

## MODELING CITIES FOR 3D\_GIS PURPOSES

E. G. V. de Jesus<sup>1</sup>, A. L. de Amorim<sup>2</sup>, N. J. Groetelaars<sup>2</sup>, V. O. Fernandes<sup>1</sup>

<sup>1</sup> Dept. of Transport and Geodesy Engineering, Federal University of Bahia, Aristides Novis Street, 2, Federação, Salvador, Bahia, Brazil - (elainegomes623, vivian.deofernandes@gmail.com)

<sup>2</sup> Architecture Faculty, Federal University of Bahia, Caetano Moura Street, 121, Federação, Salvador, Bahia, Brazil - (alamorim, natgroet)@ufba.br

Commission IV, WG IV/1

**KEY WORDS:** Geometric modeling, CityGML, 3D GIS, QGIS, Rhinoceros 3D, FME.

### ABSTRACT:

3D Geographic Information Systems (3D GIS) are systems that are capable of making spatial analyses that consider the tridimensional and semantic representation of objects. These systems make these analyses through its planialtimetric coordinates. The City Geography Markup Language (CityGML) is used for the representation of cities and urban applications. The CityGML is an international standardized data model based on XML used to store and exchange information through 3D representation of cities. This standardized data model has 5 Levels of Detail – LOD, varying from LOD 0 (least detailed) to 4 (most detailed). The main challenges for the implementation of these systems refer to the techniques used for obtaining data and the data format, and also all the software used in the geometric modeling of the urban model. The data related to the buildings were manipulated with the QGIS software in this study. This made it possible to obtain the height of the buildings by the elevation difference between the Digital Surface Model and the Digital Terrain Model. This paper presents and discusses the first results of the geometric modeling made in the campus of the Federal University of Bahia (UFBA), by using airborne laser scanning data, integrating QGIS, Rhinoceros and CityGML.

### 1. INTRODUCTION

The Geographic Information Systems (GIS) have many urban applications and may be associated to 2D, 2.5D, 3D and 4D visualizations. The 2D applications use two-dimensional objects and its respective coordinates in the 2D space. In a 2.5D application, there is a 3D information associated, represented by the color attribute from a Digital Elevation Model (DEM). The 3D applications enable the visualization of X, Y and Z representations. Lastly, the 4D applications allow the temporal study of an area through variables associated with chronology in a tridimensional representation.

Some studies have been developed to integrate GIS to the management of university campuses. Bansal (2014) for instance, used the Arcgis and Sketchup (GIS/CAD) tools in the case study of the National Institute of Technology (NIT) in Hamirpur, India. The study integrated analyses based on location, semantic attributes, digital terrain model (topographic surface), building topology, spatial and non spatial analyses in a single 3D platform.

Geospatial models of cities associate strict geometric representations with an alphanumeric database. This makes it closer to the real world by combining geometry (shape), appearance and semantic. In the geometric modeling of an urban space it is necessary to define the Level of Detail (LOD), which expresses the accuracy and reliability with which the elements must be represented and the element classes to be included (REDWEIK et al., 2007).

In this context, there are differences in the use of the term LOD for GIS and in Computer Graphics. According to Biljecki (2017), the LOD of a geometric model of a city refers to the

adhesion degree in relation to reality, while in Computer Graphics there must be a data simplification (a process of generalization), and also not consider the semantics of the objects as in GIS.

The tridimensional representation of cities is a process that involves many variables, especially in regards to: many data acquisition techniques, a variety of modeling tools, interoperability problems among them, and the usage of different techniques and patterns.

Cities are increasingly adopting 3D city models. Providing further value and additional utility over 2D geo-datasets, 3D city models are becoming ubiquitous for making decisions and for improving the efficiency of governance. Local governments use 3D city models for urban planning and environmental simulations such as estimating the shadows cast by buildings, investigating how the noise from traffic propagates through a neighbourhood, and predicting how much solar irradiation a roof of a building receives in order to assess whether it is economically feasible to install a solar panel (BILJECKI, 2017).

This paper presents the first results of the modeling of the Ondina campus – UFBA, performed from airborne laser scanning data. This model is compatible with LOD 1, aiming to test and integrate the usage of the CAD and GIS tools and its compatibilities with the CityGML pattern. It is worth mentioning that this first modeling is a basic step for the implementation of a 3D\_GIS in UFBA.

## 2. TECHNIQUES FOR OBTAINING DATA FOR GEOMETRIC MODELING OF CITIES

As a source of data for geometric modeling of cities, Bobkowska et al. (2017) highlight the Light Detection and Ranging (LIDAR) laser scan as a proper technology for data acquisition, for its great precision and short fieldwork time.

Laser scanning is a data capture system that allows the collection of three-dimensional coordinates of surface points and objects over it. This process is based on the time and intensity of emission/return of the laser pulse, which allows the calculation of the distance between the reflection point and the sensor. This system allows the acquisition of information on the landscape, the buildings, as well as other objects such as vegetation, paths, posts etc.

Point clouds are a kind of data representation that are efficient and effective, for Digital Terrain Models (DTM) and Digital Surface Models (DSM) can be elaborated from them. This data may be used in projects designed for the elaboration/updating of a cartographic base, the obtaining of the Z quota for the creation of geometric models of cities and many other urban applications.

