Geoinformatica (2012) 16:307–327
DOI 10.1007/s10707-011-0138-3

Generating seamless surfaces for transport and dispersion modeling in GIS

Fernando Camelli · Jyh-Ming Lien · Dayong Shen · David W. Wong · Matthew Rice · Rainald Löhner · Chaowei Yang

Received: 7 February 2011 / Revised: 1 July 2011
Accepted: 26 July 2011 / Published online: 10 August 2011

Abstract A standard use of triangulation in GIS is to model terrain surface using TIN. In many simulation models of physical phenomena, triangulation is often used to depict the entire spatial domain, which may include buildings, landmarks and other surface objects in addition to the terrain surface. Creating a seamless surface of complex building structures together with the terrain is challenging and existing approaches are laborious, time-consuming and error-prone. We propose an efficient and robust procedure using computational geometry techniques to derive triangulated building surfaces from 2D polygon data with a height attribute. We also propose a new method to merge the resultant building surfaces with the triangulated terrain surface to produce a seamless surface for the entire study area. Using Oklahoma City data, we demonstrate the proposed method. The resultant surface is used as the input data for a simulated transport and dispersion event in Oklahoma City. The proposed method can produce the seamless surface data to be used for various types of physical models in a fraction of the time required by previous methods.

Keywords GIS · Computational geometry · Computational fluid dynamics · Transport and dispersion · CAD · Mesh generation

1 Introduction

Triangle-based surface modeling is one the most popular representations of surfaces [1]. In GIS, the Triangulated Irregular Network (TIN) data structure is commonly used to display values along the third dimension, which can be elevation or any attribute of locations [2]. TIN is a topic of great interest in GIS starting from the early era when

Funded by the College of Science, George Mason University

F. Camelli (✉) · J.-M. Lien · D. Shen · D. W. Wong · M. Rice · R. Löhner · C. Yang
College of Science, George Mason University, Fairfax, VA, USA
e-mail: fcamelli@gmu.edu

modeling terrain and landscape had been one of the primary functions of GIS [1, 3, 4], and TIN has been used frequently to build digital terrain models (DTMs). Much effort has been devoted to develop and improve algorithms to generate "optimal" triangulations (e.g., see review in de Berg 2008 [5], and Löhner 2008 [6]). However, the use of triangulation is not limited to the GIS arena. When solving numerical partial differential equations (e.g. Finite Element Methods [7], or Finite Volume Methods [8, 9]), a tessellation of the computational domain is needed, and this tessellation can be a triangulation in 2-dimensional problems. Examples of partial differential equations are the Navier-Stokes Equations, which model fluid dynamics; the Maxwell Equations, which model electromagnetic phenomena; and the Transport and Dispersion Equations, which model mass transport. Regardless of whether the spatial domain is at a local or meso-scale, meshes in the form of triangulation are often used to represent the computational domain.

In GIS, triangulation is often used as a backdrop in depicting the landscape topography on which various geographical phenomena draw for elevation or topography information [1]. For instance, in modeling drainage systems, TIN provides critical inputs in many situations [10, 11]. However, the use of TIN in GIS is often limited to depicting one spatially continuous theme, such as topography in the form of elevation, and other themes which may be regarded as part of the landscape, such as vegetative cover or buildings, are often ignored or not integrated into the surface modeling. The way that TIN is used in other physical sciences is somewhat different from the way it is used in GIS. For instance, in the atmospheric transport and dispersion model (ATD) at the urban scale, the TIN may need to include all physical features or objects that may affect the ATD model simulation. Besides landscape topography, major features such as buildings, landmarks and monuments need to be captured by the surface model which provides the spatial domain for the numerical simulation. For other more precise applications, features such as vegetations, or surface roughness may need to be part of the model if the applications are at a very fine spatial scale. The creation of such triangulation or TIN is not typical in GIS, and in fact, posts some challenges to current GIS technology [6]. This paper focuses in the generation of TIN surfaces which include landscape features other than topographic characteristics.

GIS provide an appealing environment to physical scientists to generate their triangulations or surfaces to support their applications such as dispersion and transport modeling [12–14] because of the wide availability of various types of geospatial data, including elevation data, geometry information of buildings, and other landscape features in GIS formats. Nevertheless, the current GIS technology has some limitations in generating an integrated TIN surface. Besides the fact that most TIN generation in GIS consider only one theme (e.g., terrain elevation) but not multiple themes together (e.g., terrain, trees, and buildings), another challenge is to use the same triangulation data structure to model relatively complex buildings or structures. ATD simulations at the urban scale need to combine all the different feature themes and complex physical structures into one seamless surface as an input to the algorithm that creates the tessellation in the computational domain. Integrating the landscape topography with other feature themes to generate a TIN with elevation is a time-consuming, laborious and error-prong process that requires human intervention. Also, little has been done to incorporate complex building structures into the triangulation process in an automatic way [15]. Therefore, our objective is to improve the efficiency of related algorithms to generate a surface integrating topography and buildings. Such surfaces may be used in various scientific applications to support modeling and simulation.

2 Relevant background and research challenges

2.1 Computational fluid dynamics and transport and dispersion modeling

Transport and dispersion modeling is of great interest in environmental science, and it models advection (mass transport processes), and diffusion (mass dispersion processes) of chemicals (particulate, or gaseous, or liquid state) in the three media of soil, water and air [16, 17]. While the specific structure of models for different phenomena may be different, most models depict levels of concentration in water, air, or soil over space and time. These models are also different in their spatial coverage extents (e.g., micro-, urban-, meso-, regional-, or global-scale). For instance, models such as Weather Research and Forecasting (WRF) are used to simulate atmospheric events at a meso-scale level [18], and Computational Fluid Dynamics (CFD) models are often used to model atmospheric pollution at the urban scale [19–21].

This paper will focus on the generation of input data for models that study the transport and dispersion of gaseous pollutants in the atmosphere at the urban scale. The research on transport and dispersion of gases for scales larger than a city using Gaussian models have been the successfully applied [22, 23]. Unfortunately, these relatively simple models have been unable to reproduce all pertinent complex processes at the urban-scale level because relevant sources for the mechanical (i.e., building geometry, trees, traffic) and thermal (i.e., surface heating, HAVC systems) forces that control the dispersion at the urban scale are not captured. Dispersion models that use first principle physics are available today thanks to the sustained increase of computational capability. The use of CFD models has been noted for its ability to represent and characterized the flow patterns in complex urban environments [24–26]. The accuracy of these models has been tested and proven to be accurate in the past decades for a broad range of engineering applications [6] and for predicting atmospheric dispersion patterns [26–28]. Thus this modeling approach is valuable to support emergency planning and impact assessment effects due to atmospheric releases, either from natural or man-made events [14, 29]. These models have become a common tool in the planning of field experiments in urban environments [19].

Any ATD model based on CFD needs a tessellation of the computational domain where the partial differential equations are solved. The computational domain of modeling atmospheric dispersion is usually 3-dimensional. The tessellation of a 3-dimensional domain fills the space with geometrical objects called elements. The collection of these elements composes what is called a mesh or grid in CFD modeling. Meshes can be structured (e.g. bricks) or unstructured (e.g. tetrahedra). Any tessellation algorithm needs the boundary surface information of the computational domain. This geometry description of the boundary surface of the domain is represented as a collection of surfaces, lines and points in a CAD application. An attractive way of representing the boundary surfaces is a collection of triangles, which may be regarded as discrete surfaces [6]. Other possible representations of surfaces are the analytic, the bilinear transfinite Coon's patch, the triangular Barnhill-Gregory-Nielsen patch, or the iso-parametric patch [6].

CFD models need a correct representation of buildings and terrain in order to produce highly accurate predictions of atmospheric dispersion. Accurate representations of buildings and terrain (e.g., building shapes, building heights, terrain slopes) are prerequisites to produce an accurate solution of the flow patterns in the prevailing down wind direction. Popular data formats representing buildings include shapefiles, stereo lithography (*.stl), and proprietary formats like DWG, DWF, DXF, and OpenFlight (*.flt). Unfortunately, these data are usually not publicly available and have to be acquired from private companies or organizations. These data have to be converted into formats suitable for ATD models. The conversion process is often tedious and slow. Some approaches have been proposed to ingest commercial data formats into CFD/ATD models [13, 20, 30], but the processes still cannot be fully automated. The information about terrain and buildings is usually stored in GIS databases, e.g. TIN format for terrain and shapefile for buildings. The TIN and shapefile data are usually used as inputs for the CAD reconstruction of the computational domain. The building data are not necessary a seamless representation, making the input stage inefficient for obtaining a seamless/water tight surface for the CAD representation that is used for the tessellation step.

