Adaptive Localization Method: An Approach to Real Time Airflow Simulation and Immersive Visualization

Ravi S. Srinivasan, Ali M. Malkawi
T.C. Chan Center for Building Simulation & Energy Studies, School of Design, University of Pennsylvania
Philadelphia, PA, USA
{sravi, malkawi}@design.upenn.edu

Abstract

Traditional approaches to simulate airflow movements in buildings are computationally expensive and do not achieve real-time prediction of results. This paper discusses an Adaptive Localization Method (ALM) that significantly minimizes the simulation domain to achieve close to real-time predictions. As the user interacts with the space by modifying boundary conditions (opening a window, etc.), while being immersed in an Augmented Reality (AR) environment, the ALM detects the changes and narrows down the simulation space significantly for re-simulation instantly. This localized space is simulated and the newly generated airflow data is updated to corresponding spatial nodes for interactive, immersive AR visualization. ALM is developed based on a series of simulations conducted to identify critical variables that alter the rate of change of velocity, magnitude, and temperature of air due to changes in the boundary conditions. The ALM based real-time AR model will aid in studying "what if" scenarios for buildings, particularly for applications such as remodeling and refurbishment to improve conditions, etc.

Keywords: *Adaptive Localization Method, Building Simulation, Performance Typology, Augmented Reality, Data Visualization.*

1. INTRODUCTION

Building simulation technology is used by experts using Computational Fluid Dynamics (CFD) to predict behavior of buildings such as natural ventilation design, prediction of smoke and fire in buildings, indoor air quality assessment, etc. CFD applies numerical techniques to solve the Navier–Stokes equations for fluid fields and provides an approach to solve the conservation equations for mass, momentum, and thermal energy. Such visualizatin in Virtual Environments (VE) is an active research area, stemming from existing research on Ves and CFD data visualization. Virtual Reality (VR) and Augmented Reality (AR) form part of today's VEs. Only a few projects have investigated the potential of this technology to better understand buildings and their behavior in an interactive, immersive AR environment [1].

In an immersive AR visualization of an indoor thermal environment, the user may interact and modify the boundary conditions (e.g. increasing the supply inlet temperature or velocity, opening / closing of the doors or windows, etc.). This may lead to changes in the thermal environment dynamically (e.g. direction and magnitude of air and temperature near the supply inlet, doors or windows, etc.). As these changes are instantaneous, conducting elaborate simulation of the entire space is time consuming and will not support such immersive visualization applications. This is primarily due to the size of simulation spaces and simulation algorithms. While larger simulation spaces may take more time to converge to a solution, existing fluid flow solvers may not provide results in real-time. Several research projects were conducted to simulate fluid flow in real time. Their underlying methodology primarily focused on reducing simulation time in physically based models, increasing simulation speed via hardware programming, employing data filters to sort relevant data, and using approximate modeling techniques to approximate indoor thermal datasets. Physically based models consist of physical equations that govern complex air movements. Such models are computationally expensive and cannot be employed for real-time applications. Some of the techniques followed include using low resolution grids to capture a coarse state of pressure and density fields [2], Zero-equation turbulence models [3], texture splats for fire animation [4] and motion of hot, turbulent gas [5]. Hardware programming allowed significant reduction in fluid flow simulation time. Earlier attempts employing hardware programming for fluid flow include interaction with smoke and fire [6]. In some cases, the simulated quantities are stored as textures in order to share the processing load [7] in an attempt toward real-time simulations. Chu and Tai [8] integrated both physically based models and hardware programming to simulate two-dimensional ink dispersion in absorbent paper for art creation purposes. Data filters were employed to enable rapid exploration of post-processed data. Examples of such research include selective visualization [9] and feature extraction [10]. Our work attempts to solve this problem using a different approach.

