Eurographics Workshop on Urban Data Modelling and Visualization (2014)
V. Tourre and G. Besuievsky (Editors)

A Configurable LoD for Procedural Urban Models intended for Daylight Simulation

Gonzalo Besuievsky[1], Santiago Barroso[1], Benoit Beckers[2] and Gustavo Patow[1]

[1] Geometry and Graphics Group, University of Girona, Spain
[2] Urban System Engineering Department, Compiegne University of Technology, France

Abstract

In many applications, such as in massive urban models visualization or in the study of the impact of urban simulation at different scales, models with different levels of detail are required. In this paper we propose a flexible system for configuring level of details models using Procedural Modeling aiming to generate only the geometry required for each specific need. We test our system for a solar simulation analysis at urban scale. We evaluate the solar irradiation and the Sky View Factor in order to study the impact at different scales. We show that our tool provides a way to handle the complexity of urban scale models, and specifically to study the sensitivity of the geometry.

1. Introduction

Urban models are complex systems that can be managed at different correlated scales (building/district/city). For instance, in building energy performance, the solar potential and the access to sunlight in urban areas are directly influenced by the urban geometry, both at the neighborhood and at the building scale. For their simulation, a well-defined geometrical model of the urban environment is mandatory. Producing such kind of models at all scales accurately, as may be required by many applications, is a challenge. One of the main difficulties is the amount of data to deal with. City models have large-data sets built from different sources as cadastral data, digital images and CAD models. The management of this large data, concerning new techniques for the treatment of a different scale-model is a current research topic in modeling simulation [Rob11, Bec12] and visualization [HBT*12]. Another important aspect that should be guaranteed is that all geometry representations must be consistent between levels, meaning they must share the same basic geometric structures.

The production of a multi-scale 3D city model implies different representations of the models at different Levels of Detail (LoDs). The classic solution for introducing levels of detail, for an already generated model, is to use reduction techniques to simplify the model complexity from quality parameters [LWC*02]. However, if the simplification is done after the model is completely created it can result in unnecessary synthesis plus reduction processing. Another possibility is to embedded the level of detail representation inside the model generation process, as can be done with procedural urban modeling techniques [MWH*06a]. Using these methodology, LoD generation can be set manually as parameters inside the rule creation. This procedure provides a way to control the density of the model, which can increase excessively in the amount of geometry generated. However, there is a lack of an automatic control of the model complexity according to specific needs. Ideally, the LoD generation for urban multi-scale models should be configurable according to the final requirements of the application.

In this paper we propose a new system for configuring procedural urban models with different levels of detail in a flexible way. The main goal is to provide users with a tool capable of generating automatically different levels of detail according to the application needs. For that, we used a semantic-based procedural modeling. The basis of our approach is the introduction of a new command that can act on some selected elements of the model through configurable criteria. This command automatically detects which is the geometry affected and works on it performing the replacements for the corresponding levels of detail. We design two kind of criteria: a spatial one where the user can set the distance from a point-of-interest region and a semantic criteria where the user can instantiate specific structures by expressing them as semantic combinations (e.g. *Main doors facing an avenue*). We analyze our system for solar simulation computation at urban scale. As simulation results are sensible to geometry accuracy, the tool provides the flexibility to configure the complexity of details balancing the amount of geometry required and the accuracy of the simulation.

2. Related Work

LoD in Urban models

Previous work on level of detail for urban models can be found in the area of urban generalization, like the cartographic generalization proposed by Anders [And05], or the face collapse from known constructive structures as walls and roofs [RCT*]. Other works, like the CityGML standard [Kol09], proposes the definition and usage of five different LoD levels, but does not provide a mechanism to generate them, nor an adaptive LoD scheme.

For procedural modeling, Parish and Müller [PM01] presented an initial proposal intended for city generation based on the L-system recursive nature. Automatic LoD-generation is obtained by starting from the building envelope as axiom, and the output of each rule iteration represents a refining step in the building generation. Although it is simple and automatic, this approach does not provide control on geometric building details. Through a similar approach, in the CityEngine system [Esr12], LoDs can be added manually in the grammar-rules by using a switch-case scheme for controlling the insertion of the geometry. Recently, new approaches were proposed to integrate LoDs mechanism in the procedural processing. In [BP13b], a rewriting method of the rulesets for the buildings has been developed for further replacing the geometric operators, which produced the right level of detail for each asset according to some user-defined criteria. In [BP13a], the authors proposed a highest level of detail by enabling selection, from entire buildings up to whole blocks, for geometric reduction. These works focus more on solving rendering problems, whereas in our approach we target more on the model preparation for simulation analysis.

