Proceedings of the 2011 Winter Simulation Conference
S. Jain, R. R. Creasey, J. Himmelspach, K. P. White, and M. Fu, eds.

CONFORMAL ADAPTIVE HEXAHEDRAL-DOMINANT MESH GENERATION FOR CFD SIMULATION IN ARCHITECTURAL DESIGN APPLICATIONS

Rui Zhang

Carnegie Mellon University
5000 Forbes Ave., MMCH 415
Pittsburgh, PA 15213, USA

Khee Poh Lam

Carnegie Mellon University
5000 Forbes Ave., MMCH 415
Pittsburgh, PA 15213, USA

Yongjie Zhang

Carnegie Mellon University
5000 Forbes Ave., Scaife Hall 303
Pittsburgh, PA 15213, USA

ABSTRACT

Mesh generation is a critical and probably the most manually intensive step in CFD simulations in the architectural domain. One essential feature is the large span of dimensional scales that is encountered in design, particularly if the model aims to simulate indoor and outdoor conditions concurrently, e.g., site at the magnitude of kilometers while building elements at the magnitude of centimeters. In addressing the challenge this paper presents an approach to generate adaptive hexahedral-dominate meshes for CFD simulations in sustainable architectural design applications. Uniform all-hexahedral meshes and adaptive hexahedral-dominant meshes are both generated for natural ventilation simulation of a proposed retrofit building in Philadelphia. Simulation results show that adaptive hexahedral-dominate meshes generate very similar results of air change rate in the space due to natural ventilation, compared to all-hexahedral meshes yet with up to 90% reduction in number of elements in the domain, hence improve computation efficiency.

1 INTRODUCTION

Computational Fluid Dynamics (CFD) as a building thermal simulation tool has been demonstrated by researchers and industry practitioners to be powerful in studying the detailed thermal processes in the buildings for energy efficient design strategies. It is also an important tool for visualizing the virtual thermal performance of the proposed design strategies. However, the CFD simulation has been seen as an expensive tool with steep learning curves. One of the main reasons for this impression is due to difficulties in constructing model geometries from design drawings, and generating the mesh for CFD simulation.

Mesh generation is a critical and probably the most manually intensive step in CFD simulations in the architectural domain. Mesh generation for CFD simulation in buildings poses special challenges. Firstly, the span of the dimensional scales encountered in design is large. For example, the dimension of the building site maybe at the magnitude of kilometers, while the dimensional scale of the building elements can be at the magnitude of centimeters. To deal with this scenario, traditional CFD simulations of buildings tend to be generally divided into two categories: (1) Indoor detailed analysis with most of the building elements relatively accurately preserved. (2) Ambient wind environment simulation around the buildings, with the buildings simulated as solid blocks. Secondly, the geometry model of a building usually involves non-manifold surfaces. Non-manifold surfaces exist when there are partitions, or furniture in the space.

Thirdly, the lack of interoperability between CFD simulation tools and major architectural Computer Aided Design tools makes the application of CFD in architectural an expensive practice.

In addressing the challenges this paper presents an automated approach to generate adaptive hexahedral-dominate meshes from architectural design tools. The objectives of the mesh generation are as follows:

- Provide interoperability with architectural CAD tool, especially at conceptual design phase.
- Generate adaptive hexahedral-dominated mesh with reduced number of elements at finer resolution.
- Produce info-rich geometrical representations, whereas all the surfaces are well preserved and each piece of the surfaces will be individually tagged for possible CFD boundary condition settings.

This paper will present the mesh generation algorithm designed to the objectives. The mesh generation algorithm is applied to a live retrofit building project in Philadelphia. Both uniform all-hexahedral mesh and adaptive hexahedral-dominant mesh are generated. Mesh quality is reported in terms of aspect ratio, face squish index and volume squish index. Simulation results of air change rate of the building are compared with uniform all-hexahedral meshes and adaptive hexahedral-dominant meshes at different resolutions. Simulation results with different sizes of the simulation domain are also computed and compared.

2 PREVIOUS WORKS ON MESH GENERATION

2.1 Adaptive Mesh Generation

Mesh adaptation can be applied in the space domain as well as the time domain. A detailed description of the mesh adaptation in time evolving domains can be found in (Baker 2001). This paper will focus on the adaptation in the space domain. The mesh adaptation provides the advantage of increasing resolution while reducing the number of elements in the simulation domain. The criterion for mesh adaptation can be based on a metric-field derived from priori or posteriori error estimates for each element in the domain, or based on the complexity of the geometry forms in the domain, where meshes will be finer on complicated geometry features or where the gradients of the solution variables are high. The strategies to generate adaptive quadrilateral/hexahedral meshes fall largely into two categories: no-grid based and grid based.

2.1.1 No-grid Based Methods

The no-grid based methods usually either (1) start with fine mesh and then merge small elements to form large elements where desired, or (2) start with a coarse mesh and refine the elements to form smaller elements on local features. Borouchaki and Frey (1998) presented a method to start with fine triangular mesh and merge the triangles to form larger quadrilaterals. Woodbury, Shepherd, Staten, and Benzley (2011) presented other method that starting with hexahedral mesh and apply localized coarsening method to generated local adaptive conforming all-hexahedral meshes. The refinement methods start with a coarse mesh and either (1) apply a set of templates on the coarse elements to generate finer elements (Schneiders 1996), (Schneiders, Schindler, and Weiler 1996), (Tchon, Hirsch, and Schneiders 1997), (Tchon, Dompierre, and Camarero 2002), (Zhang and Bajaj 2006), and (Edgel 2010), or (2) conduct plane insertion in areas where denser elements are needed according to the sizing metric field or on some specific geometry features such as complicated corners (Harris, Benzley, and Owen 2004).

2.1.2 Grid Based Methods

The grid based methods use the octree (quadtree in 2-dimension) data structure to recursively subdivide the domain into eight octants to form the mesh or use the octree grid as a background mesh. The hierarchical nature of the octree structure provides the ability to generate larger elements in smoother areas and continue to subdivide the elements to form smaller elements near complex geometries. However, the background mesh generated by an octree usually is bounded by stair-step surfaces. Techniques have been developed to either fill in the gaps between the stair-steps and the actual boundaries (Schneiders 1996) or projecting the bounding vertices to the actual boundary surfaces (Tchon, Hirsch, and Schneiders 1997). When an adaptive mesh is generated with the octree data structure, usually the tree leaf node level difference is

limited to be less than and equal to 1 so as to limit the size variation between neighbouring elements to be less than and equal to 2. However, as a result of the octree level difference between neighboring elements, hanging nodes, which are vertices hanging in the middle of neighboring face or edge, are produced at the transitional areas. The dual-contouring method (Ju, Losasso, Schaefer, and Warren 2002) was extended to generate conformal meshes without any hanging nodes (Zhang and Bajaj 2006), (Zhang, Bajaj, and Xu 2009), (Marechal 2009). The algorithm presented in this paper also uses an octree based dual meshing method, which is described in detail in section 4 below.