In an effort to generate fairly accurate indoor airflow simulation data in real-time, approximate modeling methods using simplified fluid flow equations and learning algorithms were developed [11]. The models studied the integration of supervised Artificial Neural Networks (ANN) and unsupervised Reinforcement Learning (RL) algorithms. Although initial tests conducted with learning algorithms showed satisfactory results, more research work is necessary to increase the accuracy of predicted results. The tests also demonstrated that merely employing learning algorithms with no association to physical conditions of the thermal environment may lead to erroneous results.

This paper discusses an Adaptive Localization Method that significantly minimizes the simulation domain to achieve close to real-time predictions of indoor environment. As the user interacts with the space by modifying boundary conditions (opening a window, etc.) while being immersed in an Augmented Reality environment, the ALM detects the changes and significantly narrows down the simulation space for instant re-simulation. This localized space is simulated and the newly generated airflow data is updated to corresponding spatial nodes for interactive, immersive AR visualization.

2. ADAPTIVE LOCALIZATION METHOD

The Adaptive Localization Method is an iterative process that detects changes to boundary conditions and constructs individual volumetric zones within the thermal environment that require re-simulation. The advantage of such a method is the significant cutback in simulation time that would permit changes to boundaries to become visible in an immersive environment in real time. ALM consists of three modules – performance typology mapping, adaptive localization, and node connectivity modules, figure 1. While the performance typology mapping module establishes room typologies and boundary conditions of the system, the adaptive localization module identifies nodes that exhibit significant nodal intensity changes that arise from changes to boundary conditions. These nodes are connected together to form a tree structure in the node connectivity module. The tree structure is enclosed in a bounding box that represents the volumetric space that needs to be investigated in detail. This relationship between changes to performance values and bounding box shape can be learned. The reduced volumetric space can be simulated in real time and updated accordingly.

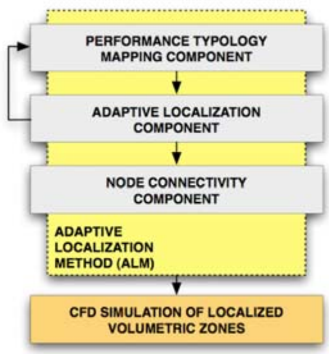


Figure 1: Adaptive Localization Method.

2.1 Performance Typology Mapping Module

Performance typology refers to a unique set of room geometries and the position / size of openings that exhibit notably different thermal behaviors when simulated. A performance typology library was generated based on a series of simulations such that one of the entities matches any given room geometry. Currently, the typology includes orthogonal room geometry with one to two openings located on the walls. Four conditions were identified for which the typology was developed – "single opening single wall," "multiple openings single wall," "multiple openings adjacent walls," and "multiple openings opposite walls," table 1. The performance typology is characterized by geometry and performance variables, table 2. The room geometry is defined by length, width and height of the room; the openings (window, door) are represented by width, height, and the opening's distance from adjacent wall and floor. The performance variables include temperature and velocity (magnitude and direction) of air entering the room through the opening. Boundary and initial conditions form the basis of setting up computational simulation for modeling energy mass flow of a system. The internal cell zone consists of the fluid (air) and the wall forms as the internal surface.

Table 1: Performance Typology Library (plan view).

Description

Single opening
Single wall

Multiple openings
Single wall

Multiple openings
Adjacent walls

Multiple openings
Opposite walls

Table 2: Geometry and Performance Variables.

	Variables	Parameters
<u>Geometry</u>		
Room		
Room Length		R_L
Room Width		R_W
Room Height		R_H
Opening		
Opening Location		O_X
Opening Left distance	O_{Lmax} , O_{Lmin}	O_L
Opening Right distance	O_{Rmax} , O_{Rmin}	O_R
Opening Floor distance	O_{Fmax} , O_{Fmin}	O_F
Opening Ceiling distance	O_{Cmax} , O_{Cmin}	O_C
<u>Performance</u>		
Opening Temperature	O_{Tmax} , O_{Tmin}	O_T
Opening Velocity	O_{Vmax} , O_{Vmin}	O_V

2.2 Adaptive Localization Module

The adaptive localization component is an iterative approach facilitating detection of nodes that significantly change as the boundary conditions are altered. These nodes are referred to as "change nodes." As a room is initialized with geometry and performance values in the performance typology mapping module, it is prepared for CFD simulation using an automated process, figure 2.