2.2 Triangulation and tessellation

The literature shows that many methods can successfully create tessellations to support various types of modeling effort and a review of common tessellation methods can be found in Okabe et al. (2000) [2] and Löhner (2008) [6]. Besides regularly gridded or raster partitions, unstructured tessellation (e.g., triangles in 2 dimensions, and tetrahedra in 3 dimensions) is probably one of the most popular alternatives for partitioning space represented by a complex geometry. Unstructured meshes are especially appealing because they can easily be used to model 3-dimensional space and are highly flexible in their implementations [1, 31–34]. Data representing surfaces in 3 dimensions can be obtained from various types of spatial sampling schemes (e.g., random, stratified, or regular-grid sampling), profiling, or contouring, and they can be triangulated to represent 3-dimensional surfaces. Therefore, triangulation, especially TIN, is regarded as a desirable approach to reconstruct topographical surfaces in GIS [1]. Outside of GIS, triangulation has been used in various applications such as computer graphics, robotics, image synthesis, stereo lithography, garment design, shape/surface reconstruction and finite element analysis [35–38].

According to Owen (1998) [39] and Löhner (2008) [6], triangulation methods can be categorized into Delaunay triangulation, advancing front triangulation, and quadtree/octree triangulation. Among the three methods, Delaunay triangulation has been regarded as the preferable one due to many desirable properties such as the sizes and angles of resultant triangles [2, 40, 41]. Other general properties of Delaunay triangulation have also been discussed in the literature (e.g., Gold 1999 [41]).However, using Delaunay triangulation to generate 3-dimensional surfaces to support the meshing of the computational domain for ATD simulations is subject to several methodological challenges. The surfaces required for ATD simulations need to describe not just the land surface topography, but also man-made structures, especially buildings, all seamlessly integrated. The surface should be coherent such that it is water-tight. However, in GIS, surface topography is often represented by DEM, or TIN, while buildings are represented by footprints of 2D polygons (shapefiles or other formats) or 3-dimensional polygons (CAD format, VRML, Collada, etc.). Existing research often resorts to laborious manual manipulations of the geometry data representing the topography and buildings to produce a coherent and consistent geometrical representation of the surface including landscape and buildings.

Past approaches have used a background mesh as a framework to merge the terrain and surfaces to obtain a seamless surface. A distance function was calculated from the background mesh points to the surfaces and terrain, where positive and negative distances may indicate outside and inside of the 3D objects. Once the distance function is obtained, the iso-surface that represents the zero level distance is extracted. This algorithm is called the level set [42, 43]. This approach show to be feasible for a small number of buildings, but is unreliable and not feasible for an entire urban area. The level set algorithm also fails to make a proper representation of the intersection between terrain and buildings or other structures. Another approach is to utilize some modified Delaunay methods. Such methods carry different names, such as the generalized Delaunay triangulation [44], constrained Delaunay or obstacle triangulation [45, 46], or restricted Delaunay triangulation [35, 47]. This family of constrained triangulation methods can deal with very complex objects and truncated surfaces, such as topography with building footprints removed. However, a prerequisite for applying this family of relatively sophisticated triangulation methods is a consistent geometrical representation of all involved surfaces (topography) and structures (buildings, landmarks, etc). As discussed before, topography and building structures are often represented by different data layers and stored in different formats in GIS, and thus the family of constrained triangulation methods is not particularly useful.

2.3 Modeling complex objects by merging 2.5D primitives into seamless 3D models

Another challenge in generating triangulated surfaces for ATD modeling is to capture the man-made landscape accurately, particularly building structures. Detailed drawings of buildings are quite often available from architectural drawings in CAD formats, but gathering them for an extensive geographical region is quite laborious. On the other hand, geospatial data at the urban-city or regional scale often include building boundaries or footprints. Depending on the data collection or acquisition process, building data can be as simple as polygons representing the outlines of buildings without any elevation or height information, or can be as complex as overlapping polygons representing the perimeters of the buildings at different elevations. Recreating the structures of buildings using this type of data is quite laborious and almost impossible with existing GIS technology. Such data, without further manipulation and processing, cannot be fed into any typical triangulation algorithms to generate surfaces including the buildings.

The challenge can be formalized in the following manner. Let $\{P_i\}$be a set of polyhedra. Although we are not limited ourselves to any type of polyhedra, the polyhedra in our examples are generated by extruding the polygons defined in a given shapefile. These polyhedra collectively represent the architectural structures when visualized from a view external to all the polyhedra.The goal is extract a well-defined and seamless boundaries of the structures. More specifically, our goal is to compute $\partial(\cup_i P_i)$, i.e., the boundary of the union of all P_i. In CAD modeling and computer graphics, this process is known as geometric boolean operations. The problem of geometric boolean operations has been studied for more than three decades and the main focus of the research is on the *robustness* of the computation because many numerical errors and degenerate cases can creep in during the computation and result in incorrect output. In addition to the robustness issues, another main challenge is the scalability of the algorithm for computing the union of a very large number of polyhedra. In our example, $\{P_i\}$ can have thousands of elements. Naïvely computing the union between pairs of polyhedra in $\{P_i\}$ can be impractical.

To show the significance of the problem, we have attempt to compute the union of all the buildings in our Oklahoma city dataset using the union function provided by ArcGIS. ArcGIS takes hours to complete the computation and requires specific types of overlaps between the components in order to generate successful unions. Therefore, our proposed approach is designed to tackle these serious deficiencies to improve both the robustness and efficiency of the process. As detailed in the experimental results section later, our new approach takes only seconds on the same dataset, and successfully handles degenerated cases. To simplify our discussion, we will first describe our method using 2D polygons and then extend it to 3D in the following section.

3 Proposed procedure to generate seamless surfaces

To generate a seamless surface including both topography and building structures, we adopt the following sequence of steps. Using geospatial data of building outlines as inputs, we demonstrate the creation of reasonably realistic building objects that can be integrated with the topographic surface. We use a constrained triangulation method to generate a water-tight surface of the topography, but excluding building objects. Then the surface describing the terrain is stitched together with the 3D building objects, creating a seamless surface that is used as input data for a mesh generator. The ultimate goal is to expedite the preparing of data to be used for ATD and other physical modeling processes.

3.1 Creating seamless building surfaces from geospatial data

3.1.1 Extracting the external boundary of the union of polygons (2D)

The main idea of our approach is to *incrementally* extract the boundary of the arrangement induced from the input polyhedra. During the extraction process, we identify and extend the extracted boundary by maintaining its desired topological properties. Since our main application is a Finite Element Method (FEM) based simulation, we are only interested in creating models from the *external boundary* of the union, i.e., the boundaries of holes will be ignored. Extracting the external boundary of the union can be done much more easily and efficiently than that of the hole boundaries.

Let $\{P_i\}$ be a set of polygons. Our goal is to compute the outmost boundary of $\partial(\cup_i P_i)$. To simplify our notation, we let $Q = \partial(\cup_i P_i)$. For each polygon P, we denote the vertices of P as $\{p_i\}$ and the edge that starts at vertex p_i as $e_i = \overline{p_i p_{i+1}}$. The edge e_i has two associated vectors, the vector from p_i to p_{i+1}, i.e., $\mathrm{v} = \overrightarrow{p_i p_{i+1}}$, and the outward normal n_i. Traditionally, $\{P_i\}$ contains only two elements, and the boundary of Q is determined by computing the arrangement of the edges of $\{P_i\}$, which is a subdivision of the space into vertices, edges and faces (cells) from a set of line segments. One way to extract the boundaries from such an arrangement is by finding all the faces that have positive winding numbers [48, 49].

Computing the arrangement can be time consuming, i.e., $O(n^2)$ for n line segments. Our method skips arrangement computation and find the boundary by computing the intersections on the fly. To start the extraction process, we compute the rightmost vertex r of all vertices in $\{P_i\}$. We further let e_r be an edge incident to r such that e_r's outward normal has the largest x coordinate among all the edges incident to r. It is simple to show that r must be a vertex of Q, and e_r must contribute to one or multiple edges of Q. See Fig. 1(a).

Our method then proceeds by incrementally discovering the vertices and edges of Q from r and e_r. To slightly abuse the notation, we let r be the latest vertex of Q discovered, and let e_r be an edge of $\{P_i\}$ that contains an edge of Q. Therefore, in every incremental step, our method will need to (1) identify which portion of e_r belongs to Q, and (2) identify the next r and e_r until all edges of Q are discovered.

To identify the portion of e_r that contributes to Q, let $\{x_i\}$ be a sorted list of intersections between e_r and other line segments $e_j \neq e_r$ in $\{P_i\}$. The intersections $\{x_j\}$ are sorted in non-decreasing order using the distance to r. Therefore x_0 is the intersection closest to r. Now we claim that x_0 must be a vertex of Q and the segment $\overline{rx_0}$ between r and x_0 must be an edge of Q. This observation is proven in Lemma 1.

Lemma 1 Let x_0 be the closest intersection to r. We say that x_0 must be a vertex of Q and the segment $\overline{rx_0}$ between r and x_0 must be an edge of Q.