Urban Physic Simulation

City models are complex systems of physical objects that can be considered as an interface between building and territory, where the main physical parameters are deeply and complexly modified. Such information is needed at all scales: at lower scale by intervention (pedestrian comfort, building thermal efficiency), as boundary conditions, and at upper scale (meteorology, climate). This requirement suggests a unified simulation approach able to represent the physical behavior of the system at each scale with a level of detail required for a given accuracy of the simulation.

Current numerical computational methods, as for example the Finite Element Method, allows to model phenomena at individual scales such as a building or a block street, with different degrees of accuracy. But fully integrated multi-scale approaches are still an open research subject [Bec12]. Beyond the dimensionality that overtakes computer capacities in memory and processing time, adaptive level of detail with respect to the analysis needs comes as an imperative requirement to deal with the problem.

Concerning solar energy simulation, defining the optimal LoD at the neighborhood scale is not a simple problem and most of the approaches are taken from an empirical perspective. In [RBPB], a study of the sensitivity of the geometry used is carried out taking into account the solar flux computation, where for a neighborhood-scale model, different levels of detail elements (windows and roofs) are evaluated.

3. Semantic Procedural Modeling

The seminal works by Wonka et al. [WWSR03] and Müller et al. [MWH*06b] introduced Grammar-based procedural modeling for buildings. A complete survey including more recent improvements can be found in [WAMV11]. The main concept of this technique is a shape grammar based on a ruleset: starting from an initial axiom primitive (e.g. a building outline), rules are iteratively applied, replacing labelled shapes with other labeled ones. New labels can be optionally assigned to the resulting geometry with the purpose of being further processed. In our system, this geometry carries all labels that the shape or any ancestor has received during the production process. We call this systems *Semantic Procedural Modeling*, because the meanings, managed as tags, are associated to all product results of each rule. For instance, a *facade avenue* semantic tag that is derived from a building rule that faces an avenue, can further be used with a command *Subdiv* to produce the semantic tags *top, middle, lower* and *floors*. Then, we can build semantic combination and identify all products that fulfill such a combination. As is was shown in [Pat12, HWM*10], this is equivalent to a graph, where commands are the nodes of the graphs and the edges represent the flux of geometry between rules.

The main potential of shape grammars lies in the variations they can produce, as each created instance of a building could look different by changing parameters of the rules. We explore this idea in combination of the *Semantic Procedural Modeling* described above to build a system that not only identifies structures in the hierarchy but is also flexible to change assets at the leafs to generate models with different levels of detail. This idea was already applied for level of detail production in [BP13b], but here we introduce a more flexible LoD system where the user can specify explicitly the geometry model for each geometry level. Other approaches also uses semantic approaches for other purposes such as local selection [LWW08] or generalization of L-systems [KPK10].

4. LoD System

Our system workflow is described in Figure 1. First, a procedural building model is generated using semantically enriched rules as described in Section 3. Then, the user should configure the levels, deciding the architecture elements that should be replaced and the criteria that should be applied for the replacements. We provide three different methods for the multi-scale generation: an explicit level-of-detail specification, a view-dependent LoD and a semantic combination method. The rest of this section describes details of each procedure.

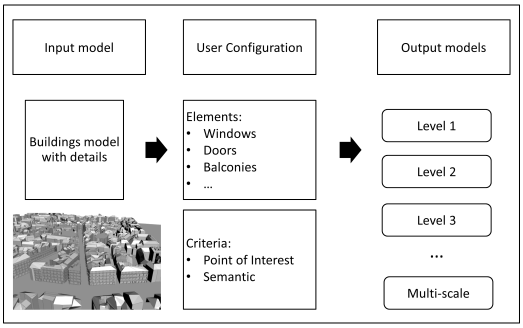


Figure 1: *System workflow*

4.1. Explicit LoD

This is a simplest but very useful procedure. We call it *Explicit LoD* because all levels of simplification of a given architectural element must be completely specified by the user. The user provides the final assets details, such as windows, doors or balconies, for all the desired level of detail (for example, windows at the winLoD0, winLoD1 and winLoD2 levels). We provide a GUI for configuring the different levels in a flexible way (see Fig. 2). This basic method allows to study the impact on a simulation for the same building with different structures at different levels of detail.

4.2. POI Distance

We implement a Point of Interest (POI) reference approach to classify the geometry according to a distance criteria. The user can define a set of distances to the reference point, and for each one, which LoD should be applied. Then, the system automatically builds the model according to the respective building evaluation (see Fig. 3). This is an interesting procedure for cases where the evaluation analysis must be concentrated on a local region and the rest of the city can be roughly approximated. It also has applications when dealing with a dynamic reference point, where an impact study should be carried along a path in the model.