The room geometry is meshed to create a three-dimensional structured grid that contains ordered set of orthogonal lines. The intersection of the grids corresponds to nodes. Gambit software is used to create a mesh grid of the geometry. The mesh grid is imported to Fluent software for CFD computation. Boundary and initial parameters are set based on the entity selected from the performance typology. For CFD simulation of performance typology, the openings act as "velocity inlets" with air temperature that ranges from 60°F to 72°F and air velocity that ranges from 0.5m/sec to 2.0m/sec. The opening temperature and velocity are increased at 2°F and 0.5 m/sec intervals respectively. CFD simulation is continued until it converges to a solution. The resultant thermal dataset for the entity being investigated is stored for comparison with the next CFD run.

As the objective of adaptive localization is to identify "change nodes," the performance variables (temperature and velocity) are perturbed in a linear fashion until there are no substantial "change nodes" present in the model. For every change to the boundary, the CFD results are compared to previous thermal data to establish "change nodes" for that particular performance type.

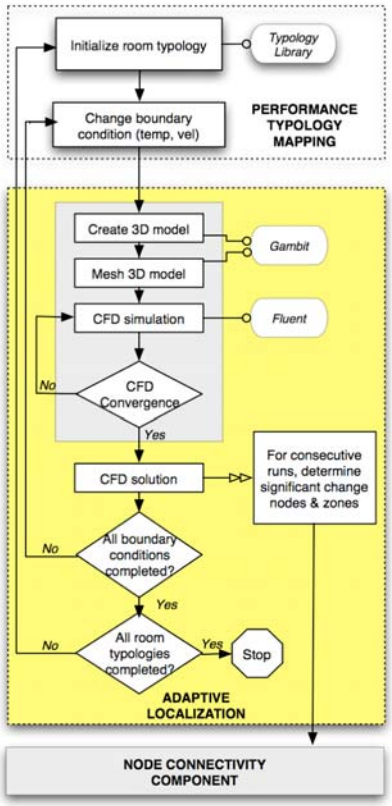


Figure 2: Automated Adaptive Localization Procedure.

If Data1 and Data2 are two datasets corresponding to changes in performance values, equations (1) to (4) compute temperature difference, velocity (magnitude) difference, and resultant polar and azimuth angles of velocity vector.

$$Data1:\{x_1, y_1, z_1, u_1, v_1, w_1, t_1\}$$
$$Data2:\{x_2, y_2, z_v, u_2, v_2, w_2, t_2\}$$

Temperature difference:

$$t_{diff} = |t_1 - t_2| \tag{1}$$

Velocity (magnitude) difference:

$$v_{dif\_mag} = \sqrt{(u_1 - u_2)^2 + (v_1 - v_2)^2 + (w_1 - w_2)^2} \tag{2}$$

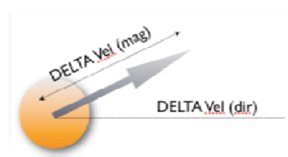
Resultant polar angle:

$$v_{diff}(\theta) = \tan^{-1}\frac{(v_1 - v_2)}{(u_1 - u_2)} \tag{3}$$

Resultant azimuth angle*:*

$$v_{diff}(\phi) = \cos^{-1}\frac{(w_1 - w_2)}{v_{dif\_mag}} \tag{4}$$

The "change nodes" identified relate to temperature and velocity (magnitude). The process is executed until all entities and changes to boundary conditions are studied. In order to understand the behavior of "change nodes" and to identify lower and upper bounds such that only significant changes are used for comparison purposes, parametric studies and sensitivity analyses were conducted. In order to visualize "change nodes," a glyph representation was developed, figure 3. The glyph is comprised of a sphere and an arrow. While the sphere radius is the summation of temperature and velocity differences, the arrow length and direction are computed from velocity difference and direction respectively. Figure 4 shows "change nodes" with a lower (2.5) and upper (6.0) temperature limit and velocity lower (0.02) and upper (0.07) limits for a new test case with one "velocity inlet." The datasets used differ in input performance values. This is evident from the location of high "change node" intensities near the velocity inlet of the room geometry.



