Methodology for Automatic Generation of Models for Large Urban Spaces Based on GIS Data

Metodología para la generación automática de modelos de grandes espacios urbanos desde información SIG

Sergio Arturo Ordóñez Medina[1]*, Jhon Alejandro Triana Trujillo[2]*, Andrés Felipe Padilla Ramos[3]*, José Tiberio Hernández Peñaloza[4]*

(1) M.Sc. Master in Systems and Computing Engineering, se-ordon@uniandes.edu.co
(2) Electronic engineer. Master in Systems and Computing Engineering, ja.triana955@uniandes.edu.co
(3) Master in Systems and Computing Engineering, a-padill@uniandes.edu.co
(4) Ph.D. Informatics ENSTA, Paris, France. Associated professor. jhernand@uniandes.edu.co
(*) Department of Systems and Computing Engineering, Universidad de los Andes, Bogotá D.C., Colombia.

Recibido 30 de septiembre de 2011, modificado 22 de octubre de 2012, aprobado 6 de noviembre de 2012

Palabras claves

Desarrollo de software de aplicación, sistemas de información geográfica, transporte terrestre-planeación-simulación por computadores, desarrollo urbano.

Resumen

En las etapas de planeación y evaluación de proyectos de infraestructura es necesario manejar grandes cantidades de información. Las ciudades son sistemas complejos que deben ser modeladas para ser intervenidas. Estos modelos permitirán medir el impacto de los cambios de infraestructura, simular escenarios hipotéticos y evaluar resultados. Este artículo describe una metodología para generar automáticamente modelos espaciales urbanos desde fuentes SIG. Un diagrama de Voronoi es usado para dividir grandes regiones urbanas, y a continuación definir las zonas de interés. Finalmente, algunos ejemplos de modelos de aplicación son presentados, uno usado para microsimulación de tráfico y el otro para simular contaminación atmosférica.

Key words

Software development, Geographic Information Systems, ground transportation-Planning-computer simulation, urban development.

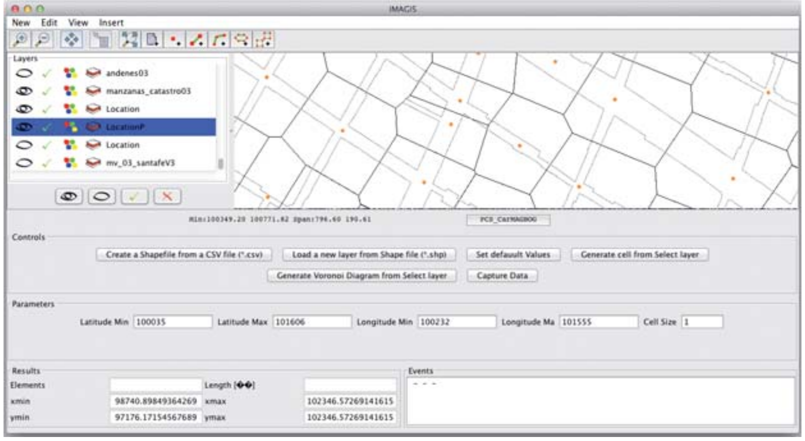
Abstract

In the planning and evaluation stages of infrastructure projects, it is necessary to manage huge quantities of information. Cities are very complex systems, which need to be modeled when an intervention is required. Such models allow us to measure the impact of infrastructure changes, simulating hypothetic scenarios and evaluating results. This paper describes a methodology for the automatic generation of urban space models from GIS sources. A Voronoi diagram is used to partition large urban regions and subsequently define zones of interest. Finally, some examples of application models are presented, one used for microsimulation of traffic and another for air pollution simulation.

Introduction

The spatial data representation of urban systems is a useful tool for the planning and evaluation stages of an infrastructure project; however, it presents several challenges. Such models allow us to measure the impact of changes, simulating hypothetic scenarios and evaluating results. Challenges remain: the data may be heterogeneous, it may be unordered, or may come from multiple sources. The classical Geographical Information Systems (GIS) approach uses layers as data representation units; each layer is related to a subject, and many layers can be overlapped to show a variety of subjects. The classical approach offers overlapping layers for the same space, though when a specific spatial partition is required the data integration may not be that efficient if there are many layers. We propose a data representation model based on spatial partitions, which saves several complex features from the specific urban space. Also the proposed data model offers a hierarchical structure that may be easier to understand than a classical approach.