According to Centeno and Mitshita (2007), the production of digital models of the landscape faces a problem with the identification of the features that correspond to the landscape from those which are above it, such as buildings and vegetation. One of the ways of obtaining the height of the objects that are above the landscape is to perform the normalization of the Surface Digital Model, namely, to calculate the difference between the SDM and the TDM and thus directly obtain the value of the heights of the buildings only.

The choice of the technique to be used is also related to the level of detail that is aimed to attain. Akmaliaa et al. (2014) highlight that the data obtained by the airborne LIDAR technology may be used to generate models that are compatible with LOD0, LOD1 and LOD2, while the close range photogrammetry (or terrestrial) are more suitable for generating detailed models in LOD3 and LOD4. Owing to the availability of the point cloud of the city obtained by the airborne LIDAR scan, the modeling of the campus compatible with LOD1 was woven for the first experiment.

## 3. PATTERNS AND FORMATS FOR THE GEOMETRIC AND SEMANTIC MODELING OF CITIES

In relation to the data pattern for modeling, Zlatanova et al. (2012) highlight some of the most commonly used formats and its characteristics.

- The Virtual Reality Modeling Language (VRML) was released in 1995 and accepted as default by the Web 3D consortium. It was conceived as a web default for the exchange of graphic data and it consists of a language for modeling 3D scenery. It results in big archives for models of great extent for not being based on XML.
- After 1998, the Web 3D consortium adopted the X3D format based on Extensible Markup Language (XML). It was an improvement of VRML, despite being less used than it. Of the little software available for visualizing graphic data on the internet, almost no one allows the import and export in X3D.

- The Keyhole Markup Language (KML) is encoded in XML and it is used for viewing objects on Google Maps and Google Earth. The KML was developed by Keyhole Inc. that was posteriorly acquired by Google. KML is recommended for web applications and it is widely accepted. However, the format does not store the semantics of the objects.
- The Collaborative Design Activity (COLLADA) constitutes a format for the interchange of 3D objects and offers resources to describe the geometry, the topology and texture, but it does not store the semantics.
- The Geography Markup Language (GML), object of the rule ISO 19136, is a pattern created for the exchange of geospatial data using a XML structure. The GML geometry model follows the ISO 19107 pattern and so it provides classes for geometric primitives from 0D to 3D. The GML 3 includes spatial and time references, as well as topology, metadata and semantics.

The City Geography Markup Language (CityGML) is an international pattern for an open data model. It is based on a XML language and it aims to store and develop applications that involve three-dimensional representations of cities. This pattern was elaborated by the Special Interest Group 3D (SIG3D), which is a work group that deals with modeling, storage, manipulation of landscapes and 3D city models.

The first version of the CityGML was submitted to the Open Geospatial Consortium (OGC), an international organization that aims to create interoperability and quality patterns for the geospatial application community. The 2.0 version was approved in 2012 and it is currently in process of completion of the CityGML 3.0 version. It shall be released by the end of 2018 or the beginning of 2019 (KUTZNER; KOLBE, 2018). The aim of CityGML is to reach a common definition (OpenGIS) of the basic entities, the attributes and relations of a three-dimensional city model.

According to Groger and Plimer (2012), the CityGML defines the three-dimensional geometry, the appearance, the topology and the semantics of the most relevant kinds of objects in urban or regional contexts. That is opposed to the purely geometric/graphic models such as Keyhole Markup Language (KML), Virtual Reality Modeling Language (VRML) and X3D, which do not possess significant semantic attributes.

CityGML supports different levels of detail, and they may vary from LOD0 to LOD4. In LOD0, the model only represents the surface of the landscape in the city scale. In LOD1, the perimeter and height of the building are represented by the extrusion of the polyhedral. In LOD2, the characteristics of the buildings such as roof shapes and textures are present. In LOD4, there is a building scale, which has a better level of detail on the outside, with ledge modeling and other details. Lastly, in LOD4, the inside of the buildings are represented as internal divisions and its equipments with rich detail. This representation can be seen in Figure 1.

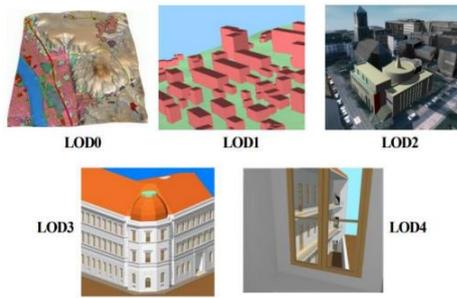


Figure 1. CityGML five levels of detail  
 Source: GRÖGER et al., 2012

#### 4. TOOLS USED IN THE GEOMETRIC MODELING OF CITIES

Various software programs for geometric modeling of cities are found in the literature, such as SketchUp and its CityEditor plugin, the CityEngine and Rhinoceros with the Heron, Meerkat and Rhinocity plugins.

The SketchUp (Trimble) is a geometric modeling software, and its 3DIS GmbH's CityEditor plugin allows the import, editing and export of CityGML models. This tool imports: CityGML models; terrain models; textures; vector data models (points, lines and polygons) in SHP, DXF, GML, JSON, GOEJSON formats; point clouds and triangulated-surface models of PLY and OBJ archives. It enables editing, addition and exclusion of attributes for buildings and surfaces and the exportation of objects for CityGML objects (3DISGmbH, 2016).