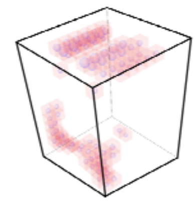


Figure 3: (Left) Glyph representation of "change nodes."
Figure 4: (Right) "Change nodes" and room geometry.

Currently, "change nodes" are detected using temperature and velocity differentials. A few alterations to the procedure will allow combinations of sensitive variables to be perturbed, normalized and employed to narrow down such nodes for robust detection. These nodes only need to be updated rather than the entire spatial system, thereby reducing simulation time exponentially.

2.3 Node Connectivity Module

The node connectivity module allows "change nodes" to be grouped collectively based on previously set intensity parameters using a "tree growth" procedure. These nodes are then bound within a three-dimensional orthogonal enclosure or a bounding box. The enclosure is the reduced space that requires further simulation.

To assemble nodes together that share similar criteria, the "tree growth" procedure is employed. It uses Breadth First Search (BFS) algorithm to search the space for nodes that meet the nodal intensity criteria. BFS has been used in Prim's minimum-spanning tree algorithm and Dijkstra's single source shortest-path algorithms [12]. BFS expands the search space between discovered and undiscovered vertices across the breadth of the frontier. The algorithm discovers all nodes at distance "k" before discovering any nodes at distance "k+1". Since the BFS algorithm searches for nodes at a distance of "k+1", the room spatial coordinates are normalized such that nodal distance equals "1". Following normalization, the change node with largest value is identified. BFS algorithm begins its search breadth-wide and connects to form a tree structure until the search criteria are met, figure 5. Once the tree structure is defined, the nodes are enclosed within a bounding box by computing min-max values along the three axes, figure 6.

Although the "tree growth" procedure detects multiple nodal tree structures, only two bounding boxes are drawn that correspond to the two most significant spaces in the room that require re-simulation. Yet, there are instances where either two nodal tree structures or bounding boxes are sufficiently close to form one entity. Currently, "trees group" criteria are under development to determine if two or more nodal tree structures can be combined to form a single bounding box.

```
For all nodes n {
        //Identify Tree_n Parent node
        (max) Intensity_node = Tree_1 Parent node (i,j,k);
        From Tree_1 Parent node (i,j,k){
                //Breadth First Search algorithm
                Check   (i,j,k+1),(i,j,k)
                        (i,j+1,k),(i,j-1,k)
                        (i+1,j,k),(i-1,j,k)
                //Criteria to connect nodes
                (min) < Intensity_node<(max)
                Length between nodes == 1
                Complete Tree_1;
                //Calculate min-max of nodal axes
                }
        Complete Tree_n
        }
```

Figure 5: Code: "Tree Growth" procedure using BFS algorithm.

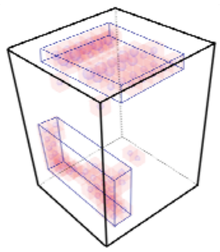


Figure 6: Bounding Box enclosing tree structure.

Thus, as ALM iteration proceeds until all variables are exhausted, the input geometry and boundary conditions are mapped to newly developed bounding boxes. Any change in the boundary conditions ("cause") results in new bounding boxes ("effect") which vary in shape and position. Tests show that changes to input conditions and shape of the bounding box are linear. For example, as the velocity increases, the bounding box morphs in linear fashion. This linear relationship is used to generalize the bounding box characteristics which then can be employed instantly for any boundary change from initial conditions. Thus, ALM allows narrowing down the simulation space exponentially for CFD simulation rather than requiring elaborate simulation of the entire system. The newly generated airflow data is updated to corresponding spatial nodes for interactive, immersive AR visualization.