2.2 Hexahedral Mesh Generation

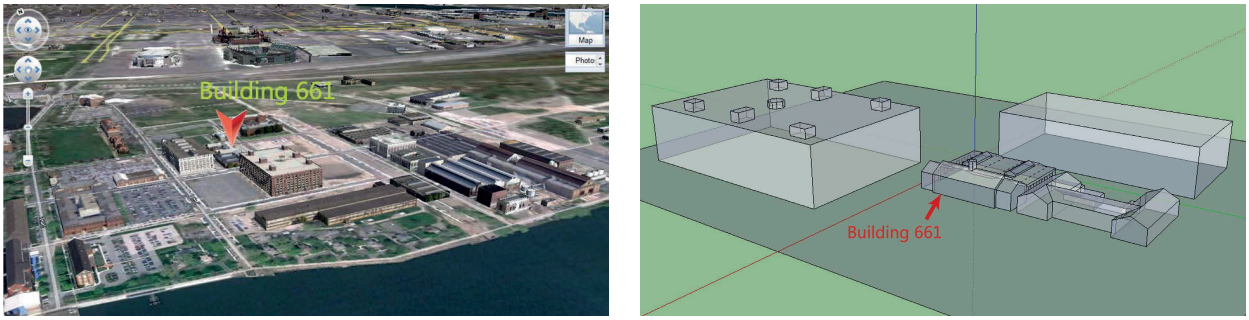
Hefny and Ooka (2008) showed that the relative error of the simulation results from hexahedral meshes is smaller than that from tetrahedral meshes, prism meshes and hybrid meshes. Benzley, Perry, Merkley, Clark, and Sjaardema (1995) also found that all linear tetrahedra produce significant higher errors than hexahedra in various finite element analyses. Several hexahedral-dominant or all-hexahedral mesh generation algorithms have been developed. Although tetrahedral meshes are more robust in terms of filling in complicated geometries, the advantages and challenges of the hexahedral mesh generation have driven great interests from the mesh generation researchers all over the world. The research team in the Sandia National Laboratory has implemented several hexahedral mesh generation algorithms in the CUBIT software including paving (Blacker and Stephenson 1991), mapping (Tautges, Liu, Lu, Kraftcheck, and Gadh 1997), plastering (Staten, Owen, and Blacker 2005) and special purpose primitives. Other methods in generating hexahedral meshes have also been created: medial axis (Ito, Shih, Koomullil, and Soni 2006), grid based ((Zhang, Bajaj, and Xu 2009) and (Marechal 2009)) and Whisker Weaving (Ito, Shih, Koomullil, and Soni 2007). The algorithm in this paper will generate uniform all-hexahedral and adaptive hexahedral-dominant mesh.

3 GEOMETRY MODEL CONSTRUCTION FOR SIMULATION DOMAIN

One significant obstacle for current CFD tools to be effectively utilized in the architectural iterative and adaptive design process, is the difficulty in geometry model construction. Hence, CFD has remained an expensive tool mainly for validating purposes. It is argued that the benefits of CFD would not be fully released unless it is deployed early in the conceptual design stage, when information provided by CFD simulation can give insights for a specific design strategy and may enlighten potential improvement. Currently the model construction and mesh generation have remained the bottle neck in the architectural domain applications, since it is generally lacking interoperability between CFD tools and current architectural concept design tools. Extracting a model construction from a CAD tool for mesh generation should be reasonably automated to support the iterative process of design.

In order to address the challenges, the mesh generation process presented in this paper will take as input a format that can be generated from architectural conceptual design tools. The input file format is the VRML format, that can be exported by most three-dimensional modelling tools. The tool used in this research is the Google SketchUp. Along with Google Earth and Building Maker, there is a large repository of the three dimensional SketchUp models of existing buildings all over the world shared by people, mostly available for free online. The building under design can be placed in the geographical location (latitude and longitude) and the terrain data can also be obtained in the SketchUp model. The existing building models and terrain data are critical for simulation of the air flow around and through the building.

The test case building and its surrounding environment is shown in Fig. 1. A rectangle surface at the ground plane of the model is added to cover the foot print of the site, from which the meshing algorithm will calculate the root octant to circumscribe the whole site. The defined names of the surfaces from SketchUp will be maintained in the meshing algorithm.



(a) (b)

Figure 1: (a) The SketchUp models of Building 661 and the surrounding buildings on Google Earth. (b) The models of Building 661 and immediate surrounding buildings in SketchUp.

4 MESH GENERATION

4.1 Octree Construction

Octree is a tree data structure that can be used to partition a three dimensional space by recursively subdividing it into eight octants (children). As shown in Fig. 2(a), the domain can be recursively subdivided to generate small octants near geometry features, or to be subdivided according to a predefined sizing function. The tree will stop collapsing if there is no feature in the octant, where a feature is a geometry object such as a node or a surface. If there are more than two nodes or more than two intersection lines of surfaces in the octant, the octant will be collapsed. The tree will stop until each leaf node has no more than one feature. Octant without any children octant is called a leaf node, and the level of the octant is the number of parents preceding it. The steps to construct the octree are as follows.

4.1.1 Octree Leaf Node Level Selection:

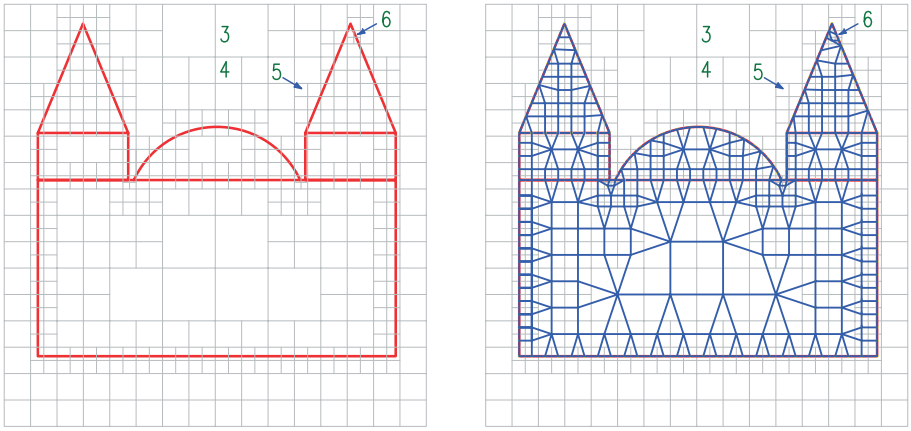
A root octant is first computed to circumscribe the whole domain, and then take three level of resolutions:

- *sizeMax*: the element size at far field from the geometry features.
- *sizeMid*: the element size at simple geometry features, e.g., planar surface.
- *sizeMin*: the element size near complicated geometry features, e.g., intersection of two surfaces.

Three octree leaf node levels will be decided corresponding to the three resolutions: top_level, $medium\_level$ and $bottom\_level$. The *sizeMin* will be the octant size at the $bottom\_level$. The octant size is multiplied by 2 each time, the octant steps up a level. By multiplying several times, the $medium\_level$ is found, that whose octant size is closest to the *sizeMid*. The same method is used to find the top_level.