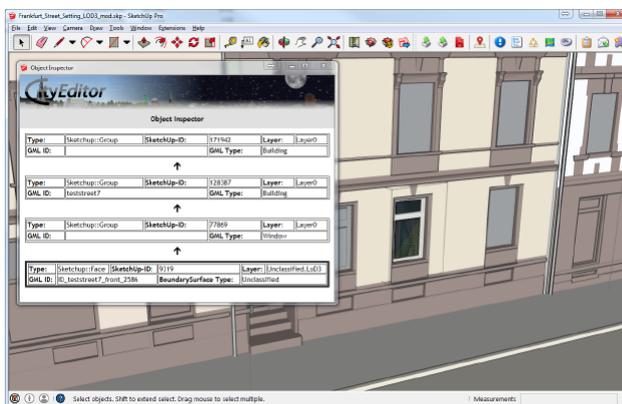


Figure 2. CityEditor plugin interface  
 Source: 3DISGmbH, 2016.

The Grasshopper is a Rhinoceros (Rhino Software) plugin and has been used for many applications, including the ones referred as “parametric urbanism”. The add-on Heron and Merkat (that are Grasshopper plugins) can be used for 3D modeling of cities. They also enable the import of shapefile (SHP) data format, which is one of the most popular formats in GIS and allow the extrusion of the building polygons from their heights, as shown in Figure 3.

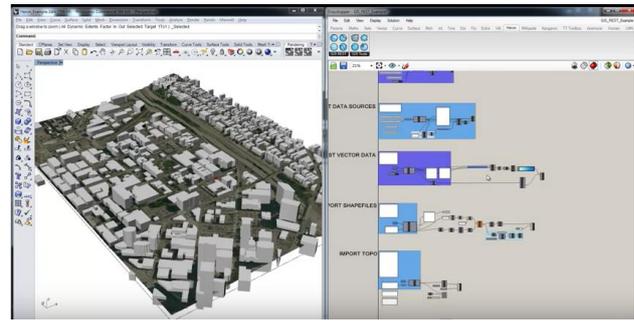


Figure 3. Grasshopper's Heron plugin interface  
 Source: Washburn, 2014

The CityEngine is an ESRI software used in urban modeling that is integrated with the GIS tools and can be used in planning, operations and monitoring. This tool uses the Computer Generated Architecture (CGA) programming language to generate models through a set of predefined rules. The CityEngine directly imports Shapefile (SHP), File Geodatabase (GDB), KML and OpenStreetMap (OSM) formats. This software interface can be seen in Figure 4.



Figure 4. CityEngine interface  
 Source: Sohail, 2013

Many works have been developed on the usage of these software programs for urban planning, parametric urbanism and parametric modeling. Wendel et al. (2017) and Adamex (2015) worked on papers on the Cityeditor software. On the other hand, Chen et al. (2017), Mota et al. (2017), Santos et al. (2016), Bini and Galafassi (2017) performed researches that involved the Rhinoceros software and its plugins, turned to urban modeling. The experiments carried out with the CityEngine software can be verified in the papers Santana et al. (2013), Lima and Freitas (2015).

However, according to Zlatanova et al. (2002), it is verified that the software programs that manipulate 3D objects have specificities, that is, a software program is used to store data, another to visualize and another to convert. That means that 3D\_GIS tools that are totally integrated, capable of structuring, manipulating, analyzing, representing and converting 3D into formats that are adequate to geometric and semantic modeling of cities were not yet developed.

Due to these challenges the website [https://www.citygml.org/] was created by 3D Geoinformation Group in the Delft University of technology. It gathers information on 3D cities, CityGML, CityGML tools etc. The CityGML software

programs are assorted in many categories from which the CityGML model generator and the CityGML model viewer stand out.

For the software programs that generate city models (geometric and semantic) in the CityGML format, the following are highlighted:

- The RandomCity consists of a modeling feature to generate buildings in many levels of detail (LOD) Multi-LOD CityGML, aiming to determine the ideal LOD for a given spatial analysis. This software was developed as a part of Biljecki’s (2017) doctoral research in Delft University of Technology.
- The OSM2CityGML uses a XML structure of the OpenStreetMap buildings and converts it in a CityGML format with the Feature Manipulation Engine (FME) software.
- The 3DFIER uses data from 2D\_GIS to extrude the polygons from its height (obtained from the LIDAR point cloud) and generates CityGML models.

For CityGML visualization models, the following are highlighted:

- The Elyx 3D View represents a tool with an emphasis on applications such as urban planning, access and mobility, traffic control together with the reduction of CO2 and noise emission, with the usage of 3D data. It can be used to manage, visualize, edit and analyze 3D models and data.
- The FME consists of a conversion and data transformation tool developed by Safe Software, to solve data interoperability problems. It can be used for reading, writing, processing and visualizing data in the CityGML standard.
- The FZK Viewer visualizes IFC and CityGML semantic models. It was elaborated by the Karlsruhe Institute of Technology’s (KIT) Institute for Automation and Applied Informatics.