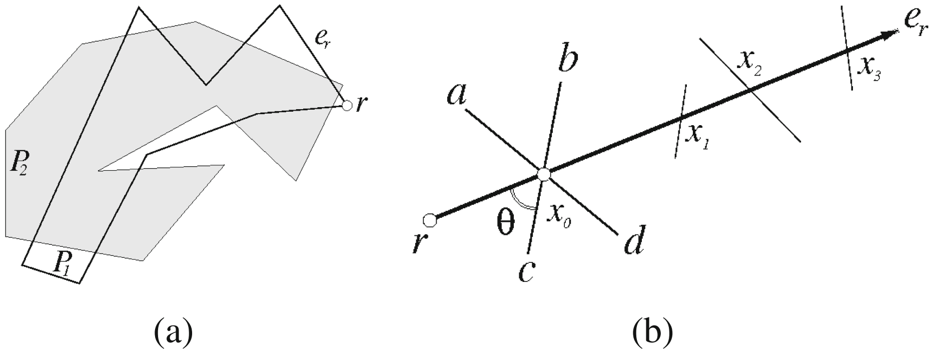


Fig. 1 **a** The union of two polygons P_1 and P_2. The vertex r is the rightmost vertex of P_1 and P_2, and e_r is the edge incident to r whose outward normal has the largest x coordinate among all the edges incidents to r. The pair r and (a subset of) e_r must be on $\partial(\cup_i P_i)$. **b** Given the last vertex r and a potential edge e_r discovered in the extraction process, the segment $r\overline{x}_0$ must be an edge of Q and the next r is x_0 and the next e_r is the edge containing $x_{\theta}c$

Proof Assuming that x_0 is not ∂Q. Then x_0 must be interior to Q. Since we know that r is a vertex of Q, when we move from r to x_0, there must be a point $x' \in \partial Q$ before we reach the interior of Q. If we wish to remain on the boundary of Q, we must move to the other edge of Q at x'. Therefore, x' must be an intersection of e_r and another segment from $\{P_i\}$. However, we know that x_0 is the intersection closest to r. This means that x_0 cannot be interior to Q, and in fact x_0 and x' must be the same point and the segment $\overline{rx_0}$ must be on ∂Q.

Therefore, $\overline{rx_0}$ is an edge of Q and x_0 becomes the next r, i.e., the last vertex discovered. In the second step, we need to find out which edge of $\{P_i\}$ (that is not e_r) incident to x_0 will contain an edge of Q. Let $S = \{s_j\} \backslash e_r$ be a set of line segments incident to x_0 excluding e_r. Then to compute the next e_r, we solve the following:

$$\arg\min_{s_j \in S} \Theta(e_r, s_j),$$

where Θ is a function measuring the clockwise angle between e_r and s_j. Intuitively, the next e_r will be a line segment that makes the largest right turn from the current e_r at x_0. Now, with r and e_r updated, we repeat the process until a closed loop is found. See Fig. 1(b).

The proposed method have many advantages over existing approaches. First, in contrast to the traditional boolean operation approach, the proposed method can handle an arbitrary number of elements in $\{P_i\}$ all at once. Second, we do not have to compute the arrangement of the input segments, i.e., we avoid computing all the intersections for all the line segments in $\{P_i\}$. Instead, we compute only the intersections of all e_r discovered during the construction of Q. This is extremely helpful when the size of $\{P_i\}$ is large and the boundary of Q has only a few features (edges and vertices). This observation is usually true when the size of $\{P_i\}$ is large and for the architectural models in which many parts contribute only a small portion to the external boundary. Because of this feature, our method is more sensitive to the output complexity than existing methods. Third, the proposed method can handle degenerated cases easily, i.e., two polygons touching at a single vertex or a line. The proposed approach can even handle non-simple polygon, whose edges may self intersect, and polychain, which do not form a loop or enclose an area.

3.1.2 Extracting the external boundary of the union of polyhedra (3D)

The boundary extracting process in 3D is similar to that for polygons. The main difference is that, instead of tracking a single pair of a vertex r and a potential edge e_r, we keep track of a “wavefront” of discovered facets and edges of Q. This wavefront will advance until the entire external boundary is identified.

Let $\{P_i\}$ be a set of polyhedra. Similar to the 2D case, we maintain a list of propagation sites. Each propagation site is an edge e of Q and an incident facet f_e of e that contains one or multiple facets of Q. At each step of the wavefront propagation, we explore around a current propagation site and add the new sites to the list.

This involves the computations of all the intersections of f_e with the other facets in $\{P_i\}$. These intersections are line segments and form a subdivision (i.e., a 2D arrangement) on f_e. We then compute the face f_0 in the subdivision that is incident to e, see Fig. 2. We claim that f_0 must be a facet of Q, and the edges of f_0 must be the edges of Q. See the proof in Lemma 2.

Lemma 2 Let A be the arrangement on fe induced by the intersections of fe and other facets in $\{P_i\}$. Let $f_0 \in \underline{A}$ be the face incident to e. We say that f_0 must be a facet of Q and the boundary of f_0 must be the edges of Q.

Proof This proof is very similar to Lemma 1. Briefly, since we know that e must be on the boundary of Q, then, if we traverse on the f_e starting from any point of e, at certain point x, we will enter the interior of Q. If we wish to remain on the boundary of Q, we must move to the other facet of Q. This means that the point x must be on the intersection of f_e and other facets in $\{P_i\}$. Therefore, a set of points that are connected to e and are on the boundary of Q must be bounded by e and a set of intersection segments between f_e and $\{P_i\}$. This region is exact f_0.

At this point, we already have the portion of f_e, i.e., f_0 that forms a facet of Q and a set of future propagation sites whose associated facets are missing, i.e., the boundary of f_0. Let us consider one of the edges $e_0 \notin e$ of f_0. The other edges can be handled exactly the same. In order to continue the propagation, we will need to identify the associated facet for e_0 that will potentially contain one or multiple facets of Q.

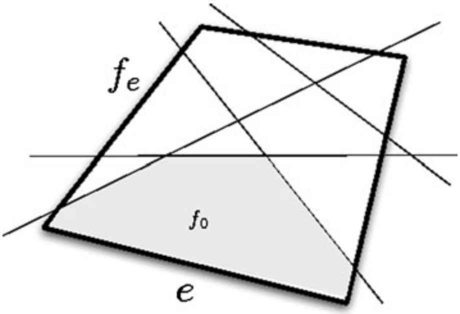


Fig. 2 The figure shows a propagation site which is composed of an edge e and a potential facet f_e discovered in the extraction process. The arrangement A on the facet f_e is induced by the intersections between f_e and the rest of the facets. $f_0 \in A$ is incident to e must be a face of Q, and the boundary of f_0 must be the edges of Q

Let $F = \{f_j\} \backslash f_e$ be a set of facets of $\{P_i\}$ incident to e_0 excluding f_e. Then in order to compute the associated facet for e_0, we solve the following:

$$\arg\min_{f_j \in F} \Theta\left(f_e, f_j\right),$$

where Θ is a function measuring the clockwise angle between f_e and f_j at e_0. That is, the next f_e will be a facet that makes the largest right turn from the current facet $f_e$at e_x.

3.2 Stitching external boundaries of buildings to a terrain surface

One of the difficulties in creating a seamless surface to depict both the terrain and buildings stems from the fact that these data usually are in different formats and may have to be processed separately. Merging them together into a triangulation surface can be laborious. We propose a novel approach to expedite this process. This approach is based upon two assumptions. The terrain surface can be triangulated with any type of triangulation method with the elevation points within the building boundaries removed. In other words, our approach is independent of the triangulation method adopted to generate the terrain surface. Another assumption is that each building object is also represented by a triangulated surface. Again, the actual method of triangulating building surfaces is not a concern. Thus, the perimeter of a building base consists of line segments with points corresponding to the vertices of triangles on the faces of the building.

After a triangulated surface is generated for each building according to the processes described in the above sections, the perimeter, or the bottom outline of the building in 3D space needs to be identified. For simple structures, the bottom outline can be the perimeter of the building and can be represented by a simple 3D polygon. But for some buildings with openings inside, then the bottom outlines may include holes within the outer 3D polygons representing the perimeters of buildings. Another example of a complicated building structure is a building that consists of two or more polyhedra on the ground, but they are only connected above ground. In these cases, the bottom outlines are represented by complex polygons.

To extract the bottom outline of a building, the wire frame of the polyhedron or polyhedra representing the building is first computed using the algorithm described in the previous section. Then points with the lowest elevation are identified. These points represent the bottom outline of the building. To stitch the building surface to the terrain surface, points defining the bottom outlines are projected onto the triangulated terrain surface. Many existing methods, such as the bilinear triangulation or finite element shape function, can be used to determine the height of the projected points on the terrain surface. Simultaneously, triangles where the points are projected are identified, and subsequently removed. Using the projected points and vertices of the reminding triangles, triangulation is performed again to stitch building surfaces and the terrain surface together into a seamless surface.