4.1.2 Octree Starting Level Selection:

There are two methods to construct the octree. One method is to start from the root to collapse the tree nodes to its appropriate level. The other is to start constructing the octree from a $starting\_level$, which is smaller than the calculated top_level. The computed top_level is dependent on the ratio between the size of the domain and the desired largest element size. For applications in the architectural domain the top_level is usually around $4 \sim 7$. There are two time consuming procedures in the octree collapsing process. One is to check if the cell is in the simulation domain (referred as *inZoneCheck* here after), the other is to check if there are geometry features in this cell (referred as $featureCheck$ here after). Let n be the number of input geometry surfaces. The time complexity of the *inZoneCheck* is $O(n)$, and the time complexity of the $featureCheck$ is also $O(n)$. Due to the octree structure, a child node will only inherit the features from its parent node, thus the time complexity of *inZoneCheck* and $featureCheck$ for child nodes will become constant $O(1)$. At the starting level, each cell will have to perform the *inZoneCheck* and the $featureCheck$ procedure. The time complexity of processing the starting level is $O(n \times 2^{3 \times level})$.



(a) (b)

Figure 2: (a) The 2D demonstration of quadtree (the 2D analogue of octree, recursively subdividing a quad into four quads.) grid generated around the outline of a building model. The larger quads near the center of the building is at level 3, and quads near the surface of the outline is at level 5. There are also some quads at level 6 (near the two tips of the triangular shape, and near the corner where the rectangle and the arcs approaching each other), which are the smallest quads near complicated geometry, in order to preserve the features. (b) The dual mesh generated based on the quadtree grid.

Therefore, the time complexity increases exponentially as the starting level increases. It is found through experiential exercise that the starting level of 4 works efficiently for most geometry models. The algorithm will start collapsing the octree from $level = 4$. All octant cell will be collapsed till all the octant cells are at top_level, afterwards the decision whether to collapse the octant cell is based on whether there is any feature in the octant cell or the type of feature in the cell.

4.1.3 Leaf Node Generation Criterion:

The tree will start collapsing from the $start\_level = 4$, each cell inside the simulation domain will be collapsed until each node is at the top_level. To further collapse the tree, nodes with feature will be further collapsed to mid_level on simple geometries and collapsed to $bottom\_level$ near complicated geometries. The meshing algorithm will also balance the tree by limiting the level differences between two neighboring octants to be no more than one. Therefore, the small octants will propagate gradually into the space until the maximum level difference criterion is met.

4.2 Dual Mesh Generation

4.2.1 Vertex Calculation

When the octree is fully collapsed and balanced, the dual mesh generation will find a vertex for each of the leaf octant cell. The methods to calculate the vertex are as follows:

- If there is no feature in the cell, the center of the octant will be taken as the vertex for the cell.
- If there is a geometry node (end point of an edge), the node will be taken as the vertex for the cell.
- If there is one geometry surface, all the intersection points between the octant and the geometry surface will be computed, and the gravity center of all the points will be projected to the surface. The projected point will be taken as the vertex for the cell, as shown in Fig. 3.
- If there is one shared intersection line of all the surfaces in this cell, the center of the intersection line will be taken as the vertex for the cell.

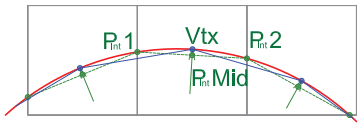
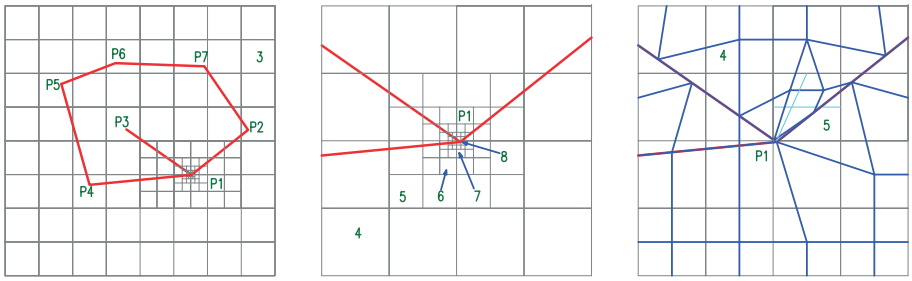


Figure 3: The vertex calculation for dual mesh generation. For each grid cell (in grey color), the intersection points between the grid and geometry surface are calculated ($P_{int}1$ and $P_{int}2$), then the gravity center of all the intersection points is calculated ($P_{int}Mid$). Finally, the vertex (Vtx) is found by projecting the gravity center to the geometry surface along its normal vector.

4.2.2 Modified Grid Method - Special Treatment for Cells with Two Disconnected Surfaces

In theory, the octree collapsing process will stop when each cell has no more than one geometry surface or one intersection line. However in some situation in order to meet the criterion the level of the leaf nodes will be very deep, as shown in Fig. 4. The normal octree will have to collapse to Level 8 in order to have one feature in each cell. A grid modification method is designed to reduce the level depth and maintain the geometry features. The modified grid method will virtually move the grid so that each cell will have one feature at a smaller collapse level as shown in Fig. 4. The modified grid method maintained the topological relationship between each cell, so that the geometry forms are preserved.



(a) (b) (c)

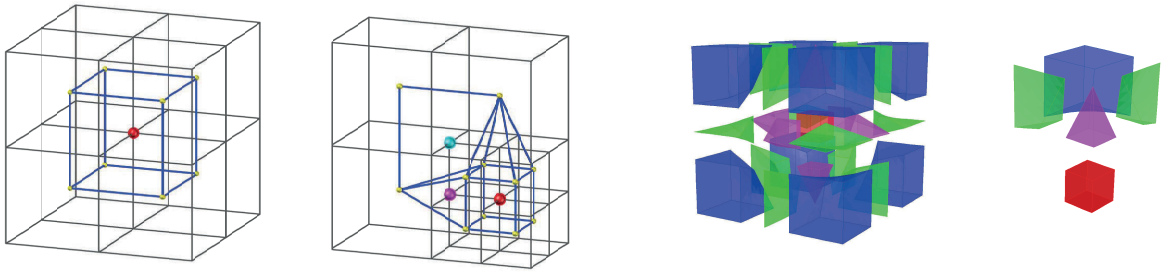
Figure 4: The 2D demonstration of the modified grid method (a) The fully collapsed octree grid for a non-manifold geometry model. (b) Zoom-in view of the fully collapsed grid near point P1. The point P1 is the joint between three lines. In order to meet the criterion that each cell has no more than one feature, the collapsing level needs to be 8, which is much deeper than the other cells. (c) Zoom-in view of the dual mesh after the modified grid method is deployed, the collapsing level is one level deeper than the other cells (Level 5 in this case).

4.2.3 Element Generation

For each grid point in the octree there are at most 8 cells sharing the grid point. For the uniform octree, each grid point is shared by eight octant cells as shown in Fig. 5(a). For non-uniform octree, a grid point at the connection between leaf nodes at different levels will be shared by less than eight octant cells as shown in Fig. 5(b). An element is formed by vertices from each of the octant cells sharing the same grid point. For example, in uniform octree, a hexahedron is formed by the eight vertices in each of the eight octant cells as shown in Fig. 5. Fig. 5(b) shows the elements formed at the connection of two levels, where the number of cells sharing the same grid point is less than 8, thus forming pyramids, wedges and polyhedra.