## 5. STUDY AREA

The study area represents the Ondina-Federação campus of the Federal University of Bahia (UFBA) in Salvador, Bahia, Brazil. The campus is comprised of an area of approximately 60ha, and contains 75 buildings that include:

- 15 university units;
- 6 classroom complex buildings;
- 2 dining halls;
- 2 libraries and
- 2 banks.

The university units are in agreement with the statute and regiment of UFBA (2010), which are implementing bodies of the academic activities and the teaching and administrative staff. These units comprise of two modalities: Colleges or Schools teaching units, research and extension defined by its aim to graduate in academic careers in general scientific fields or disciplinary, multidisciplinary and interdisciplinary fields of knowledge.

All the other buildings are appendices of the university units and buildings in which administrative functions are practiced. The Figure 5 shows the location of the study area.

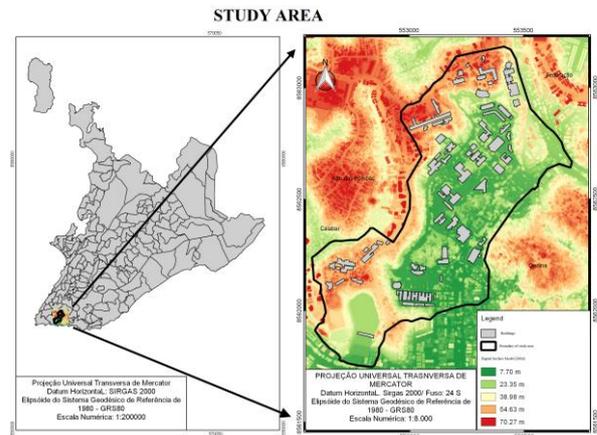


Figure 5. Location of the Study Area

## 6. METODOLOGY

The following resources were used to carry out this work:

- The base map of the Cartographic System of the Metropolitan region of Salvador (SICAR, 1992) in a scale of 1:2000; and Cartographic and Cadastral System of the County of Salvador (SICAD, 2006) in a scale of 1:2000, from which the buildings footprints were extracted;
- Footprints of the campus buildings updated in 2017 by mapping images obtained with Unmanned Aerial Vehicle (UAV), made in the DETG-UFBA scope by Elias (2017);
- Semantic data on the campus buildings extracted from the websites of UFBA’s university units;
- Point cloud (with approximately 10 points/m<sup>2</sup>), Digital Terrain Models (DTM), Digital Surface Models (DSM) and Orthophotos from Airborne Laser Scanning;
- QGIS software add-ons: “Point Sampling Tools” used to assign to the centroids of the bases of the buildings the values of their heights. This corresponds to the differences between the DSM and DTM in the considered point and “Qgis2ThreeJS” for the extrusion of the buildings perimeters;
- Rhinoceros software and its plugin (Grasshopper) and the Meerkat and Heron addons used for the geometric modeling of the buildings;
- The Feature Manipulation Engine (FME) software for the data conversion.

A compilation of the data was made for the base map elaboration. That is, the vector databases SICAR (1992), SICAD (2006) and Elias (2017) were gathered. The location of the building footprints was verified and some of the buildings that were not on the base map were vectorized through orthophotos, in the QGIS software.

After that, centroids were generated for each building footprint (in the SHP format). They were assigned the height value of the terrain and of the buildings based on DTM and DSM respectively, through the QGIS’s “point sampling tools” complement.

The DSM normalization was made to determine the height of the buildings, by calculating the DSM height difference in relation to DTM's (Figure 6).

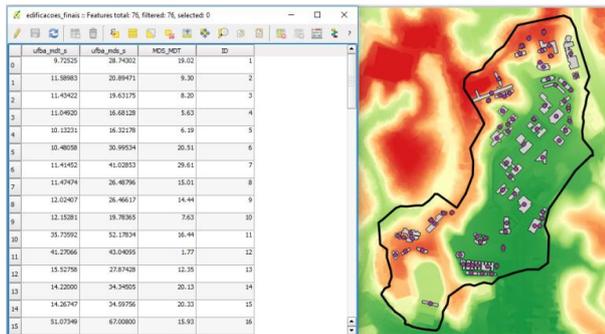


Figure 6: Values obtained from the DSM normalization for each centroid

The average height attributed to the centroids and the semantic data in CSV format containing the name of the university unit, the knowledge area, graduation course, post-graduation courses and the number of classrooms were added to the shapefile archive of the buildings footprints.

A geometric modeling of the buildings was made in the QGIS software by using the Qgis2ThreeJS add-on. This add-on allows the extrusion of the buildings footprints based on their medium height and it enables to add semantics to the objects. It is necessary to choose the field based on which the building heights will be attributed, according to Figure 7.