4 Demonstration examples

The Joint Urban 2003 transport and dispersion field experiment of a passive tracer was held in Oklahoma City during 2003 [50]. This experiment was part of a series of major field experiments performed in US cities. These experiments attempted to gain a better understanding of the physical phenomena of transport and dispersion in urban settings and generate enough quality data to be used in numerical modeling. The data of the Joint Urban 2003 experiment have been used to validate several CFD codes [51, 52]. We used our proposed methodology to generate a seamless surface and obtain a computational domain suitable as input for CFD models. We simulate a hypothetical transport and dispersion event using FEFLO-Urban, a general purpose finite element model [19, 26, 53, 54]. The following sections describe the data preprocess and simulation steps.

4.1 Data preprocess for Oklahoma City

The building data from Oklahoma City are in ESRI shapefile format and the terrain data are in ESRI ARC Grid format. The size of the domain that encompasses all buildings in the study region is 1.8 by 1.7 km. About 350 buildings are in the area of interest, and the buildings are represented as a set of polygons as they are depicted in Fig. 3(a). All the downtown buildings and terrain are shown together in Fig. 3(b). Although Oklahoma City is mainly in an area of flat terrain, the height difference in the area of interest is 29 m between the northwest and the southeast corners of Downtown. The horizontal resolution of the terrain data is 10 m. The building and terrain data were read using the Geospatial Data Abstraction Library (GDAL). We extruded the polygons for buildings defined in the Oklahoma City shapefile as the first step in the union process. The extruded polygons produce a set of polyhedra that represent the building structures of Oklahoma City. The extrusion height is determined by the attribute stored in the shapefile data.

Each building can be composed from one to several polygons. As an example of creating a seamless surface for a single building, we present the Ford Center Arena that is shown in Fig. 4. Figure 4(a) shows the polygons that represent the Ford Center Arena before being merged. There are 38 polygons in total that describe this particular building. Figure 4(b) shows partial sets of polyhedra obtained after the extrusion of the polygons.

All 38 polyhedra that represent the Ford Center Arena are shown in Fig. 5(a). The seamless surface that represents the Ford Center Arena is shown in Fig. 5(b). The process to

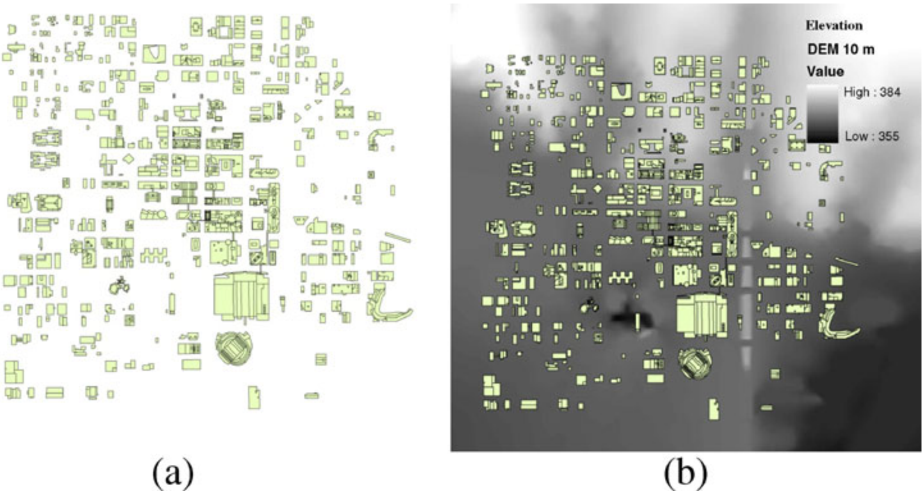


Fig. 3 **a** Down town Oklahoma City. About three hundred buildings are represented in this image. The buildings are in Shapefile format. **b** The terrain elevation and buildings in Oklahoma City. The elevation is in meters. There are about 30 m of height difference between the North West corner (highest area) and the South East corner (lowest area). The terrain representation is in DEM with a resolution of 10 m

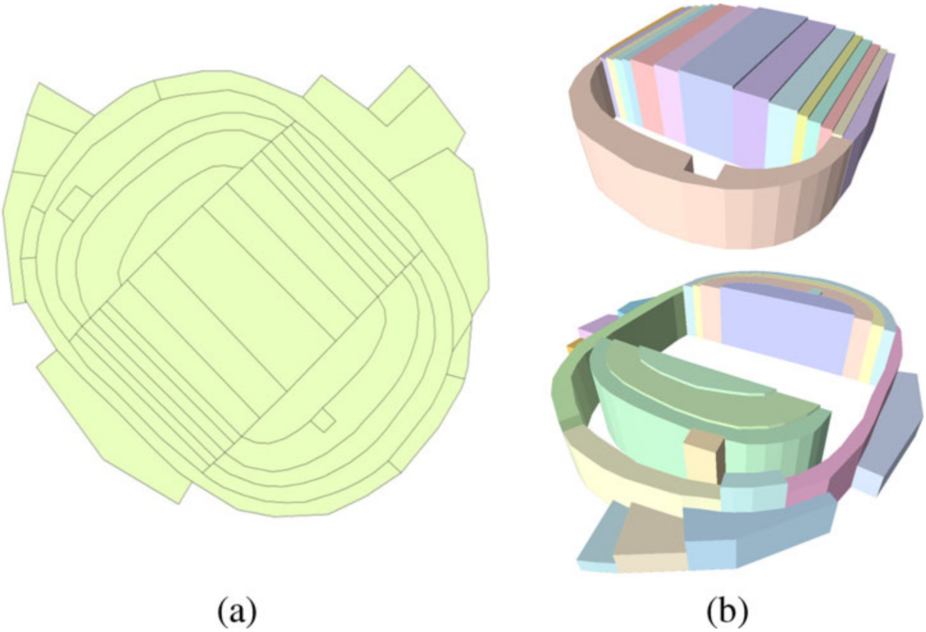


Fig. 4 Ford Center Arena. **a** 2D polygons that represent the Ford Center Arena. The Ford Center Arena is composed of 38 polygons. **b** Two sets of extruded polygons that represent the Ford Center

obtain the seamless surface for all buildings (about 350) in the study area took approximately 7 s on a 2.54 GHz Intel CPU.

In order to merge the seamless surface of the buildings and the surface terrain, we need to identify the bottom outline of each building. Figure 6(a) shows a group of buildings in Oklahoma City. The bottom surface of the group of buildings is shown in Fig. 6(b). The

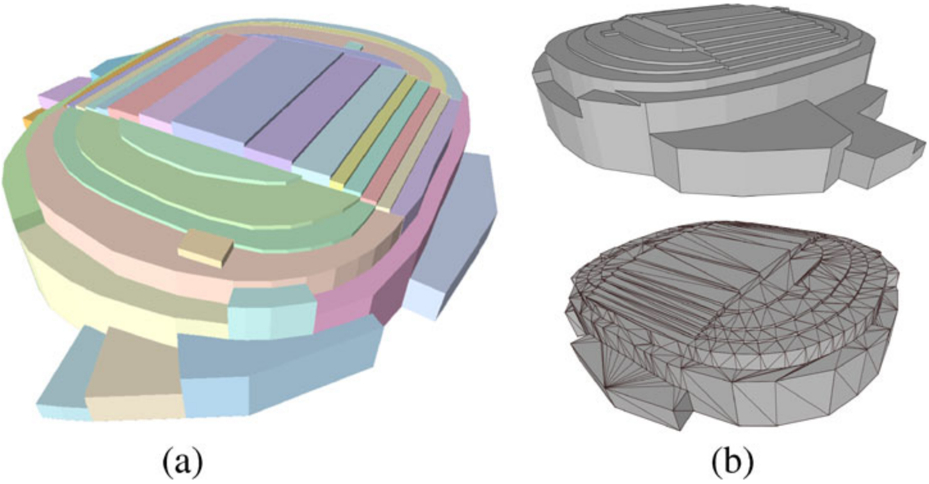


Fig. 5 Ford Center Arena. **a** All the 38 polyhedra that represents the architectural structure of the Arena. **b** Two different views of the 38 merged polyhedra that represent the Ford Center in 3 dimensions

outline of the whole group is shown in Fig. 6(c). The outline of a single building represented by a complex polygon is shown in a zoom-in area in Fig. 6(d).

The wire frames of the polyhedra were first derived. Points that are part of the lines that represent the bottom side polygons were identified. This process was repeated for all the buildings. After obtaining all the bottom 3D polygons, points representing these polygons were projected onto the triangulated terrain surface. All triangles on the terrain surface that host one or more projected points were deleted and a new connectivity preserving the projected points was constructed using a constrained triangulation.