4.3 Mesh Quality

Aspect ratio, face squish index and volume squish index are computed, which are listed in the CFD solver Fluent (Fluent 2007) as important mesh quality metrics. The definitions of these metrics follow the Fluent documentation. The aspect ratio is the smallest ratio of the following distances: distance between element centriod and face centroids, distance between element centriod and nodes. For a unit hexahedral element,



(a) (b) (c) (d)

Figure 5: (a)&(b): The 3D demonstration of the construction of one element and several elements at the connecting octree levels. (a) One hexahedral element by the vertices in the eight octant cells sharing the red grid point. (b) One hexahedral element is formed by the vertices in the eight octant cells sharing the red grid point. One pyramid is formed by the five octant cells (one at the parent level and the other four at the child level) sharing the purple grid point. A polyhedron is formed by five vertices in the octant cells sharing the cyan grid point. (c)&(d): The 3D exploded view of the types of elements when one octree level is connected with a parent/child level. The elements formed when one octant is collapsed in the grid. The elements are formed by eight octants at the parent level, and eight octants at the child level. There are 9 hexahedra (blue color and red color), 6 pyramids (pink color), and 12 polyhedra (green color).

the maximum distance is 0.866 and the minimum distance is 0.5. Therefore, the aspect ratio is 0.57, which will be the optimum aspect ratio for a hexahedral element. The volume squish index is the dot product of each vector pointing from the centroid of a cell toward the center of each of its faces. Thus, the worst elements will have a squish index close to 1, good elements will have smaller squish indices tending towards 0. The face squish index is the dot products of each face area vector, and the vector that connects the centroids of the two adjacent cells. Thus the worst elements will have a squish index close to 1, while good elements will have smaller squish indices tending towards 0.

5 APPLICATION EXAMPLES

5.1 Sphere in A Cube

The meshing algorithm is first applied to a sphere in the cube model. Meshes with three levels of resolution are generated, as shown in Fig. 6. The dimension of the cube is $10 \times 10 \times 10(m)$, and the radius of the sphere is $2m$. The three levels of resolution are $0.2m$, $0.1m$ and $0.05m$. The number of vertices and elements generated for the three sets of mesh are summarized in Table 1. Three sets of all-tetrahedral mesh with resolutions of $0.2m$, $0.1m$ and $0.05m$ are also generated using a commercially available meshing software deploying the Delaunay Triangulation algorithm (Shewchuk 2001). The number of vertices and elements in each set of the meshes are summarized in Table 1. As can be seen for the same mesh resolution, the hexahedral meshes contain much fewer number of elements (as low as 14%), and the adaptive hexahedral-dominant meshes produce even fewer number of elements (as low as 7%), compared to that of the all-tetrahedral meshes.

5.2 Building 661, Navy Yard, Philadelphia

GPIC (Greater Philadelphia Innovation Cluster) is one of the three DOE Innovation Hubs, which are to pursue transformative breakthroughs in technology that can help us meet our energy and climate challenges. The GPIC project focuses on developing innovative energy building technologies, designs, and systems. One initial task to demonstrate such developments is the retrofit of Building 661 to serve as the GPIC center. Natural ventilation is regarded as one of the important features of green design. The meshing algorithm will be used for the CFD simulation of natural ventilation conditions in Building 661. First,

Table 1: A comparison of the number of vertices and elements between three mesh types at three levels of resolution on the same geometry model of the sphere in the cube.

Type of Mesh	Resolution (m)	Number of Vertices	%	Number of Elements	%
Adaptive Hexahedral-Dominant	0.2	32,435	4%	31,161	24%
	0.1	132,443	3%	129,207	13%
	0.05	540,140	1%	533,562	7%
Uniform All-Hexahedral	0.2	129,371	85%	120,837	14%
	0.1	1,000,984	110%	966,536	18%
	0.05	7,869,492	97%	7,731,934	16%
Uniform All-Tetrahedral	0.2	152,116	–	857,481	–
	0.1	913,406	–	5,261,119	–
	0.05	8,150,512	–	47,324,568	–

Note: All percentage values are the percentage of vertices or elements of the tetrahedral meshes with the same resolution.

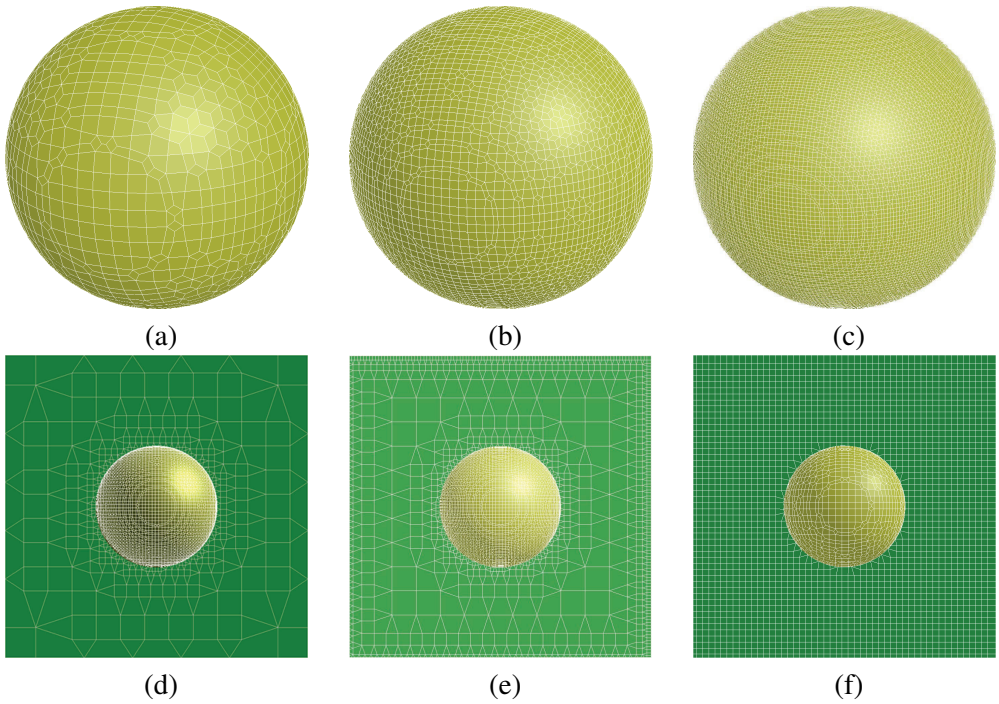


Figure 6: The adaptive mesh generated for a sphere in the cube model. The dimensions of the cube is $10 \times 10 \times 10(m)$, and the radius of the sphere is $2m$. (a)∼(c) The surface mesh with the smallest resolution of $0.2m$, $0.1m$ and $0.05m$. (d)&(e) The surface mesh and the mesh of a plane cutting through the center of the model for mesh with the resolution of $0.1m$. (f) The all-hexahedral mesh with resolution of $0.1m$.

both uniform all-hexahedral mesh (referred as uniform mesh here after) with resolution of $1m$ and adaptive hexahedral-dominant mesh (referred to as adaptive mesh here after) with different resolutions for a domain with Building 661 and one immediate adjacent building are generated. The simulation results in terms of air change rates in the space are compared. Then, both the uniform mesh and the adaptive meshes are generated with two larger levels of domain size, in order to test the scalability of the meshing algorithm on large model with dimension at kilometer level, and at the same time, investigate the effect of simulation domain sizes on natural ventilation simulations. The generated uniform and adaptive meshes are shown in Fig. 7, and the quality mesh metrics are summarized in Table 2.