This paper presents a methodology for the generation of a spatial data model; and two implementations showing use cases of the proposed model's functionality. This paper is or-



Min:100349.20 100771.82 Span:796.60 190.61

Latitude Min 100035 Latitude Max 101606 Longitude Min 100232 Longitude Ma 101555 Cell Size 1

Elements		Length [��]	
xmin	98740.89849364269	xmax	102346.57269141615
ymin	97176.17154567689	ymax	102346.57269141615

Figure 1. Data Processing Application Screenshot.

ganized in the following way: the first section reviews some previous works in urban representation schemes. The second section exposes the methodology for the construction of the model. The third section shows an example of its implementation in an urban landscape and showcases two different modeling applications using our methodology as a basis, an urban traffic simulator and an urban air quality simulation. Finally, some conclusions are presented.

Related work

Geographic Information Systems (GIS) have two types of generalizations: cartographical and model-based generalizations [1]. The former is a constraint-based process used by cartographers to reduce the complexity of maps in scale reduction processes, whilst the latter is used in many application areas for structuring elements for many other cartographic objects [2]. Cartographical and model-based generalizations may be combined in order to help cartographers and urban planners. Some examples of this are shown in this section.

In [2], we present a model generalization approach, based on a computational application of graph modeling principles, which is a functional representation of a city based on a street network and its main objective is retaining the main characteristic elements of a given street network. Vertices represent streets and edges represent intersections. The proposal is validated using a middle-sized Swedish city, Sätra. CityGML is an open data model for the representation of virtual 3D city models [3] and is based on the Geography Markup Language. CityGML is an international standard issued by the Open Geospa- tial Consortium (OGC). It includes object semantics added to the shape and graphical appearance of city models. The semantic features include entities to support navigation, tourism, environmental protection, architecture, and urban planning.

Model and definitions

Partition of the Urban Space for Scalability and Distribution

The following model proposal is made in order to achieve a design of an urban space, supporting distribution and scalability, allowing the development of a microsimulation capable of handling entire cities or big portions of them depending on the processing capacity available. Hence, the overall time for the simulation depends on the most complex, and therefore more computationally expensive partition of the region of interest.

Definition and Generation:

We chose road intersections as computational units of the model, because it is precisely at such intersections that the most complex mobility phenomena take place. For this reason, the partition of the urban space is based on two-dimensional points which represent each intersection in the road network. A Voronoi diagram [4] is constructed based on these points, generating a convex polygon around each intersection. Before building the polygon, clustering algorithms based on the Euclidian distance may be applied to the intersection points, avoiding an excessive division of the urban space (for example, a cloverleaf interchange). The area enclosed by each polygon may be simulated in separated processors. Fig. 1 shows the Voronoi polygons drawn over the GIS data.

Handling the Distribution Issues (shared regions):

In order to synchronize the results between distributed simulations, a method of sharing information between neighbor intersections is required. Therefore, shared regions have been defined between every pair of adjacent polygons, and they are calculated automatically in the data entry step.

There are two types of shared regions associated to a Voronoi polygon. The first is a subset of the polygon located near to its boundaries. Changes on the variables inside this subset

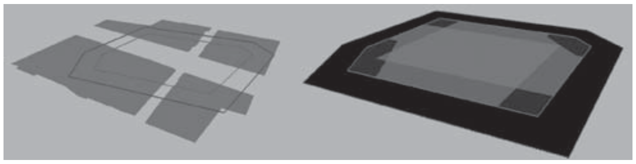


Figure 2. Urban Mobility Simulator. Voronoi Polygon and Shared Regions

must be replicated to the corresponding neighbor intersection. The second type is a region outside the polygon near to its boundaries. This space is controlled by the neighbor polygons and it must replicate the information of the neighbor polygons. Fig. 2 illustrates the region encased in the Voronoi polygon (dark grey) and the shared regions with neighbor polygons (white).

The automatic generation of the shared regions is controlled by a reference distance, which determines the size of the shared regions. A group of parallel lines to the Voronoi polygon at reference distance to each side of it defines a corresponding enlarged, shared polygon. The intersections between each neighbor enlarged polygon and the current polygon define the first type shared regions. The intersections between the current enlarged polygon and each neighbor polygon define the second type ones.

Urban Model: Structuring the Voronoi Polygon

We propose three categories of elements for describing the content of a polygon: zones, stretches networks and point objects.