The process of merging the building surfaces and the terrain was also performed using ArcGIS. The quality of triangulation results from both merging procedures were analyzed using the quality factor Qt [55]. The Qt factor is defined as twice the ratio of the radius of the largest inscribed circle in the triangle, and the radius of the smallest circumscribed circle in the triangle. The value of the Qt factor is between zero and one, where the value of one corresponds to an equilateral triangle. Therefore, a value of Qt closer to one indicates a better quality triangulation. The ArcGIS procedure produced an average Qt factor of 0.63, while the procedure we proposed gave an average Qt factor of 0.81, superior to the result provided by ArcGIS. The distributions of Qt values for all buildings are shown in Figs. 7 and 8.

Once the buildings and terrain were all integrated into a seamless surface, this surface was used as the input for an advancing front mesh generator [31]. The Ford Center Arena

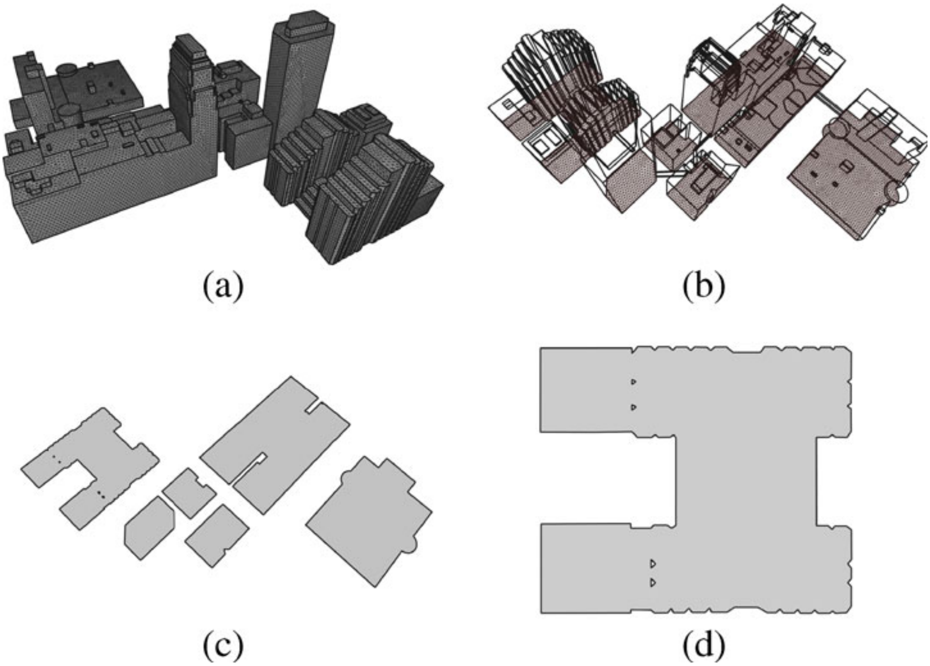


Fig. 6 Extracting the outline of the bottom side of the buildings. **a** Group of buildings in down town Oklahoma: Oklahoma Tower, Robinson Renaissance Building, Corporate Tower, Park-Harvey Center, Leadership Square, Court Plaza, First National Tower, Century Center Parking, Westin Hotel, Bank First, Bank One, and Main Street Parking. **b** Bottom side of the seamless surface of the buildings. **c** Only the bottom surface of the buildings. **d** Detail of the bottom outline of the Park-Harvey Center, Leadership Square, and Court Plaza. The outline is composed of an outer polygon and four small polygons

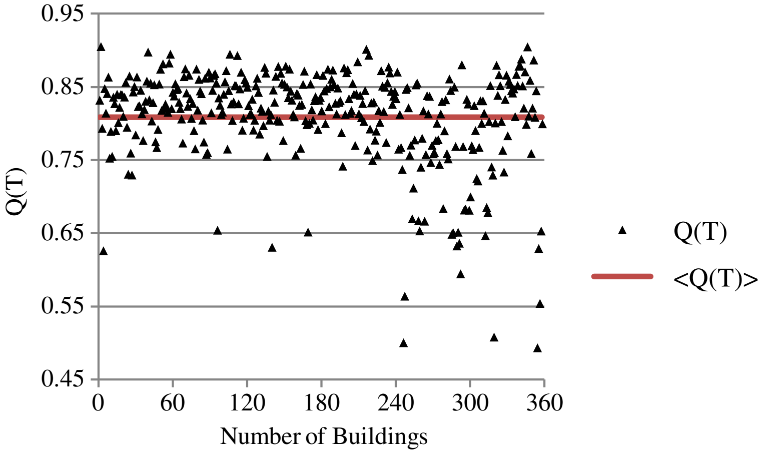


Fig. 7 Q(T) values for triangles generated with the proposed improved triangulation algorithm. The average value <Q(T)> is 0.81

finite element surface mesh produced is shown in Fig. 9(a). The triangles that form the surface mesh of the Ford Center Arena are all of uniform size. The Ford Center Arena merged with the terrain surface mesh is shown in Fig. 9(b). The triangle size is not uniform in the case of Ford Center Arena merged with the terrain. The smaller triangles are close to the terrain surface (see Fig. 9(b). The final mesh of the entire urban area is depicted in Fig. 10. The final finite element volume mesh contains 39 million tetrahedra and 7 million points.

A simulation of the flow, and the transport and dispersion of a gas was performed using the volume mesh produced with the proposed data processing methodology. The simulation assumed a boundary condition for the inflow of a logarithmic profile of 2 m/s with a velocity at 10 m from the ground. The boundary conditions in the outflow boundaries assumed a prescribed pressure. The flow was assumed to follow the incompressible Navier-Stokes equations. The first stage of the simulation was to establish a quasi-steady state regime of the flow. After this regime was reached, the transport and dispersion simulation

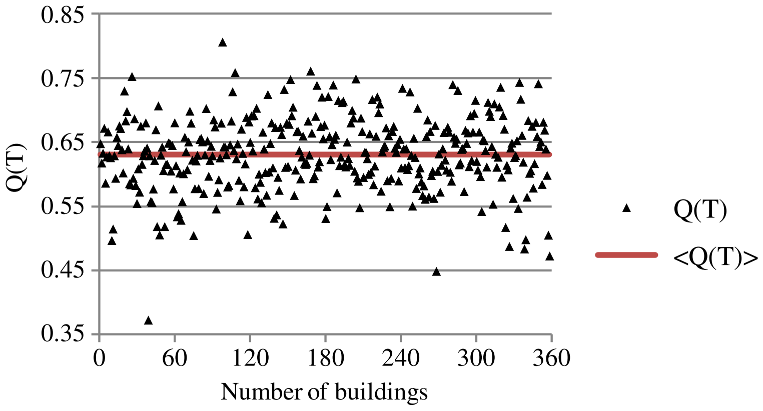


Fig. 8 Q(T) values for triangles generated using ArcGIS 9.3 TIN Management and Conversion toolset under 3D Analyst Toolbox. The average value <Q(T)> is 0.63

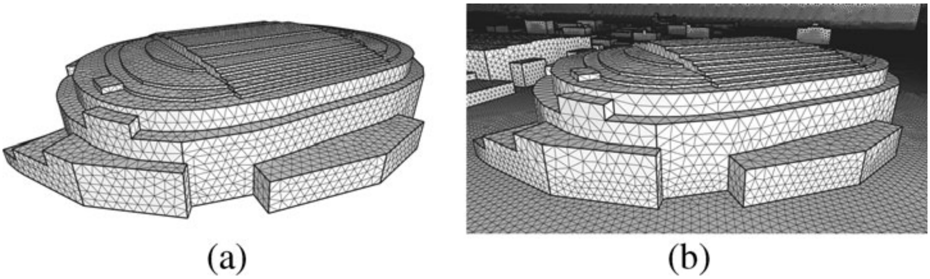


Fig. 9 Finite Element surface mesh of the Ford Center Arena. **a** Surface mesh of the merged polyhedra. **b** Surface mesh of the merged polyhedra and terrain

was initiated. The temperature effects in the flow were neglected. The characteristics of the simulated flow are complex with a large degree of variability over time. Figure 11 shows instantaneous streamlines of the flow. In Fig. 11(a) a group of streamlines in the vertical direction are shown. The streamlines at greater heights are in the direction of the main inflow, while the streamlines closer to the ground depart from the main direction. Figure 11 (b) shows how the streamlines are split into two groups, one that follows the inflow direction of the wind (aligned with the street direction), and a second group that turns from the main direction. These figures show that the flow behaves in more regularized ways at greater heights, and is more chaotic close to the ground level as was expected.