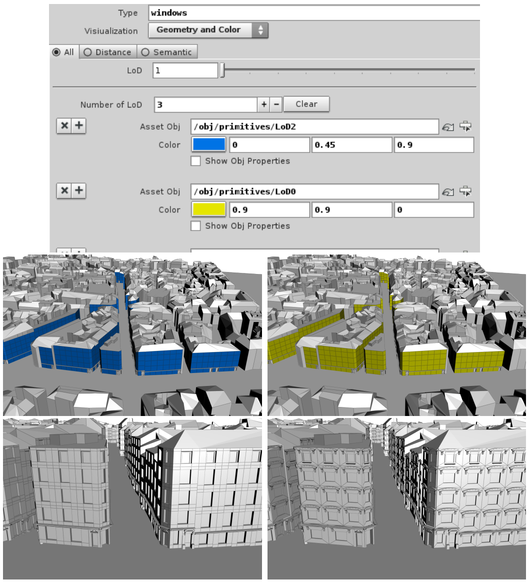

Type windows
Visiualization Geometry and Color
● All ○ Distance ○ Semantic
LoD 1
Number of LoD 3
Asset Obj /obj/primitives/LoD2
Color 0 0.45 0.9
☐ Show Obj Properties
Asset Obj /obj/primitives/LoD0
Color 0.9 0.9 0
☐ Show Obj Properties

Figure 2: *Explicit LoD: interface configuration for the element window (top), geometry affected (middle) and final replacements (bottom).*

4.3. Semantic LoD

The semantic LoD method allows to customize any defined building structure. The definitions should be associated in the initial model with textual labels. They can be basic architectural elements as *windows* or *doors*, but also building meanings like the *top floor* or the *ground floor*. The user can select several structures using boolean combination to define them. For instance, to focus on the *top floor* that faces to the *main avenue* buildings. For each case, the geometry that is going to be replaced should be instantiated in the same way as the two methods previously described. The system automatically searches for geometry that accomplishes a valid combination and builds the final model (see Fig 4). This procedure is interesting when the analysis is focused on a particular structure of a neighborhood. The method could also be used to store a model with a multi-scale description, as for example using the standard format CityGML [Kol09].

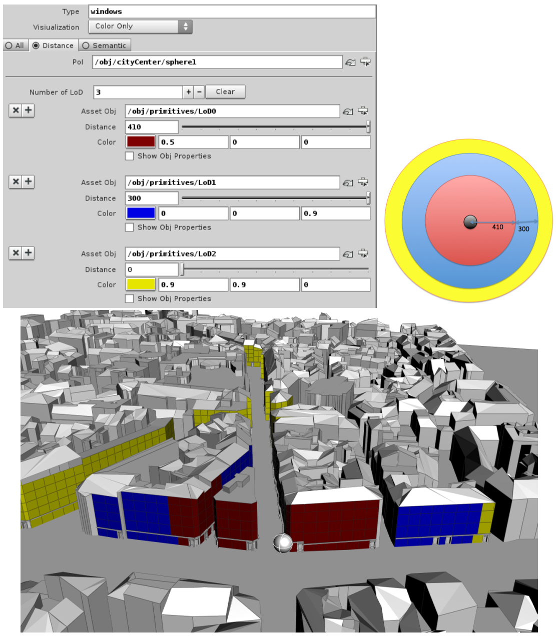
4.4. Implementation

The base of our system essentially is an automatic transformation of the initial building model. For this purpose we create a new command called *LoD* that is first instantiated with the list of architectural elements (windows, doors, etc) selected where different levels of detail should be applied. Then, we perform a search on the graph-model of the building and we transform it by inserting the new rule according to the nodes that are affected. For each element selected, a new command *LoD* will be inserted. Once the command is created, all parameters are allowed to be configured in a flexible way from the GUI interface, where each LoD method has their own specific parameters (see Figures 2, 3 and 4).

The other important operation involved in the method is the replacement of the geometry. This is the most time consuming procedure as all the affected buildings should be recomputed for each specific level of detail configuration. The classical *Insert* shape-grammar command [MWH*06a] is embedded in the new *LoD* rule and used to associate the assets at each level. All geometric transformation involved in the insertion of the new assets can also be reconfigured from the *LoD* interface. For fast visualization, we also provide a color mode rendering that indicates the placement where the geometry is inserted at each level in different colors. This is particulary interesting for the case of POI, where the distances for each level should be set before the final computation. For classifying the structures according to the reference point selected, we evaluate the distance from the point to the center of each elements affected by the selection. The final model resulted can be output as independent geometry for each level in standard 3D formats, as required for the application. For example, for the test results of Section 5 we exported each level in separate STL files.