3. INTEGRATION TO AR SYSTEM

The multimodal AR system tracks the user's movement in real time and poses graphical representations of CFD simulation datasets on the Head Mounted Device (HMD) for the user to visualize and interact with. The multimodal Human Computer Interaction (HCI) enables efficient data manipulation by users while still being immersed in the visualization of CFD datasets, in actual space. It consists of a library of speech and gesture recognition tasks that aid in data manipulation. IBM Viavoice was employed for speech recognition. The gesture recognition system captured global hand motion using trackers attached to the glove and local finger motion as a set of joint angles. Using custom-prepared functions, the hand posture data was transformed into commands that allowed data manipulation. [1] provides detailed information of the multimodal immersive AR system.

4. CONCLUSION

The paper discussed an Adaptive Localization Method to minimize simulation space exponentially. It presented a "tree growth" procedure using BFS algorithm to connect "change nodes" together to form a bounding box. CFD simulation is conducted for this volumetric space and the resultant data is updated to the corresponding nodes of the room geometry instantly. Thus, ALM allows users to interact with space and visualize airflow changes instantly. Currently, ALM permits localization for one set of performance typology that comprises of up to two openings. The performance variables studied include temperature and velocity (magnitude and velocity) of air passing through the openings. Although present tests demonstrate the potential of using ALM to narrow down simulation space, a robust learning methodology is required to map the changes to boundary conditions to the shape of the bounding box. Such a learning methodology will permit rapid generation of bounding boxes as the input conditions change. Moreover, the present study involves one performance typology. Further study with more performance typologies including supply air inlet and outlet need to be developed.

5. REFERENCES

[1] Malkawi AM, RS Srinivasan [2005]. A new paradigm for Human-Building Interaction: The use of CFD and Augmented Reality, *Automation in Construction Journal 14*(1):71-84.

[2] Foster N and D Metaxas [1996]. Realistic Animation of Liquids, *Graphical Models and Image Processing Journal*, 58(5):471-483.

[3] Chen Q and W Xu [1998]. A Zero-equation turbulence model for indoor airflow simulation, Energy & Buildings Journal 28:137-144.

[4] Wei X, W Li, K Mueller, A Kaufman [2002]. Simulating fire with texture splats, Proceedings IEEE Visualization.

[5] Foster N, D Metaxas [1997]. Modeling the motion of a hot turbulent gas, Proceedings SIGGRAPH, 181-188.

[6] Stam J [2000]. Interacting with smoke and fire in real time, Communications of the ACM 43(7):76-83.

[7] Harris MJ [2004]. Fast fluid dynamics simulation on the GPU. In: GPU Gems: Addison Wesley.

[8] Chu NSH, CL Tai [2005]. MoXi: Real time ink dispersion in absorbent paper, Proceedings of ACM SIGGRAPH.

[9] Walsum T [1995]. Selective visualization on curvilinear grids, PhD thesis, Delft University of Technology, The Netherlands.

[10] Kenwright D, C Henze, C Levit [1999]. Feature extraction of separation and attachment lines, IEEE Transactions 5(2):135-144.

[11] Srinivasan RS, AM Malkawi [2005]. Real time simulations using learning algorithms for immersive data visualization in buildings, International Journal of Architectural Computing 3(3).

[12] Cormen TH, Stein C, Rivest RL, Leiserson CE [2001]. Introduction to Algorithms, McGraw-Hill.

About the authors

Ravi Srinivasan is a Ph.D. candidate and Research Associate at the T.C. Chan Center for Building Simulation and Energy Studies. His research focuses on real-time immersive simulations for built environments.

Ali Malkawi is Professor at the School of Design, University of Pennsylvania. He is the Director of the T.C. Chan Center for Building Simulation and Energy Studies.