The transport and dispersion simulation was performed for a period of 500 s real time. The release type was continuous, and the simulated gas was a passive tracer with density close to the air density. The total time for the simulation, including the initialization time and dispersion, took approximately one week on a workstation running a four core Xeon X5560. An iso-surface of the concentration of level 10^{-4} ppm was extracted from the volume data for 10, 100, 250, and 500 s after the release was initiated and the corresponding clouds are shown in Fig. 12. The cloud at 10 s, Fig. 12(a), is uniformly distributed and close to the ground and it is slightly elongated along the wind direction. The

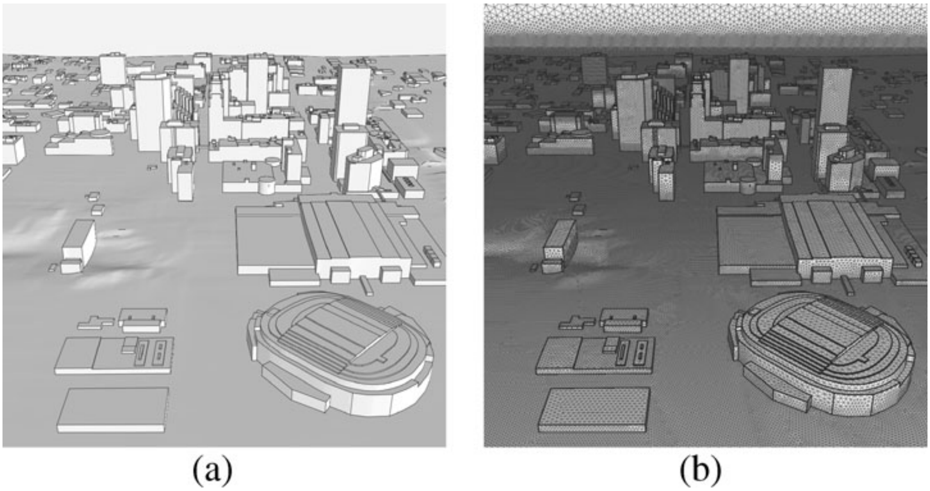


Fig. 10 Overview of Oklahoma City. **a** Shading surface of all buildings and terrain. **b** Surface mesh of all buildings and terrain

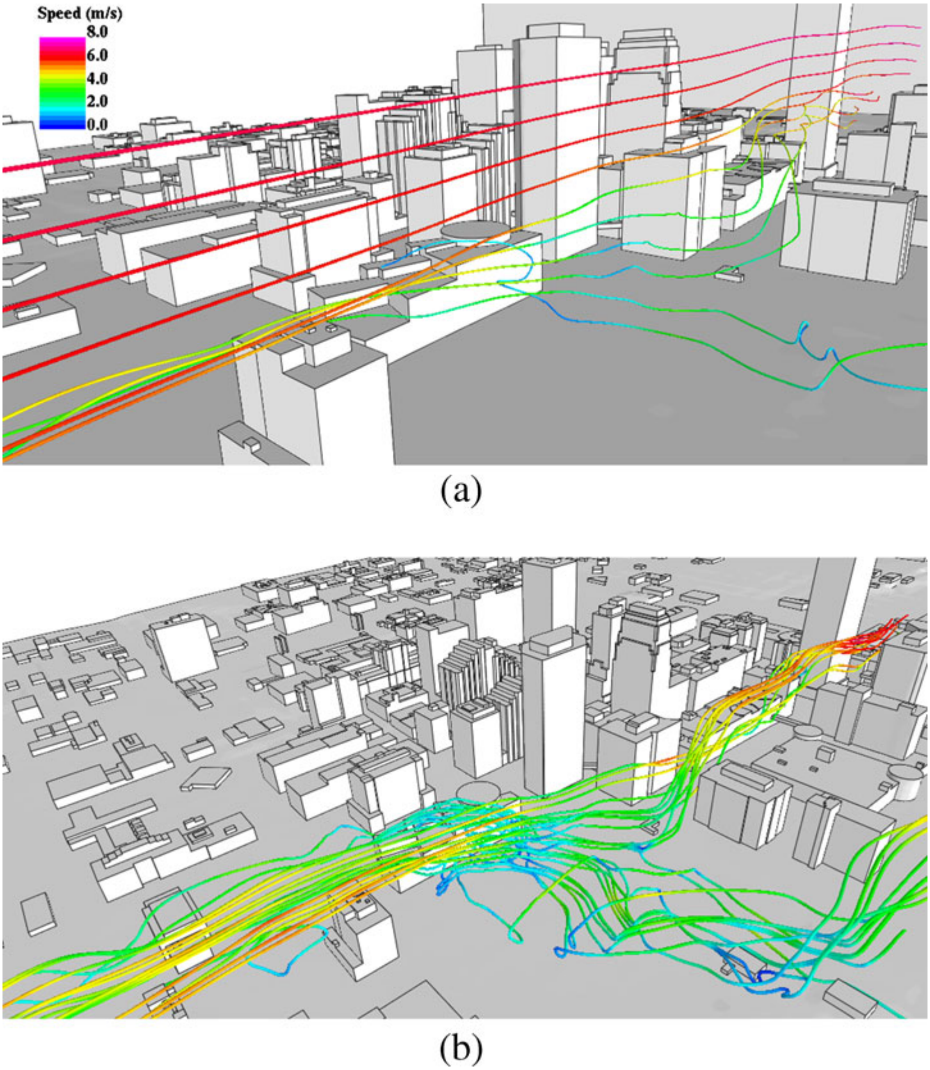


Fig. 11 Stream lines of the absolute value of the velocity. The stream lines are colored with the speed value. **a** Ten stream lines are drawn at 10 meter intervals in the vertical direction. The stream lines close to the ground show a lower velocity value (blue color) and they deviate from the main wind direction due to the objects that are in their way down wind. The stream lines at higher heights show a higher velocity (red to magenta colors) and they are all aligned with the main wind direction. **b** The stream lines depicted are five in the vertical direction by five in the transversal direction to the main wind direction. These set of stream lines are also colored with the speed velocity. They show the main deviation of the flow in the levels close to the ground due to buildings in their way down wind, while the top lines follow the main wind direction

subsequent clouds at 100, 250, and 500 s are not uniformly shaped, and they are not necessarily stretched along the wind direction. For example, at 500 s, small clouds detached from the main cloud, and these detached clouds are moving away from the wind direction, showing the effect of the wind close to the ground that was transversal to the main wind direction (Fig. 12(d)).

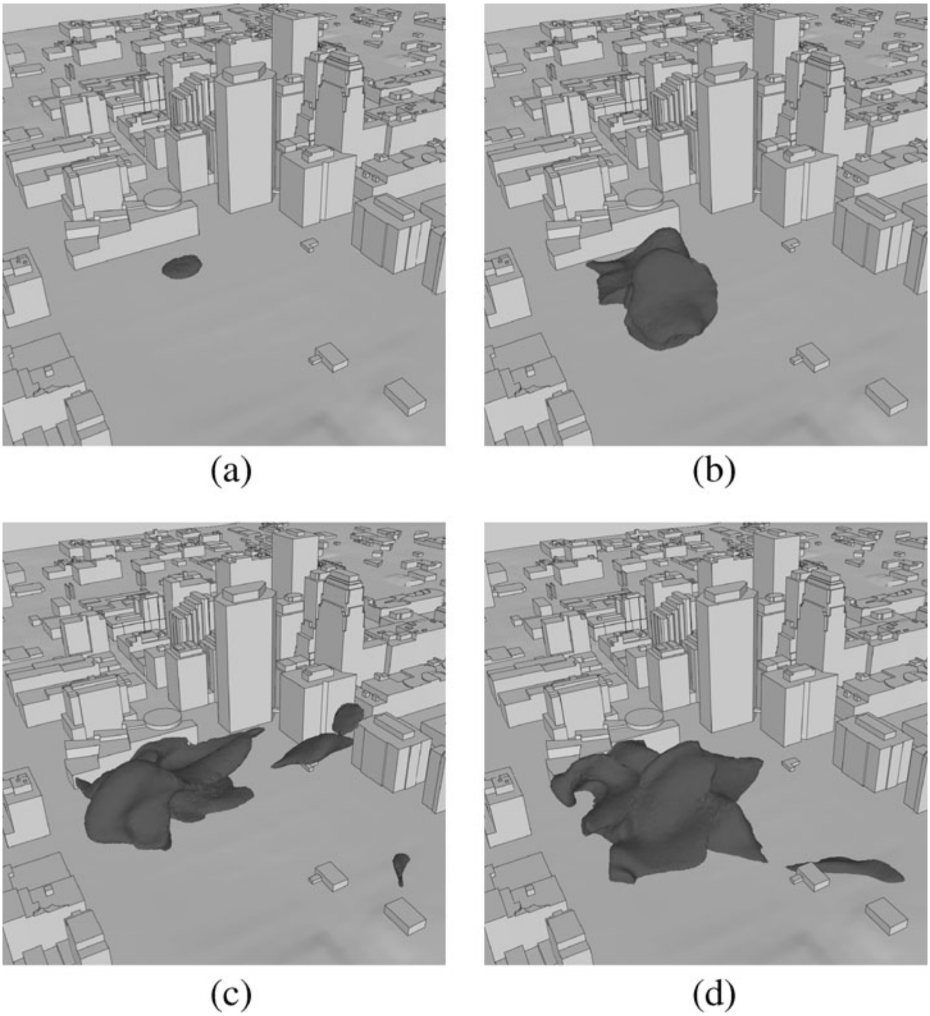


Fig. 12 A cloud is depicted at 4 different time instances. The cloud is transported and diffused by the effects of the wind and turbulence. **a** Cloud at 10 s from the beginning of the release. **b** Cloud at 100 s from the beginning of the release. **c** Cloud at 250 s from the beginning of the release. **d** Cloud at 500 s from the beginning of the release. The size and shape of the cloud change with time. The cloud represents an iso-surface of concentration level of 10^{-4} ppm

