biferale L, Sbragaglia M, Succi S, Toschi F. Mesoscopic modeling of a two-phase flow in the presence of boundaries: the contact angle. E 2006; 74(2):021509.

Berger, Matthias, and Verina Cristie (2015). CFD Post-processing in Unity3D. Procedia Computer Science 51: 2913-922.

Bill Bao, Yuanxun, and Justin Meskas (2011). Lattice Boltzmann Method for Fluid Simulations.

Chen, Shiyi and Gary D. Doolen (1998). Lattice Boltzman Method for Fluid Flows. Annual Review of Fluid Mechanics 30.1: 329-364. Web

Elhadidi, Basman, and H. Ezzat Khalifa (2013). Comparison of Coarse Grid Lattice Boltzmann and Navier Stokes for Real Time Flow Simulations in Rooms. Building Simulation 6.2: 183-94.

Huang, Haibo, Zhitao Li, Shuaishuai Liu, and Xi-Yun Lu. Shan-and-Chen-type Multiphase Lattice Boltzmann Study of Viscous Coupling Effects for Two-phase Flow in Porous Media (2009). International Journal for Numerical Methods in Fluids 61.3: 341-54.

Jin, M, Zuo, W and Chen, Q (2012) Improvements of Fast Fluid Dynamics for Simulating Air Flow in Buildings. Heat Transf. Part B Fundam., 62, 419–438.

Kaijima, Sawako, Roland Bouffanais, Karen Willcox, and Suresh Naidu (2013). Computational Fluid Dynamics for Architectural Design. Architectural Design 83.2: 118-23.

Lallemand, Pierre, and Li-Shi Luo (2003). Lattice Boltzmann Method for Moving Boundaries. Journal of Computational Physics 184.2: 406-21.

Lallemand, Pierre, and Li-Shi Luo (2000). Theory of the Lattice Boltzmann Method: Dispersion, Dissipation, Isotropy, Galilean Invariance, and Stability. Physical Review E 61.6: 6546-562.

Stam, Jos (1999). Stable Fluids. Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques - SIGGRAPH '99: n. pag. Web.

Zuo, Wangda, and Qingyan Chen (2010). Fast and Informative Flow Simulations in a Building by Using Fast Fluid Dynamics Model on Graphics Processing Unit. Building and Environment 45.3: 747-57.

Zuo, Wangda, Jianjun Hu, and Qingyan Chen (2010). Improvements in FFD Modeling by Using Different Numerical Schemes. Numerical Heat Transfer, Part B: Fundamentals 58.1: 1-16.

Kinect Calibration: Technical description http://wiki.ros.org/kinect_calibration/technical

Kinect BodyTracking: https://msdn.microsoft.com/en-us/library/dn799273.aspx

Kinect Skeletal Tracking. https://msdn.microsoft.com/en-us/library/hh973074.aspx

Blockena, Carmelieta, Stathopoulos

CFD evaluation of wind speed conditions in passages between parallel buildings—effect of wall-function roughness modifications for the atmospheric boundary layer flow (2007). Journal of Wind Engineering and Industrial Aerodynamics