Zones:

A zone is a two-dimensional connected division of the Voronoi polygon, so it is not necessarily a convex polygon. Zones represent abstractions on the urban space; they have special characteristics which make them important to urban modelers. Zones do not define partitions over the Voronoi polygon; there are points in the polygon which do not belong to any zone, and there are points which belong to multiple zones. The sources for these zones are specific layers from GIS files that contain polygons or multi-polygons, and sometimes lines or polylines; the sources also hold important features such as cadastral registers. Next, we describe the principal zone types:

- Block: Blocks are the smallest area surrounded by streets.
- Property (land lot): Properties are divisions of a block. There may be buildings on them; GIS data contains the height of buildings on each property, if there are any.
- Street: Streets are areas where vehicles may move. Streets do not always belong to the road network.
- Sidewalk: Sidewalks is a pedestrian's pathway located along a street.
- Plaza (Town Square): Plazas are public open spaces, where people may gather and interact.
- Horizontal signs: The horizontal signs are two-dimensional regions that affect the mobility of mobile actors from one or more types. Generally they are marked on the pavement.
- Anomaly: Anomalies represent urban areas where order has been broken. They may refer to, for example, structural damage on a street.

Stretches and Stretches Networks:

A stretch (also called segment) is a zone where mobile actors may move. A stretch has a direction of movement, which indicates the course of movement of mobile actors using the stretch. A stretch may even have two directions of movement, like a highway with more than two lanes.

A stretch network is a composition of stretches. Stretch networks determine the movement space for mobile actors (for example, automobile networks, pedestrian networks, subway networks, etc.) Hence, a stretch network is also associated to a mobile actor type. The stretches data sources are lines and polylines with direction of movement, common nomenclature, number of lanes, etc.

Point Urban Objects:

A point urban object represents the space positioning of an urban object of interest. Examples of objects of interest which may be represented in this way are trees, street lamps, utility poles or sculptures. However, we identified vertical traffic signs as the most relevant point objects, because of their influence on urban mobility. We identified two types of such signs.

- Passive vertical signs: passive vertical signs do not change over time. Examples of this kind of sign are warning signs and restrictive signs.
- Active vertical signs: active vertical signs present a programmed behavior. Examples of this kind of signal are traffic lights and railroad signs.

Urban space methodology: implementation and usage

Formal Methodology

Using the following process enables the creation of an urban space model based on geographic information from a Geographic Information System (GIS). There are optional steps because not all simulators need all the steps; however, many simulations require discrete representations and/or shared regions.

Function Definitions:

- *degree(*v*): returns the degree of vertex* v
- *euclidDistance (*p_i*,* p_j*): returns the euclidean distance between points* p_i *and* p_j

- *isConvexPolygon(r): returns true if polygon r is convex, false otherwise*
- *squareTiling($\mathbb{R}^2$): defines a regular squared lattice partition over a plane region*
- *center (c): calculates the center of the square c*
- *distEPPol (p,pol): calculates the lowest distance between a point and a the polygon.*

Methodology:

I. A *stretch* is a straight segment on a road network.

II. Each road network stretch is represented by a line segment S_i, where is the segment set. Let points p_{Ai} and p_{Bi} be end points of S_i, then only two spatial dimensions will be considered in their representation (latitude and longitude).

III. For $(\forall S_i, S_J \in S|(\exists p_M|p_M \in S_i \wedge p_M \in S_j))$ four line segments will be created and added to S with end points S_{Ai}: (p_{Ai}, p_M), S_{Bi}:(p_M, p_{Bi}), S_{Aj}: (p_{Aj}, p_M), S_{Bj}:(p_M, p_{Bj}). Then S_i and S_j are eliminated from S, and any segment with the same initial and end points is eliminated from S also.

VI. A planar graph $Gv < A,V >$ is built, where $V = \{\nu: (\exists S_i \in S|p_{Ai} = \nu \vee pBi = \nu)\}$ and $A = \{a:(\exists S_i \in S|\{p_{Ai}, p_{Bi}\}=a)\}$. Thus, $(\forall \nu \in V|degree(\nu) > 0)$.

V. *(Optional)* A clustering algorithm is applied to V grouping elements of V with elements of A in common and with a small Euclidean distance. A geometric average point is calculated for each group and it is added to V. Then, the points of each group are eliminated from V. The elements of A whose two vertices were eliminated are also eliminated. The elements of A whose one vertex was clustered, replace the vertex with the correspondent average point.