5.2.1 Boundary Conditions for Natural Ventilation Simulation

The natural ventilation assumed deployable conditions are conditions when the outdoor dry bulb temperature is between $18°C \sim 26°C$. The hourly Typical Meteorological Year (TMY3) (Wilcox and Marion 2008) weather data for Philadelphia is used to extract the hours that natural ventilation is deployable. It is found that there are 2232 hours (25% of the 8760 hours of a year) when the out door dry bulb temperature is suitable for natural ventilation. The wind rose of these extracted hours is shown in Fig. 8. The south west prevailing wind direction is used as incoming wind direction in the natural ventilation simulation. The average wind speed of $4m/s$ is the incoming wind speed. The left (west), right (east) and upper (sky) surfaces of the domain is set as symmetry surfaces. The north domain surface is set as free outflow.

5.2.2 Mesh Scalability Analysis

A uniform mesh is generated with the resolution of $1m$, resulting in $2,777,733$ vertices and $2,852,107$ elements. Seven sets of adaptive meshes are generated with different resolutions. The small element resolutions range from $0.4m \sim 1m$, and the large element resolutions range from $1.6m \sim 16m$. The generated meshes are shown in Fig. 7, and the mesh qualities are summarized in Table 2.

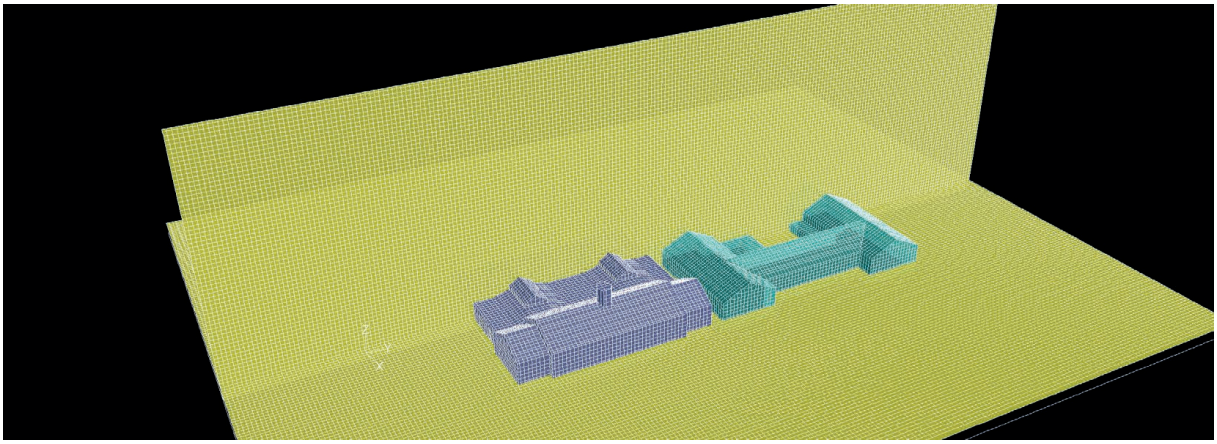
For natural ventilation simulation, the air change rate is one of the important variables. Let F denote the total volume flow rate of air that going into the building through all the openings, with the unit of m^3/s. Let V be the total volume of the building indoor space, with the unit of m^3. Then the Air Change rate per Hour (ACH) is defined as $ACH = F/V \times 3600$, with the unit of $1/hr$. In this study the ACH is used as a metric to compare simulation results with the uniform mesh and the adaptive mesh at different resolutions.

The ACHs and the number of elements, vertices are summarized in Table 3. As is shown, the resulting ACHs from adaptive meshes are very close to that from the uniform mesh. However, the number of vertices and number of elements are greatly reduced. When the largest element in the adaptive mesh is less than $4m$, the resulting ACH is within 0.5% difference compared to that of the uniform mesh, and the number of vertices and elements is about 4% of that of the uniform mesh.

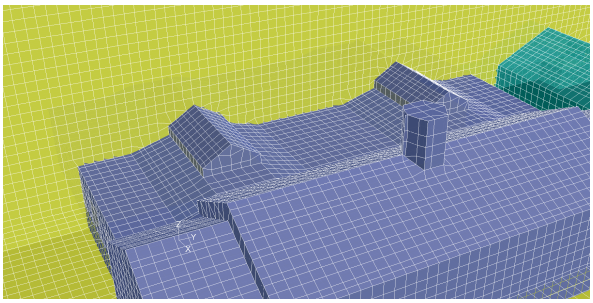
To further reduce the smallest element size in the adaptive mesh to as low as 0.4, the number of vertices and number of elements are still within 40% of that of the uniform mesh at the resolution of $1m$. There is a change in the resulting ACH from the mesh with a small resolution of $0.4m$ of 5% compared with that of the uniform mesh, which may indicate that the resolution in the uniform mesh is not fine enough to capture the characteristics of the flow. A finer mesh may be needed.

5.2.3 Simulation Domain Size Scalability Analysis

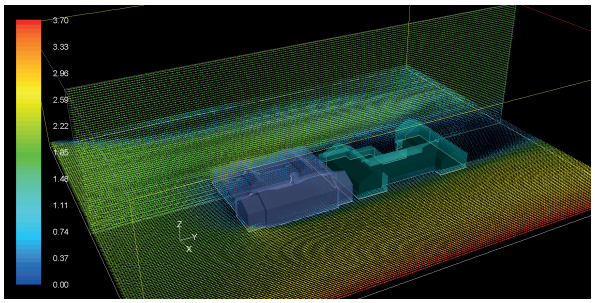
Guidelines for CFD simulation of wind environment at the urban level are not commonly available. The European Cooperation in Science and Technology (COST) provided recommendations based on publish simulation and measurement results (Franke, Hirsch, Jensen, Krus, Schatzmann, Westbury, Miles, Wisse, and Wright 2004). The Architectural Institute of Japan (AIJ) proposed guidelines for CFD simulations based on wind tunnel experiments, field measurements and computation results (Tominaga, Mochida, Yoshie, Kataoka, Nozu, Yoshikawa, and Shirasawa 2008). Both AIJ and COST suggested that the foot print of the building being simulated should not be larger than 3% of the total floor area of the simulation domain.