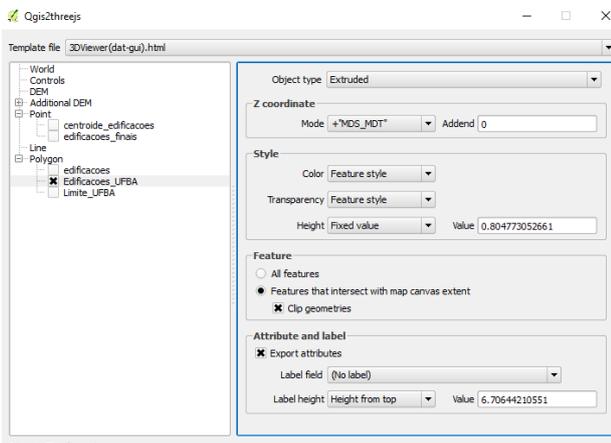


Figure 7: Extrusion of the buildings from the corresponding Z coordinate

The geometric modeling was done by the 3D Rhinoceros through the Heron and Meerkat plugins. It seeks to investigate the most adequate for modeling from shapefile archives from the SIG. The Heron and Meerkat are addons that allow the integration of Rhinoceros and Grasshopper, and they use modeling from many formats of data, including .shp.

The Heron plugin modeling was done from the Heron Example archive, available on the website [https://www.food4rhino.com/app/heron] and includes the following steps: 1) Geocoding addresses through the “Geocode” component. This uses the address of the location and converts the coordinates X and Y; 2) Defining a point of origin for the modeling aiming to minimize the distortion in the transformation of a plane curve; 3) obtaining images through

online searches in the USGS database. However, it is possible to search for images from sources like Google Maps and Bing; and 4) Import the shapefile archive with the polygons related to the buildings in the study area that enabled the creation of the modeling.

The Meerkat plugin modeling is simpler than the Heron plugin modeling. It only allows the usage of shapefile archives from the user and the usage of Google satellite pictures. The shapefile coordinates were used as a limit and the extrusion of the polygons was done with the “Extrude” component from the shapefile field corresponding to the Z coordinate. This procedure was performed using the Meerkat plugin tutorial, available on the website [https://github.com/CenterForSpatialResearch/gis\_tutorials/blob/master/14\_Downloading\_Spatial\_Data\_In\_Grasshopper\_Using\_Meerkat.md].

Lastly, the FME software was used to convert the model produced with the Rhinoceros, resulting from the modeling with Heron and Meerkat for the CityGML format.

## 7. RESULTS AND DISCUSSION

The results presented consist of the performing of the geometric modeling of the buildings done in the QGIS and the Rhinoceros, from which its model was transformed to CityGML, as it is presented below.

### 7.1 Modeling performed in QGIS

The geometric modeling done through the Qgis2ThreeJS complement in QGIS showed that the terrain presents altitudes that vary from 8 to 59m, and the buildings reach up to 30m of height. The modeling was made using the planimetry of the terrain and the buildings (Figure 8) and considering the Digital Terrain Model (DTM) verified in Figure 9. Both of them used the values corresponding to the Z coordinate for the buildings extrusion. It was possible to verify that the modeling meets the requirements of the CityGML model for LOD 1, enabling the identification of the semantic information (Figure 8) and aggregating geometry, appearance and topology.

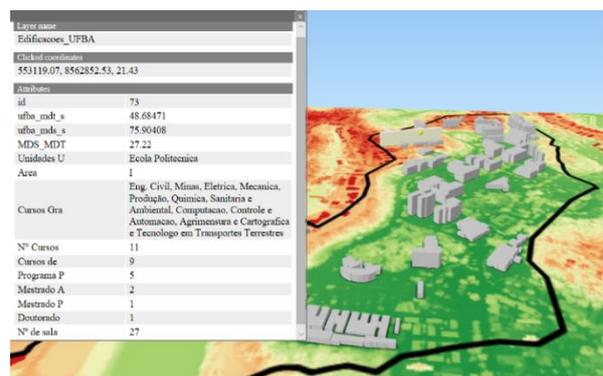


Figure 8: Query performed on the geometric model indicating the attributes of the features

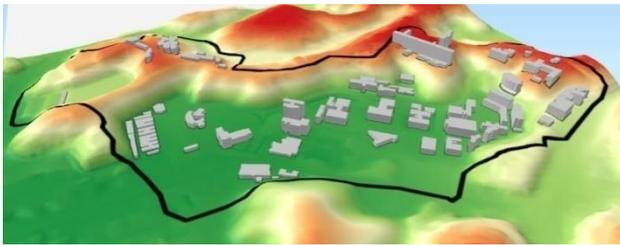


Figure 9: The result of the geometric modeling from the DTM and the planimetric survey of the campus

In spite of being possible to add semantics and topology to the model generated through the QGIS2ThreeJS complement, it was noticed that this is not the most viable option for the interoperability with other software programs. This is due to the fact that the model can only be visualized in the HTML format and the softwares used for the CityGML visualization and exportation does not import models in this format. Therefore, this renders the modeling infeasible in this environment.

## 7.2 Modeling performed in Rhinoceros

The modeling was made in Rhinoceros with the Heron plugin, using the “Heron example”, as it was explained on item 3, obtaining the following result:

The geocodification was performed by inserting the following address “Federal university of Bahia (UFBA), Bahia, Brazil” from which the “Geocode” component attributed the LAT/LONG coordinates (as it was shown in the code in Figure 10). Then, a satellite image of the ArcGIS Base Map catalogue was used from the saved study area according to the code in Figure 11.