5 Conclusions

We have presented in this paper a robust and efficient procedure that produces seamless 3D surfaces of buildings from 2.5D data GIS data. The methodology was tested with the Oklahoma City 2.5D shapefile data. We have also used a simple algorithm that stitches the 3D polyhedra representing buildings created from the 2.5D data with the terrain surface. This final surface depicting both the buildings and terrain is a seamless surface. The final seamless surface was successfully used as an input to create a 3D volume tetrahedral mesh. The time of the entire process, including the creation of building polyhedra and stitching the building surface to the terrain, was less than one minute. In the past, the preparation of the input data for ATD modeling required a much longer time if the use of a water tight surface was required [21]. Previous attempts to simulate large urban areas resort to different approaches other than a seamless surface as input, such as porosity [52], or embedded models [20]. The pre-processing of input data to obtain the volume mesh usually took from several weeks to months. The input data, i.e. shapefile data and DEM data, had to be processed by hand before being used by a grid generator. The proposed approach combines creating seamless surfaces of buildings and merging buildings and terrain using a robust methodology that requires almost no user intervention. This processing time is very much shorter than the manual pre-processing required in creating a seamless surface from 2.5 data. The volume mesh was used to simulate the flow in Downtown Oklahoma city and to study the dispersion of a continuous release. We have also compared the performance of using the existing procedure in ArcGIS and our proposed procedure of stitching the building polyhedra with the terrain. The resulting triangles from the existing GIS procedure are of inferior quality when they are compared with the triangles created with our stitching process. We have also explored the use of GIS procedures to create seamless surfaces using 2.5D building data. Unfortunately, existing GIS procedures do not ensure creating a seamless surface and the time for the procedure to complete the same task was orders of magnitude longer than our proposed method.

A recent trend in GIS is the increasing use of true 3D data (e.g., Batty and Lin [56]). While many channels are available to gather and store building data with vertical information, an obvious challenge is to process these data to create realistic and accurate 3D models of building objects in large quantity efficiently for extensive areas. The computational geometry-based algorithm we proposed here is relatively robust and efficient, and should serve as one of the foundation tools to create 3D building models. To illustrate how our proposed approach can generate the seamless surface combining terrain and buildings, we use atmospheric transport and dispersion as an application example. However, many models in physical sciences also require such seamless water-tight surfaces. Our method to create such surfaces is relatively efficient, and will benefit scientists outside of the GIS arena. In this paper, the resultant seamless surface includes only terrain and buildings. The proposed method can surely accommodate the inclusion of other surface objects or features, large or small, if the modeling efforts require more detailed and comprehensive descriptions of the landscape.

References

1. Li Z, Zhu Q, Gold C (2005) Digital terrain modeling: principle and methodology. CRC Press, Florida
2. Okabe A, Boots B, Sugihara K, Chiu SN (2000) Spatial tessellations. Concepts and applications of voronoi diagrams, 2nd edn. Wiley, Chichester
3. van Kreveld M, Nievergelt J, Roos T, Widmayer P (1997) Algorithmic foundations of geographic information systems. Springer, Berlin
4. Gold CM (1997) Simple topology generation from scanned maps. In: Proceedings of Auto-Carto 13. ACM.ASPRS, pp 337–346.
5. de Berg M, Cheong O, van Kreveld M, Overmars M (2008) Computational geometry: algorithms and applications, 3rd edn. Springer, Berlin
6. Löhner R (2008) Applied cfd techniques: an introduction based on finite element methods, 2nd edn. Wiley, Chichester
7. Zienkiewicz OC, Taylor RL (2000) The finite element method, vol. 1: The Basis. Elsevier, Oxford
8. Hirsch C (1991) Numerical computation of internal and external flows, fundamentals of numerical discretization, vol 1. Wiley, Chichester
9. Hirsch C (1987) Numerical computation of internal and external flows, vol 2. Wiley, Chichester

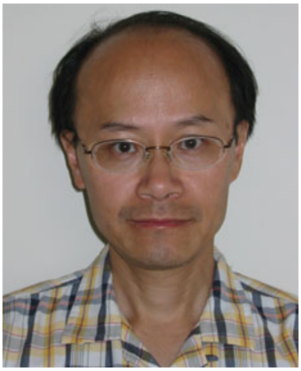
10. Nelson EJ, Jones NL, Miller AW (1994) Algorithm for precise drainage basin delineation. ASCE J Hydraulic Eng 120(3):298–312
11. Yu S, Kreveld MV, Snoeyink J (1996) Drainage queries in tins: from local to global and back again. In: 7th International Symposium on Spatial Data Handling.
12. Camelli F, Wong DW, Sonwalkar M, Löhner R (2007) Coupling computational fluid dynamic models (cfd) and geographic information systems (gis). Paper presented at the 11th Annual George Mason University Conference on Atmospheric Dispersion Modeling, Fairfax, VA, July.
13. Coirier WJ (2001) Development of high fidelity pc based simulator for modeling the atmospheric transport and dispersion of nuclear, chemical, biological and radiological substances in urban areas. CFD Research Corp, Huntsville
14. Coirier WJ, Kim S (2006) Cfd modeling for urban area contaminant transport and dispersion: model description and data requirements. In: Society AM (ed) Sixth Symposium of the Urban Environment, Atlanta, GA.
15. Wong DW, Camelli FE, Sonwalkar M (2007) Integrating computational fluid dynamics (cfd) models with gis: An evaluation on data conversion formats. In: SPIE (ed) Geoinformatics 2007: Geospatial Information Science China. International Society for Optical Engineering. doi:10.1117/12.761763.
16. Clark MM (2009) Transport modeling for environmental engineers and scientists, 2nd edn. Wiley, New York
17. Schnoor JL (1996) Environmental modeling. Fate and transport of pollutants in water, air, and soil. Environmental science and technology. Wiley, New York
18. Boybeyi Z (ed) (2001) Mesoscale atmospheric dispersion. WIT, Southampton
19. Camelli F, Coirier B, Huber A, Hansen O, Kim S, Hanna S, Brown M (2006) An intercomparison of four computational fluid dynamics models: transport and dispersion around madison square garden. In: 14th Joint Conference on the Applications of Air Pollution Meteorology with the Air and Waste Management Assoc, Atlanta, GA. American Meteorological Society, p J2.6.
20. Camelli FE, Hanna SR, Lohner R (2006) Feflo cfd model study of flow and dispersion as influenced by tall buildings in new york city. In: Society AM (ed) 86th Annual Meeting of the American Meteorological Society, Atlanta, p J5.7.
21. Hanna SR, Brown MJ, Camelli FE, Chan ST, Coirier WJ, Hansen OR, Huber AH, Kim S, Reynolds M (2006) Detailed simulations of atmospheric flow and dispersion in downtown manhattan. An application of five computational fluid dynamics models. Bull Am Meteorol Soc 87(12):1713–1726
22. Hanna S, Chang J, Strimaitis D (1993) Hazardous gas model evaluation with field observations. Atmos Environ 27(15):2265–2285
23. Hanna SR, Briggs GA, Hosker RP (1982) Handbook on atmospheric diffusion. Atmospheric Turbulence and Diffusion Laboratory, National Oceanic and Atmospheric Administration, Oak Ridge.
24. Hanna SR, Tehranian S, Carissimo B, Macdonald RW, Löhner R (2002) Comparisons of model simulations with observations of mean flow and turbulence within simple obstacle arrays. Atmos Environ 36(32):5067–5079
25. Coirier WJ, Fricker DM, Furmanczyk M, Kim S (2005) A computational fluid dynamics approach for urban area transport and dispersion modeling. Enviro Fluid Dynamics 5:443–479
26. Camelli F, Löhner R, Hanna S (2006) Vles study of flow and dispersion patterns in heterogeneous urban areas. AIAA Paper 2006–1419.
27. Camelli F, Löhner R, Hanna S (2005) Vles study of must experiment. AIAA Paper 2005–1279.
28. Coirier WJ, Kim S (2006) Summary of cfd-urban results in support of the madison square garden and urban dispersion program field tests. In: Society AM (ed) Sixth Symposium on the Urban Environment, Atlanta, GA.
29. Hsie T, Ward IC (2006) A gis-based method for determining natural ventilation potentials and urban morphology. In: 23rd International Conference on Passive and Low Energy Architecture, Geneva, Switzerland, September 2006.
30. Löhner R, Baum JD, Mestreau E, Sharov D, Charman C, Pelessone D (2004) Adaptive embedded unstructured grid methods. Int J Num Method Eng 60(3):641–660
31. Löhner R (1997) Automatic unstructured grid generators. Finite Elem Anal Des 25:111–134
32. Löhner R (1996) Regridding surface triangulations. J Comput Phys 126:1–10
33. Löhner R (1994) Recent progress in tetrahedral grid generation via the advancing front technique. In: Proc. 3rd International Meshing Roundtable.
34. Löhner R, Parikh P (1988) Three-dimensional grid generation by advancing front method. Int J Num Method Eng 8:1135–1149
35. Anglada MV (1997) An improved incremental algorithm for constructing restricted delaunay triangulations. Comput Graph 21(2):215–223
36. Piegl LA, Richard AM (1993) Algorithm and data structure for triangulating multiply connected polygonal domains. Comput Graph 17(5):563–574