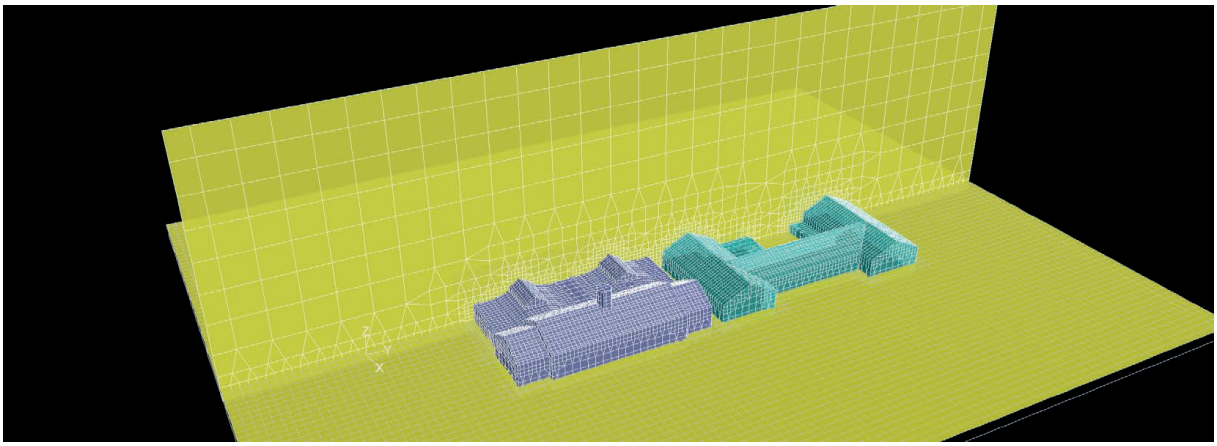
(a)



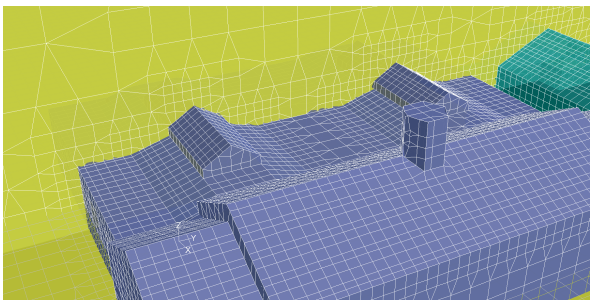
(b)



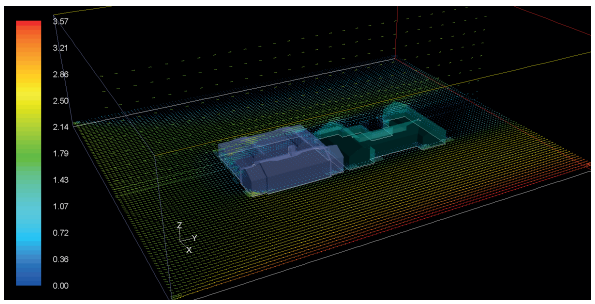
(c)



(d)



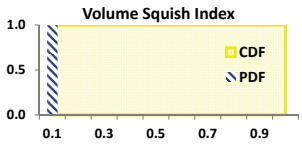
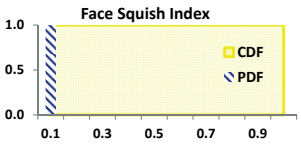
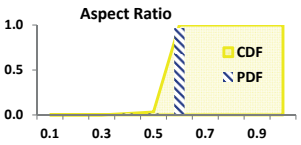
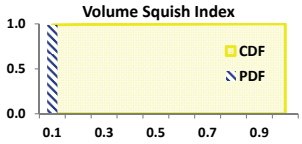
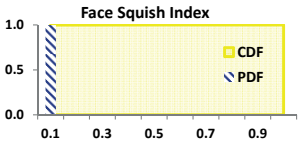
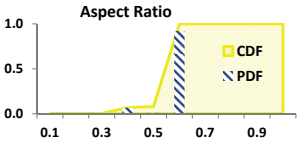
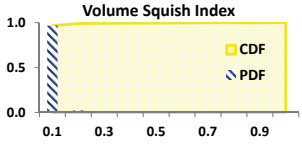
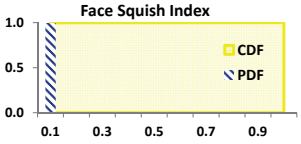
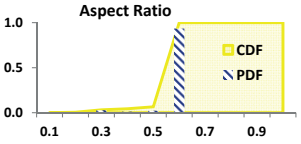
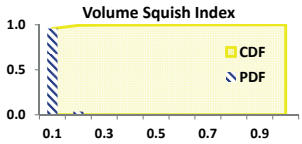
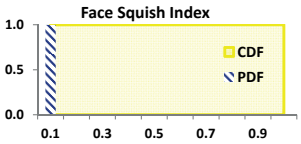
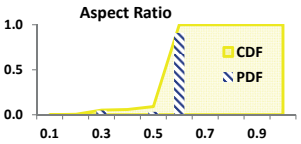
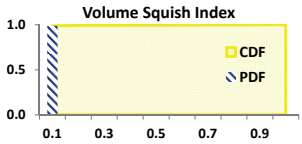
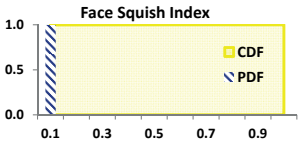
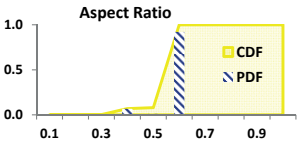
(e)



(f)

Figure 7: The generated uniform mesh $(a) \sim (b)$ with the resolution of $1m$ and the simulation results (c). The generated adaptive mesh $(d) \sim (e)$ with the resolution of $1m \sim 8m$, and simulation results (f).

Table 2: Mesh quality summary of generated meshes.

Mesh	Volume Squish Index (avg, stdDev, max, min)	Face Squish Index (avg, stdDev, max, min)	Aspect Ratio (avg, stdDev, max, min)
Uniform All-Hexahedral			
Building 661, resolution=1m	(0, 1, 0.0007, 0.02)	(0, 1, 0.0002, 0.01)	(0.12, 0.58, 0.57, 0.03)
			
Sphere-in-cube, resolution=0.1m	(0, 0.99, 0.005, 0.05)	(0, 0.99, 0.001, 0.02)	(0.08, 0.58, 0.56, 0.05)
			
Adaptive Hexahedral-Dominant			
Building 661, resolution=1$m \sim 2m$	(0, 1, 0.009, 0.06)	(0, 1, 0.002, 0.03)	(0.08, 058, 0.56, 0.07)
			
Building 661, largest domain, resolution=1$m \sim 8m$	(0, 1, 0.008, 0.05)	(0, 1, 0.001, 0.02)	(0.08, 0.58, 0.55, 0.08)
			
Sphere-in-cube, resolution=0.1m	(0, 0.99, 0.005, 0.05)	(0, 0.99, 0.001, 0.02)	(0.08, 0.58, 0.56, 0.05)
			

Note: X axis in the figures are the values corresponding to each row. Y axis in the figures are the Probability Density Function (PDF) values and the Cumulative Density Function (CDF) values.