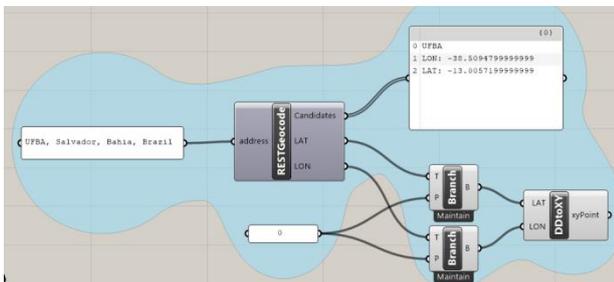


Figure 10: Geocodification of the address of the study field

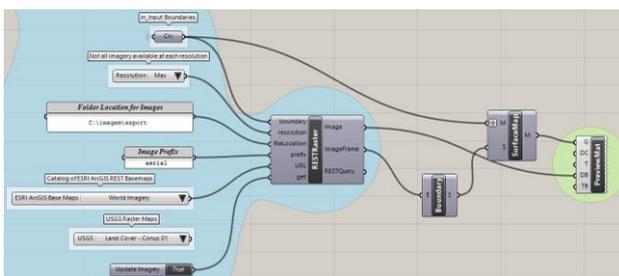


Figure 11: Obtaining the image of the study area

The modeling itself consists of an import of a *shapefile* archive from a polygon that defines the limit of the area that is going to be modeled through the “Import Shapefile” component and its input and output data displayed in step 1 of the code, as represented in Figure 12.

This step also shows that the fields associated with shapefile (semantic attributes) were implemented in a list format and the

values of each field are in the shape of a tree. In step 2, the “List Item” component uses one of its own specific items that in this case correspond to the Field “MDS\_MDT”, which refers to the height of the buildings. In step 3, the coordinates of the points that are transformed into a surface through the “Boundary surface” are used. They create flat surfaces from a collection of curves in a specific limit. After that, this data is converted into mesh through the “Mesh” component. Lastly, in step 4 the specific item of the list (height of the buildings) and the mesh in its format are connected to the “Weaverbird’s mesh thicken” component, which create solid enclosures with the heights of each building and enables the visualization of the modeling through the “Custom preview materials” component.

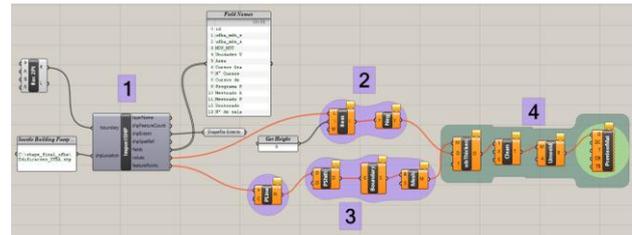


Figure 12: Importing the shapefile archive and geometric modeling methodology

The geometric modeling of the buildings in the campus of UFBA with the Heron plugin can be visualized in Figure 13. It was verified that it is possible to do the modeling with this plugin with a shapefile archive and performing the extrusion of the buildings from the fields that contained the value of their heights. However, the Rhinoceros software does not enable the direct conversion of the model to the CiTYGML format, but it enables the exportation in other formats such as: 3D Studio (3DS), Collada (.dae), OBJ (.obj), PLY (.ply), Sketchup (.skp), VRML (.vrm), X3D (.x3dv) among others, that can easily be imported to other software programs and then converted to CityGML.

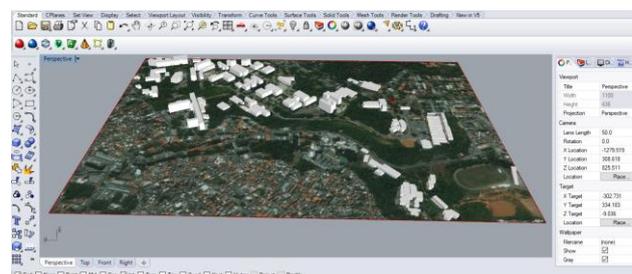


Figure 13: Visualization of the Rhinoceros geometric model done with the Heron plugin

The Meerkat plugin imports the shapefile archive and converts it to mkgis in the process. A satellite image of the area was selected for the modeling and a shapefile archive with the building was added with the “Parse Meerkat file (Parse .mkgis)” component, which provides access to the geometry of the file and to the shapefile attributes.

The “Vector 2PT” component was added, which creates a vector between two points. That is, a point of origin for the coordinates was attributed and then the coordinates of the other point were determined with the “move” component. Posteriorly this vector was interpolated making curves through points, which were extruded with the “extrude” component. The extrusion was done from the “List Item” component recovering the parameter that corresponds to the height of the

buildings. The code can be seen in Figure 14, and the resulting model in Figure 15.