37. Renka RJ (1997) Algorithm 772: stripack: Delaunay triangulation and voronoi diagram on the surface of a sphere. ACM Trans Math Software 23(3):416–434
38. Domiter V, Zalik B (2008) Sweep-line algorithm for constrained delaunay triangulation. Int J Geogr Inform Sci 22(4):449–462
39. Owen SJ (October 1998) A survey of unstructured mesh generation technology. In: Proceedings 7th International Meshing Roundtable, Dearborn, MI.
40. Gold CM (1979) Triangulation based terrain modelling—where are we now? In: Aangeenburg RT (ed) Proceedings, Auto-Carto 4, International Symposium on Cartography and Computing, Baltimore, MD, USA, pp 104–111
41. Gold CM (1999) Crust and anti-crust: a one-step boundary and skeleton extraction. In: Symposium on Computational Geometry. pp 189–196.
42. Cebral JR, Löhner R, Choyke PL, Yim PJ (2001) Merging of intersecting triangulations for finite element modeling. J Biomech 34:815–819
43. Cebral JR, Camelli F, Löhner R (2002) A feature-preserving volumetric technique to merge surface triangulations. Int J Num Method Eng 55:177–190
44. Lee DT, Lin AK (1986) Generalized delaunay triangulation for planar graphs. Discrete Comput Geom 1 (1):201–217
45. Chew LP (1986) There is a planar graph almost as good as the complete graph. Proceedings of the Second ACM Symposium on Computational Geometry, Yorktown Heights, NY, pp 169–177.
46. Chew LP (1989) Constrained delaunay triangulations. Algorithmica 4(1):97–108
47. Sheng X, Hirsch BE (1992) Triangulation of trimmed surfaces in parametric space. Comput Aided Des 24(8):437–444
48. Flato E (2000) Robust and efficient construction of planar minkowski sums. Tel-Aviv University, Tel-Aviv
49. Wein R (2006) Exact and efficient construction of planar minkowski sums using the convolution method. Paper presented at the Annual European Symposium on Algorithms (ESA), Zurich, Switzerland, September 2006.
50. Allwine KJ, Flaherty JE (2006) Joint urban 2003: study overview and instrument locations. Pacific Northwest National Laboratory, Oak Ridge
51. Flaherty JE, Stock D, Lamb B (2007) Computational fluid dynamic simulation of plume dispersion in urban oklahoma city. J Appl Meteorol Climatol 46(12):2110–2126
52. Chan ST, Leach MJ (2007) A validation of fem3mp with joint urban 2003 data. J Appl Meteorol Climatol 46(12):2127–2146
53. Camelli F, Hanna SR, Löhner R (August 2004) Simulation of the must field experiment using the feflo-urban cfd model. In: Fifth Symposium on the Urban Environment, Vancouver, Canada. American Meteorological Society, p 13.12.
54. Löhner R, Yang C, Cebral J, Soto O, Camelli F, Baum JD, Luo H, Mestreau E, Sharov D, Ramamurti R, Sandberg W, Oh C (2001) Advances in feflo. AIAA Paper 2001–0592.
55. Ju L (2007) Conforming centroidal voronoi delaunay triangulation for quality mesh generation. Int J Num Anal Model 4(3–4):531–547
56. Lin H, Batty M (eds) (2009) Virtual geographic environments. Science Press, Beijing

Fernando E. Camelli is Assistant Professor in the School of Physics, Astronomy and Computational Sciences at George Mason University. He received his Ph. D. diploma in Computational Sciences and Informatics from George Mason University in 2002. His undergraduate studies were in the University of Buenos Aires Argentina. His research focuses in transport and dispersion of pollutants in urban settings, and Computational Fluid Dynamics (CFD). He developed algorithms to integrate data in GIS format into CFD models. He has researched the flow around the superstructure of ships for the Navy to help in the design of the HVAC systems of ships, and he studied the flow patterns in the landing decks of ships. He develops algorithms to preprocess the geometry extraction of buildings and complex terrain with almost no user interaction. He is part of the research team on the Center for Blast Mitigation at George Mason University.

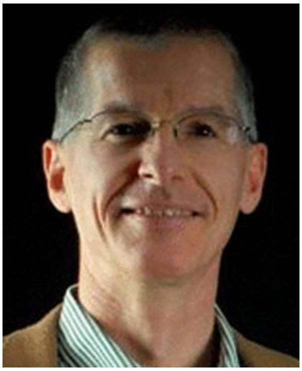
Prof. Jyh-Ming Lien is an Assistant Professor of the Department of Computer Science and is affiliated with the Motion and Shape Computing (MASC) group and the Autonomous Robotics Laboratory at George Mason University. Prof. Lien received his BS degree in Computer Science from the National ChengChi University in Taiwan in 1999 and his Ph.D. degree also in Computer Science from Texas A&M University in 2006. In 2006–2007, Prof. Lien was a Postdoctoral Researcher in the Department of Electrical Engineering and Computer Sciences at the University of California, Berkeley. His research is in the areas of geometric processing, computational geometry, computer graphics, and robotics. His research goal is to develop efficient algorithms for representing, manipulating and analyzing massive geometric data of shape and motion.

Dayong Shen obtained a Ph.D. in GIS and Cartography from Peking University in 2002. He is now a Research Assistant Professor in the College of Science at George Mason University. Dr. Dayong Shen is working on the research project of integrating 3-4D atmospheric data into GIS systems supported by the College of Science, George Mason University. His research interests are mainly data formats, algorithms and visualization of GIS, and the interdisciplinary research of integrating GIS, remote sensing with hydrological and environmental science.

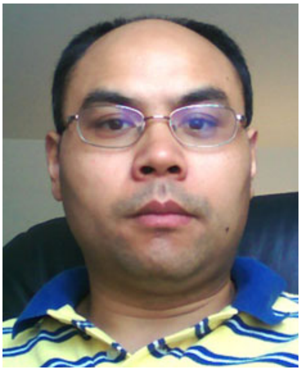


David W Wong is Professor in the Department of Geography and GeoInformation Science at George Mason University. He received his Ph.D. in Geography from the University at Buffalo. His research interests include spatial analysis-statistics, statistical visualization, measuring spatial segregation, and 3D GIS. He serves on the editorial board of several geographical and GIS journals.

Matthew Rice is an Assistant Professor in the Department of Geography and Geoinformation Science at George Mason University. He earned his a Ph.D. in Geography (2005) from the University of California Santa Barbara, where he contributed as a researcher and post-doctoral fellow to the National Center for Geographic Information and Analysis. His research interests are multimodal aspects of geovisualization and geospatial data infrastructure.



Rainald Löhner is the head of the CFD center at the department of computational and data sciences of George Mason University in Fairfax, VA, in the outskirts of Washington, D.C. The CFD center carries out research on all aspects of CFD, with particular emphasis on five strategic areas: blast-structure interaction, contaminant transport in urban environments, free-surface flows, cerebral arterial flow, and optimal shape and process design. Prof. Löhner received a MSc in Mechanical Engineering from the Technische Universit at Braunschweig, Germany, as well as a PhD and DSc in Civil Engineering from the University College of Swansea, Wales. His areas of interest include numerical methods, solvers, grid generation, parallel computing, visualization, pre-processing and fluid-structure interaction. He is the author of more than 500 articles covering the fields enumerated above, a textbook on Applied CFD Techniques, and the principal author of the FEFLO CFD package.



Prof. Chaowei Yang founded and co-directs the Center for Intelligent Spatial Computing (CISC) at George Mason University (GMU). His research has been focused on utilizing spatial principles to optimize distributed computing for enabling physical sciences. He received over 20 grants from a variety of agencies and companies, such as NASA, FGDC, and Microsoft. His 50 peer-reviewed publications appear in a variety of journals, such as the Proceedings of the National Academy of Sciences and the International Journal of Geographic Information Science.