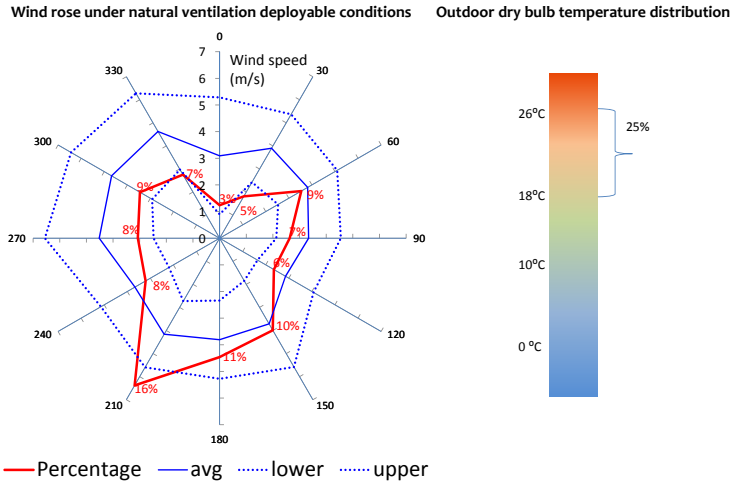


Figure 8: Wind rose when the outdoor dry bulb temperature between the range of $18°C \sim 26°C$. The number of hours outdoor condition falling in this range is 2232 hours (25% out of 8760 hours of the year).

Table 3: A comparison of the simulation results and mesh sizes between the uniform all-hexahedral mesh and the adaptive hexahedral-dominant mesh.

ID	Resolution (m)	Flow In (m^3/s)	Flow Out (m^3/s)	$\%\Delta In\&Out$ (%)	ACH $(1/hr)$	# Vertices	# Elements
Uniform All Hexahedral mesh							
1	1	80.34	74.83	-7%	12.67	2,777,733	2,852,107
Adaptive hexahedral dominant mesh (with the smallest element resolution at $1m$)							
2	$1 \sim 2$	80.34	74.83	-7%	12.68 (0%)	395,148 (14%)	380,991(13%)
3	$1 \sim 4$	79.99	73.99	-8%	12.61 (0%)	112,484 (4%)	108,971 (4%)
4	$1 \sim 8$	79.33	73.58	-7%	12.51 (-1%)	77,453 (3%)	79,736 (3%)
5	$1 \sim 16$	78.61	72.67	-8%	12.39 (-2%)	74,080 (3%)	77,151 (3%)
Adaptive hexahedral dominant mesh (with the smallest element resolution smaller than $1m$)							
6	$0.7 \sim 1.4$	80.17	72.40	-10%	12.64 (0%)	1,098,601 (40%)	1,069,734 (38%)
7	$0.5 \sim 2.0$	79.68	77.29	-3%	12.56 (-1%)	583,469 (21%)	581,100 (20%)
8	$0.4 \sim 1.6$	76.15	71.87	-6%	12.01 (-5%)	985,620 (35%)	981,572 (34%)

Note: the percentage values in the columns of # Vertices, # Elements and ACH are the percentage of values compared with the values for uniform all-hexahedral mesh (mesh $ID = 1$).

Thus the following study attempts to look at the impact of domain size on natural ventilation simulations, and also compare the number of elements generated for both uniform all-hexahedral mesh and adaptive hexahedral-dominant mesh. Three levels of domain size are selected: small, medium and large. The small domain is the same size as the model being studied in the previous section, which consists of only Building 661 and its closest neighboring building to the north. The medium domain consists of all the immediate surrounding buildings near Building 661. The large domain is five times as large as the medium domain, which will satisfy the recommendation from AIJ and COST of less than 3% foot print coverage ratio. The statistics of the generated meshes are summarized in Table 4. As shown, there is minimum changes in results between the uniform mesh and adaptive mesh. However, there are significant reduction in ACH rates when the domain size increases, which also support the recommendations from COST and AIJ.

Table 4: A comparison of the simulation results and size of vertices and elements between uniform all-hexahedral meshes and adaptive hexahedral-dominant meshes, with different simulation domain sizes.

Domain Size (m)	Type of Mesh	Resolution(m)	# Vertices	# Elements	ACH ($1/hr$)	% Change
$(227, 182, 68)$	Uniform all-hexahedral	1.0	2,777,733	2,852,107	12.67	–
	Adaptive hexahedral-dominant	1.0/1.0/2.0	395,148	380,991	12.68	0%
$(338, 269, 102)$	Uniform all-hexahedral	1.0	12,218,100	11,991,081	10.71	−15%
	Adaptive hexahedral-dominant	1.0/1.0/4.0	558,204	560,214	10.65	−16%
$(1343, 1692, 507)$	Adaptive hexahedral-dominant	1.0/1.0/8.0	1,398,321	1,362,217	7.78	−39%

Note: the percentage change values are the changes of ACH from that of the uniform all-hexahedral mesh with the smallest domain size.

6 CONCLUSION AND FUTURE WORKS

This paper presents an automatic mesh generation tool to generate the adaptive hexahedral-dominant mesh and the uniform all-hexahdral mesh from architecture conceptual design tools for CFD simulation in the architectural domain. The meshing tool will take as input the VRML model generated by Google SketchUp, which is an architectural conceptual design tool. The octree data structure is used to generate with a background grid, and the octree is balanced with the neighboring octants level difference of no more than one. The contouring technique is then used to generate conformal uniform all-hexahedral mesh or adaptive hexahedral-dominant meshes. A special method is designed to limit the octree collapsing level and at the same time keep most of the sharp features in the geometry model.

The number of vertices and elements generated by the algorithm are compared with uniform-tetrahedral meshes generated by Delaunay Triangulation algorithm. The hexahedral meshes (uniform and adaptive meshes) significantly reduce the number of elements with same resolution and simulation domain size.

The presented meshing algorithm is applied to a live retrofit project (Building 661 in the navy yard, Philadelphia). The generated meshes show good quality in terms of aspect ratio, face squish index and volume squish index. Simulation results in terms of the air change rate of the indoor space show that with much less number of elements (as low as 4%), the adaptive mesh generates very close (less than 1% difference) results with the uniform mesh. A reduction in mesh resolution shows a change in the resulting ACH (5%). The study on simulation domain sizes also supports the recommendations by the European Cooperation in Science and Technology (COST) and the Architectural Institute of Japan (AIJ).

The algorithm is by no means the end result of this research effort. Future work includes the following:

- Develop an adaptive all-hexahedral meshing algorithm.
- Apply pillowing technique to generate the boundary layer, and apply smoothing techniques to improve the mesh quality.
- Implement a more robust method to handle sharp features in non-manifold geometry models.
- Compare CFD simulation results with experimental data, to investigate the effectiveness of polyhedral meshes compared to hexahedral meshes.

- Conduct field measurement of the completed retrofit project in Philadelphia to validate the finding on the impact of simulation domain sizes on natural ventilation simulations.

REFERENCES

Baker, T. J. 2001. "Mesh movement and metamorphosis". In *Proceedings of The 10th International Meshing Roundtable*, edited by A. Sheffer, 387–396.

Benzley, S., E. Perry, K. Merkley, B. Clark, and K. Sjaardema. 1995. "A comparison of all-hexahedral and all-tetrahedral finite element meshes for elastic and elasto-plastic analysis". In *Proceedings of The 4th International Meshing Roundtable*, edited by T. Tautges, 179–191.

Blacker, T. D., and M. B. Stephenson. 1991. "Paving: a new approach to automated quadrilateral mesh generation". *International Journal for Numerical Methods in Engineering* 32:811–847.