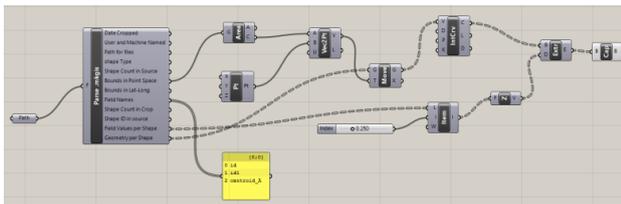


Figure 14: Shapefile import through the “Parse Meerkat file” component

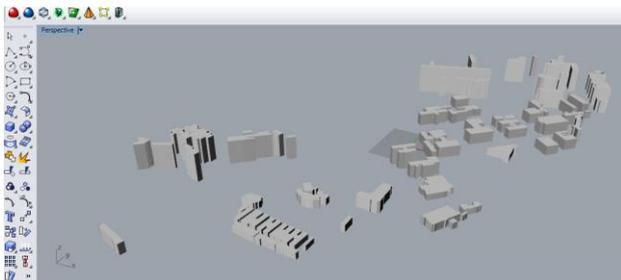


Figure 15: Geometric import in Rhinoceros using the Meerkat plugin

It was observed that both the Heron and Meerkat plugins have the possibility to import vector data in the shapefile format, as well as the same export possibilities for the available file in Rhinoceros. However, using one of the plugins is at the user’s discretion, since the Heron has more functionalities such as researching different sources of satellite images and querying online vector databases. However, these plugins are very similar regarding the processes of extrusion of the footprints of the buildings based on their heights and the resulting models.

### 7.3 Model conversion done in Rhinoceros

As the model produced in QGIS could only be exported in HTML format, it cannot be used in this step. The model done in Rhinoceros was exported in the COLLADA format and the imported in the FME software, enabling its conversion to the CityGML format. The conversion and visualization process in the CityGML model can be seen in Figure 16.

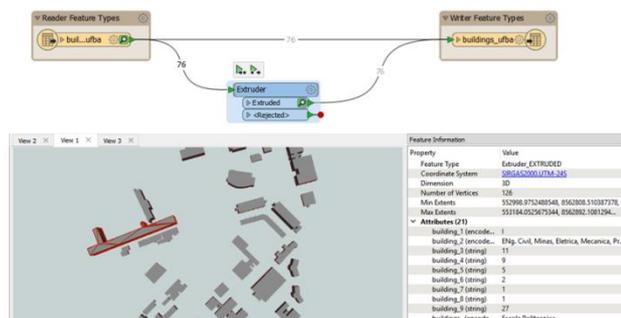


Figure 16: Conversion of the Collada format to CityGML and visualization of the model in FME software

## 8. CONCLUSIONS

This article identified some difficulties in the geometric modeling process with regard to the interoperability among file formats and some tools. This reveals that with regards to data integration to compose a 3D GIS, most of the modeling

software programs import archives in the shapefile format from plugins with specific purposes, but the exportation of the models in the CityGML format is still a matter to be resolved. However, many studies have been performed for the development of software programs that enable the integration of CAD-BIM/GIS data and allow the importation/exportation of data in the CityGML format.

It was verified that the QGIS and Rhinoceros (through their plugins) were efficient for the production of geometric models that are compatible with the LOD1 of the CityGML standard. However neither of them enables the direct exportation of the model in the CityGML format, being so necessary to resort to another software program to do so.

In conclusion, this article sought to discuss the difficulties involved in geometric modeling for the 3D\_GIS applications. This experiment sought to integrate data in the shapefile format (building footprint), posteriorly extruding them according to their respective heights, creating simplified models (LOD1) to which attributes associated to the buildings were associated.

In this sense, it is understood that this paper represents one more step in relation to the implementation of a 3D\_GIS application in UFBA. However, for the geometric modeling of buildings that is compatible with the LOD2 e LOD3, it will be necessary to have more detailed data from short-range photogrammetry and the use of software programs that enable the exportation of the models in the CityGML format. These are the goal still to be achieved in the continuity of this project.

## REFERENCES

3DISGmbH, 2016. CityEditor User Manual, Bocholt, Germany.

Adánek, P., 2015. Import 3D informace do GIS při zachování atributové složky na příkladu stavebních objektů. Plzeň, Tchéquia.

Akmalia, R., Setan, H., Majid, Z., and Suwardhi, D., 2014. Representing 3D model of building from TLS data scanning in CityGML. *Jurnal Teknologi (Sciences & Engineering)*. Malaysia: Penerbit UTM Press Universiti Teknologi Malaysia, v. 71, n. 4, pp. 47–51.

Amorim, A. L., 2016. Estabelecendo requisitos para a Modelagem da Informação da Cidade (CIM): espaços e fronteiras da Modelagem da Informação da Cidade (CIM). In: *Encontro da Associação Nacional de Pesquisa e Pós-Graduação em Arquitetura e Urbanismo*, 4., 2016, Porto Alegre. Anais... Porto Alegre: Programa de Pesquisa e Pós-Graduação em Arquitetura e Urbanismo da Universidade Federal do Rio Grande do Sul.

Bansal, V. K., 2014. Use of geographic information systems in spatial planning: a case study of an institute campus. *Journal of Computing in Civil Engineering*. Fairfax: American Society of Civil Engineers, v. 28, n. 4.

Biljecki, F., 2017. Level of detail in 3D city models. 2017. 353 f. Tese (Doctor of Philosophy). Technische Universiteit Delft, Delft.