VI. A partition over V is defined in the following way:

- End Points = $\{\nu : \nu \in \mathrm{V} \wedge degree(\nu) = 1\}$
- Unions = $\{\nu : \nu \in \mathrm{V} \wedge degree(\nu) = 2\}$
- Intersections = $\{\nu : \nu \in \mathrm{V} \wedge degree(\nu) > 2\}$

VII. A Voronoi Diagram $Vor(V)$ is built defining on the planar graph V planar regions: $R = \{r: (\forall p \in r|p \in \mathbb{R}^2 \wedge (\exists \nu \in V|(\nexists w \in V|euclidDistance(w,p) < euclidDistance(\nu,p))))\}$. These regions are a partition over $\mathbb{R}^2$. With such a definition, we can assert the existence of two bijective functions between elements from R and elements from V, $point(r)$and $region(\nu)$. Construction ensures $(\forall r \in R|isConvexPolygon(r)\}$, $(\forall r \in R|(\nexists w \in V|w \neq point(r) \wedge w \in r))$ and $(\forall r \in R|(\exists p \in r|(\exists a \in A|isNode(a,point(r)) \wedge p \in a) \vee (\exists a \in A|\neg isNode(a,point(r)) \wedge p \in a)))$.

VIII. A set of two-dimensional regions (*zones*) are created over each element of R. This defines the function $zones(r)$. Zones are polygons where $isConvexPolygon(zones)$ is not always true. A zone is defined by a functional characteristic of the city, for example blocks or parks. As such, zones' definitions could be expanded when a functional characteristic of the city needs/wants to be exposed. Furthermore, zones could overlap, so they are not a partition over an element of R.

IX. *(Optional)* Another partition is defined over $\mathbb{R}^2$. In this case, it is a regular squared lattice defining $C = \{c : c \in squareTiling(\mathbb{R}^2)\}$. If the phenomenon of interest must be modeled in three dimensions a cubic lattice can also be defined.

X. *(Optional)* A partition over C is defined as follows: $I = \{i: (\forall c \in i|c \in C \wedge (\exists r \in R|center(c) \in r))\}$. Thus, each element of I defines a discrete polygonal area related to an element of R. This defines the bijective function $discretePolyg(r)$ that returns the corresponding element from I.

XI. *(Optional)* Each element of C, from an I element related to a region r, is related to a set of elements of $zones(r)$. $zones(c) = \{z:(\exists r \in R, i \in I|z \in zones(r) \wedge i = discretePolyg(r) \wedge c \in i \wedge center(c) \in z))\}$

XII. *(Optional)* Shared regions are generated as follows:

- $E = \{e: (\forall c \in a|c \in C \wedge (\exists r \in discretePolyg(r) \vee distEPPol(center(c),r) < D))\}$. E is not a partition of C. Each element of E is a discrete polygon related to an element of R. This defines the bijective function $enlargedDiscretePolygon(r)$ that returns the corresponding element from E.
- Taking two different elements, or R called $r1$ and $r2$, two regions are defined between them as follows:

$$Sh1(r1,r2) = \left\{ \begin{matrix} c{:}c \in C \wedge c \in discretePolygon(r1) \wedge \\ c \in enlargedDiscretePolygon(r2) \end{matrix} \right\}$$

$$Sh2(r1,r2) = \left\{ \begin{matrix} c{:}c \in C \wedge c \in discretePolygon(r2) \wedge \\ c \in enlargedDiscretePolygon(r1) \end{matrix} \right\}$$

- Hence, the first shared region defines elements of C that belong to one polygonal region 1, but they must share the information with another polygonal region $r2$. And the second shared region defines elements of C that despite belonging to another polygonal region $r2$, must update their information to the region $r1$.

Implementing the Model

A sample application extracting the data required for our model was made using GIS data from the Colombian city of Bogotá. The implementation was divided into three steps: data entry, data processing and data persistence. The application was developed using Java for the data entry and data processing steps and MySQL for the persistence step. A preliminary step is required to guarantee the data quality and the conflicting data points from the different data sources. To accomplishing this, interactive software tools were developed to manually alter the GIS data.

The data entry step reads the GIS data containing the required information, using an open source GIS processing API, GEOTOOLS [5]. GEOTOOLS opens GIS files and manipulates their data as a Java collection. After loading the required files, the data processing step transforms the original GIS data to generate the nodes for creating the Voronoi diagram and the zones are structured into Voronoi polygons and finally, specialized objects are created to contain the processed data. Some basic operations such as data format conversions, concatenation, grouping or relating are necessary for data processing. Fig. 1 shows the data processing application in action, after dividing the urban space in Voronoi polygons. Finally, in the data persistence step, the generated objects are persisted using the DBMS MySQL.