Borouchaki, H., and P. J. Frey. 1998. "Adaptive triangular-quadrilateral mesh generation". *International Journal for Numerical Methods in Engineering* 41:915–934.

Edgel, J. 2010. *An Adaptive grid-based all hexahedral meshing algorithm based on 2-refinement*. Master thesis, Brigham Young University, Prove, UT, USA.

Fluent 2007. *Fluent 6.3 Documentaion*.

Franke, J., C. Hirsch, A. G. Jensen, H. W. Krus, M. Schatzmann, P. S. Westbury, S. D. Miles, J. A. Wisse, and N. G. Wright. 2004. "Recommendations on the use of CFD in wind engineering. In: van Beeck, J.P.A.J. (Ed.), COST Action C14, Impact of wind and storm on city life built Environment.". In *Proceedings of The International Conference on Urban Wind Engineering and Building Aerodynamics*, 5–7.

Harris, N. J., S. E. Benzley, and S. J. Owen. 2004. "Conformal refinement of all-hexahedral element meshes based on multiple twist plane insertion". In *Proceedings of The 13th International Meshing Roundtable*, edited by A. Ungor, 157–168.

Hefny, M. M., and R. O. Ooka. 2008. "Influence of cell geometry and mesh resolution on large eddy simulation predictions of flow around a single building". *Building Simulation* 1 (3): 251–260.

Ito, Y., A. M. Shih, R. P. Koomullil, and B. K. Soni. 2006. "A solution-based adaptive redistribution method for unstructured meshes". In *Proceedings of The 15th International Meshing Roundtable*, edited by P. P. Pebay, 147–162.

Ito, Y., A. M. Shih, R. P. Koomullil, and B. K. Soni. 2007. "An extension of the reliable whisker weaving algorithm". In *Proceedings of The 16th International Meshing Roundtable*, edited by M. L. Brewer and M. David, 14–17.

Ju, T., F. Losasso, S. Schaefer, and J. Warren. 2002. "Dual countouring of hermite data". In *Proceedings of SIGGRAPH*, edited by T. Appolloni, 339–346.

Marechal, L. 2009. "Advances in octree-based all-hexahedral mesh generation: handling sharp features". In *Proceedings of The 18th International Meshing Roundtable*, edited by B. Clark, 65–84.

Schneiders, R. 1996. "A grid-based algorithm for the generation of hexahedral element meshes". *Engineering with Computers* 12:168–177.

Schneiders, R., R. Schindler, and F. Weiler. 1996. "Octree-based generation of hexahedral element meshes". In *Proceedings of The 5th International Meshing Roundtable*, edited by S. Mitchell, 205–215.

Shewchuk, J. R. 2001. "Delaunay refinement algorithms for triangular mesh generation". *Computational Geometry: Theory and Applications* 22:21–74.

Staten, M. L., S. J. Owen, and T. D. Blacker. 2005. "Unconstrained paving & plastering: a new idea for all hexahedral mesh generationn". In *Proceedings of The 14th International Meshing Roundtable*, edited by B. W. Hanks, 399–416.

Tautges, T. J., S.-s. Liu, Y. Lu, J. Kraftcheck, and R. Gadh. 1997. "Feature recognition applications in mesh generation". *Conference: Joint ASME, ASCE, SES Symposium on Engineering Mechanics in Manufacturing Processes And Materials Processing*:117–121.

Tchon, K., J. Dompierre, and R. Camarero. 2002. "Conformal refinement of all-quadrilateral and all-hexahedral meshes according to an anisotropic metric". In *Proceedings of 11th International Meshing Roundtable*, edited by N. Chrisochoides, 231–242.

Tchon, K., C. Hirsch, and R. Schneiders. 1997, June. "Octree-based hexahedral mesh generation for viscous flow simulations". In *Proceedings of the 13th AIAA Computational Fluid Dynamics Conference*.

Tominaga, Y., A. Mochida, R. Yoshie, H. Kataoka, T. Nozu, M. Yoshikawa, and T. Shirasawa. 2008. "AIJ guidelines for practical applications of CFD to pedestrian wind environment around buildings". *Journal of Wind Engineering and Industrial Aerodynamics* 96:1749–1761.

Wilcox, S., and W. Marion. 2008, May. "Users manual for TMY3 data sets users manual for TMY3 data sets". Technical report, National Renewable Energy Laboratory.

Woodbury, A. C., J. F. Shepherd, M. L. Staten, and S. E. Benzley. 2011, January. "Localized coarsening of conforming all-hexahedral meshes". *Engineering with Computers* 27:95–104.

Zhang, Y., and C. Bajaj. 2006. "Adaptive and quality quadrilateral/hexahedral meshing from volumetric data". *Computer Methods in Applied Mechanics and Engineering* 9 (195): 942–960.

Zhang, Y., C. Bajaj, and G. Xu. 2009. "Surface smoothing and quality improvement of quadrilateral/hexahedral meshes using geometric flow". *Communications in Numerical Methods in Engineering* 25 (1): 1–18.

AUTHOR BIOGRAPHIES

Rui Zhang is a PhD Student at School of Architecture, Carnegie Mellon University at the time the paper was written. She joined IBM T.J. Watson Research Center as a post doctoral researcher in Aug. 2011. She received dual B.Eng. in Environmental Engineering and Computer Science, and M.Eng. in Environmental Engineering from Tianjin University China, and Ph.D. from Carnegie Mellon University in the field of Building Performance and Diagnostics. Her research interests include the development and application of computation tools for the design and performance diagnostics of sustainable buildings. One focus of her research interest is the automated and advanced mesh generation for complex free form architectural designs. She is also working on developing analytics and automation models for smart buildings. Her email address is zhangrui@us.ibm.com.

Khee Poh Lam is a Professor of Architecture at School of Architecture, Carnegie Mellon University, PA. He teaches architectural design, building performance modeling, building controls and diagnostics, acoustics and lighting. His fields of research are in total building performance studies and the development of computational design support systems. His work has been widely published and he serves as a member of the Editorial Boards of the Journal of Building Performance Simulation (UK), and Building Simulation: An International Journal (China). He is a member of the US Energy Foundation Board of Directors. He actively works with their China Sustainable Energy Program on a range of activities including green building codes and standards and green design of various demonstration projects in China. He is also a building performance consultant for several major award winning projects in the private and public sectors in Singapore, and remains engaged in ongoing projects in the USA and Taiwan. He is currently Visiting Professor at the Department of Architecture, Chinese University of Hong Kong. His e-mail is kplam@cmu.edu.

Yongjie Zhang is an Assistant Professor in Mechanical Engineering at Carnegie Mellon University. She received her B.Eng. in Automotive Engineering and M.Eng. in Engineering Mechanics from Tsinghua University (China), and M.Eng. in Aerospace Engineering & Engineering Mechanics and Ph.D. in Computational Engineering & Sciences from the University of Texas at Austin. Her research interests include computational geometry, mesh generation, computer graphics, visualization, finite element method, isogeometric analysis and their applications. She co-authored over 70 publications in international journals and conference proceedings. She is the recipient of Office of Naval Research Young Investigator Award and the George Tallman Ladd Research Award. Her email address is jessicaz@andrew.cmu.edu.