Bini, C., Galafassi, M., 2017. Estudo e comparação de softwares relacionados ao projeto climático para utilização no curso de arquitetura e urbanismo. XIV Encontro Nacional de

- Conforto no Ambiente Construído and X Encontro Latino-Americano de Conforto no Ambiente Construído, Associação Nacional de Tecnologia do Ambiente Construído, Balneário Camboriú, pp. 976-985.
- Bobkowska, K., Szulwic, J., Tysiac, P., and Ziółkowski, P., 2017. GIS three dimensional modelling with geo-informatics techniques. In: International Conference "Environmental Engineering", 10., 2017, Vilnius. Anais... Vilnius: Vilnius Gediminas Technical University, 2017.
- Centeno, J. A. S., Mitishita, E. A., 2007. Laser scanner aerotransportado no estudo de áreas urbanas: experiência na UFPR. In: Simpósio Brasileiro de Sensoriamento Remoto, 8., 2007, Florianópolis, Anais... São José dos Campos: Instituto Nacional de Pesquisas Espaciais, pp. 3645-3652.
- Chen, K. W., Janssen, P., Norford, L., 2017. Automatic Generation of Semantic 3D City Models from Conceptual Massing Models. 17th Conference of Computer Aided Architectural Design Futures, Computer-Aided Architectural Design Foundation, Istanbul, pp. 84-100.
- Elias, E. N. N., 2017. Atualização cartográfica através de imagens obtidas com Veículo Aéreo não Tripulado (VANT) quadrotor. Área de Estudo: Campi da Universidade Federal da Bahia. Trabalho de Conclusão de Curso (Graduação em Engenharia de Agrimensura e Cartográfica) – Escola Politécnica, Universidade Federal da Bahia. Salvador.
- Gröger, G., Plümer, L., 2012. CityGML – Interoperable semantic 3D city models. *Journal of Photogrammetry and Remote Sensing*. Amsterdam; New York: Elsevier, v. 71, pp. 12–33.
- Gröger, G., Kolbe, T. H., Nagel, C., and Häfele, Karl-Heinz., 2012. OGC City Geography Markup Language (CityGML) encoding standard. Wayland: Open Geospatial Consortium, 344 p.
- Kutzner, T., Kolbe, T. H., 2018. CityGML 3.0: sneak preview. In: Wissenschaftlich-Technische Jahrestagung der DGPF, 38., 2018, Munique. Proceedings... Munique: Deutschen Gesellschaft für Photogrammetrie, Fernerkundung und Geoinformation (DGPF).
- Motta, S. F., Moura, A. C. M., Ribeiro, S. R., 2017. Going Beyond Data-Driven and Knowledge-Driven to Propose Visual-Driven Evaluation in Multicriteria Analysis: Case Study of Modeling in Grasshopper+Rhino 3D. *Revista Brasileira de Cartografia*, n. 69, v. 8, pp. 1521-1535.
- Lima, M. Q. C., Freitas, F. S., 2016. Modelagem paramétrica e os limites dos mecanismos tradicionais de regulação da forma urbana. *Revista Políticas Públicas & Cidades*, v. 4, n. 1, pp.117 – 138.
- Redweik, P., Teves-Costa, P., Vilas-Boas, I., and Santos, T., 2017. 3D city models as a visual support tool for the analysis of buildings seismic vulnerability: the case of Lisbon. *International Journal of Disaster Risk Science*. Beijing: Beijing Normal University, v. 8, n. 3.
- Santana, S., Moura, A. C. M., Zyngier, C., 2013. The Trends of Geotechnology to Support Urban Planning: New Paradigms and Challenges. *Journal of Earth Science and Engineering*, v. 3, pp. 484-496.
- Santos, A. P., Leite, D. P., Pereira, I. A. A., and Hirookinaga, C., 2016. Digital City: the construction of public shapefiles as a teaching environment, research and extension. XX Congreso de la Sociedad Iberoamericana de Gráfica Digital, Faculty of Architecture, Design and Urbanismo of Universidad de Buenos Aires, Buenos Aires, pp. 272-277.
- Sohail, A., 2013. Creating smart 3D city models with Esri CityEngine. <https://www.youtube.com/watch?v=zScyrwQLf0k&t=2158s> (10 June 2018).
- Washburn, B., 2014. Heron for Grasshopper. <https://www.youtube.com/watch?v=Wy3xKMjfJKw&t=287s> (10 June 2018).
- Wendel, J., Simons, A., Nichersu, A., and Murshed, S. M., 2017. Rapid development of semantic 3D city models for urban energy analysis based on free and open data sources and software. 3th International Workshop on Smart Cities and Urban Analytics, Association for Computing Machinery, Redondo Beach.
- Zlatanova, S., Rahman, A. A., Pilouk, M., 2002. Trends in 3D GIS Development. *Journal of Geospatial Engineering*. Hong Kong: Hong Kong Institution of Engineering Surveyors, v. 4, n. 2, pp. 71-80.
- Zlatanova, S., Stoter, J., Isikdag, U., 2012. Standards for Exchange and Storage of 3D Information: Challenges and Opportunities for Emergency Response. In *Proceedings of the the International Conference on Cartography GIS* (Albena, Bulgaria, June 2012), Bandrova T., Konecny M., Zhelezov G., (Eds.), pp. 17-28.