Using the Model: Urban Mobility Microsimulation

We are developing an urban traffic microsimulator in order to introduce psychological traits and other behavior patterns in mobile agents; for example, motor vehicles and pedestrians. Microsimulations are computationally expensive because they simulate a great quantity of agents, so an adequate space partitioning method is required for the distribution of each space partition to available processors [6]. Therefore, our urban model is useful for these kinds of simulations. Each Voronoi polygon and its related agents may be processed by a different CPU. Microsimulations involve dynamic elements known as agents; each of these agent needs to know its position and its surroundings in order to calculate each step in the simulation. To facilitate this a discrete Voroni polygon is used which generates a series of equal-sized square partitions known as *cells*. Each cell contains information about overlapping zones, signs and mobile actors. Fig. 3 shows a Voronoi Polygon with information relating to blocks and streets. It also shows the discrete partitioning in cells 0.5 meters wide.

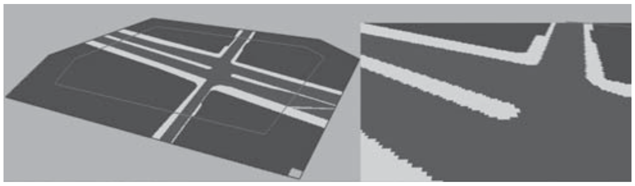


Figure 3. Right, Urban Mobility Simulator. Left, Cell Partitioning.

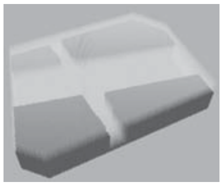


Figure 4. Urban Air Quality Model, Voronoi Polygon Rendered Buildings.

Using the Model: Urban Air Quality Modeling

We are also developing an Air Quality Model (AQM), in order to provide a Computer Fluid Dynamics infrastructure based on the Lattice Boltzmann Method (LBM), to an air quality simulator. The AQM needs information about the land lots and the buildings on them, and the streets on each Voronoi polygon, in order to feed the LBM with a group of solid obstacles which oppose the air flow. It is planned for the AQM to be supplied with sources of pollutants from the urban mobility simulator, using the automobile vehicle actors. Fig. 4 illustrates a Voronoi Polygon with rendered buildings from the land lots data and air parcels over the streets space.

Conclusions

A methodology for generating an urban model based on spatial partitions of existing GIS data was proposed in order to support distributed applications, for example, micro simulation-based models for urban phenomena.

The strategy of using intersections to systematically divide the city GIS representation and the use of Voronoi diagrams have offered good results. The geometric features of Voronoi diagrams allow us to extend the model naturally in order to add in concepts such as neighbor intersections and shared zones.

The presented implementation supports the construction of models for distributed simulations. Two examples of such simulations were presented: a microsimulation of urban mobility phenomena and an air quality simulation. Whatever potential this automatic approach may have, it also exposes the need for interactive software tools that can help us complete missing data from the GIS data frequently available in developing countries.

References

[1] J.C. Muller, et al. *GIS and Generalization: Methodology and Practice*. London: Taylor and Francis. 1995, pp. (19-56).

[2] B. Jiang, C. Claramunt. “A Structural Approach to the Model Generalization of an Urban Street Network”. *An International Journal on Advances of Computer Science for Geographic*.Vol., 8 No. 2 June, 2004, pp. 157- 171

[3] T.H. Kolbe. “Representing and Exchanging 3D City Models with CityGML”. In *3D Geo-Information Sciences*. J. Lee & S. Zlatanova (Eds.). Berlin: 2009 (pp. 15–31).

[4] M. De Berg. *Computational Geometry: Algorithms and Applications*. Berlin: Springer-Verlag, 2000. pp (147-162).

[5] “GeoTools, The Open Source Java GIS Toolkit”. Geo Tools. August 20, 2010. Available: http://www.geotools.org/.

[6] J. Ibarra, J. Hernández, S. Ordoñez. (Nov, 2009). “Multi-Modal Simulation for Urban Mobility Analysis: An Approach Based on a Model of Behavior and Infrastructure-related Anomalies”. En: *13th Congress of Iberoamerican Society of Digital Graphics (SiGraDi 2009)*.