5. Application for Solar Simulation

We test the usability of our system for a daylight analysis in a urban model. The full city model (see Figure 2) at the highest level of detail (LoD2) is composed by 92.606 polygons. The purpose here is not to perform a real daylight analysis but to study the impact of the geometry at different resolutions can have. Also, a secondary objective is to show the simplicity of our system to perform an analysis on geometry sensitivity at different levels of detail.



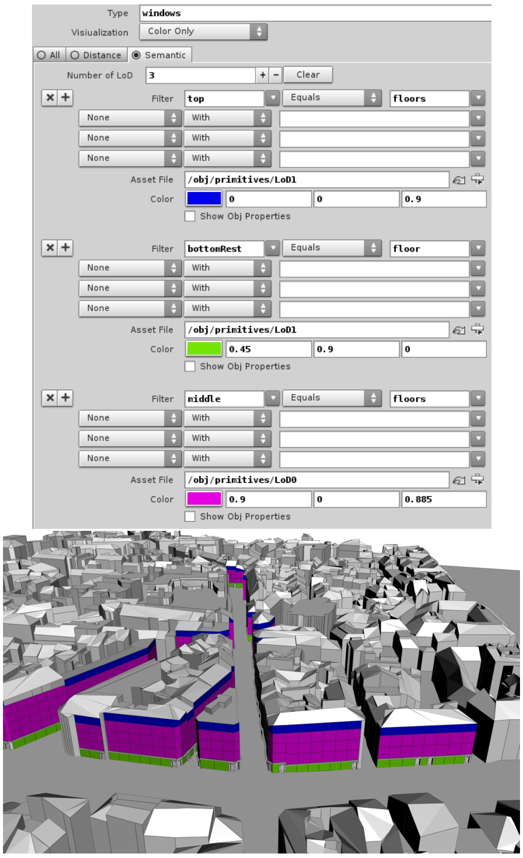
Type windows
Visiualization Color Only
○ All ● Distance ○ Semantic
PoI /obj/cityCenter/sphere1
Number of LoD 3

Asset Obj /obj/primitives/LoD0
Distance 410
Color 0.5 0 0
☐ Show Obj Properties

Asset Obj /obj/primitives/LoD1
Distance 300
Color 0 0 0.9
☐ Show Obj Properties

Asset Obj /obj/primitives/LoD2
Distance 0
Color 0.9 0.9 0
☐ Show Obj Properties

Figure 3: *Point of Interest LoD. The user defines elements and affected distances (top) from a reference POI (the ball) that results is geometry replacements (bottom)*



Type windows
Visiualization Color Only
○ All ○ Distance ● Semantic
Number of LoD 3

Filter top Equals floors
None With
None With
None With
Asset File /obj/primitives/LoD1
Color 0 0 0.9
☐ Show Obj Properties

Filter bottomRest Equals floor
None With
None With
None With
Asset File /obj/primitives/LoD1
Color 0.45 0.9 0
☐ Show Obj Properties

Filter middle Equals floors
None With
None With
None With
Asset File /obj/primitives/LoD0
Color 0.9 0 0.885
☐ Show Obj Properties

Figure 4: *Semantic LoD. Windows at top, lower and middle floors are selected through semantic combinations*

We used the windows of the buildings as target elements to compute the direct solar irradiation per unit area of the glazing surface and the sky view factor [Ung09]. Both of them are intensive magnitudes. The Direct solar irradiation is evaluated taking into account the absorption of the atmosphere for a clear day. Sky View Factor (SVF), currently used in daylight assessment, is defined as the percentage of sky visible from a surface, taking into account the angle of inclination to the sky vault. It is a pure geometrical parameter that has a physical meaning. All daylight computation were performed using the engine Heliodon [BM12].

Figure 5 shows results of the simulation for direct solar irradiation and sky view factor analysis for three different levels of detail for the windows: a single plane (LoD0), a simple window-frame model (LoD1) and the full model with details (LoD2). All windows having the same exposed glazing area. The whole city model is simplified to 23.909 polygons for LoD0 and to 33.527 polygons for LoD1. For the solar impact we computed the total-day irradiation for the summer solstice (21th June) at Barcelona. Considering the whole facade computed for the selected building, the relative error for the two simplifications, given as $Er_{LoD_i} = (I_{LoD_i} - I_{LoD_2})/I_{LoD_2}$, are 200% and 87% for LoD0 and LoD1, respectively, which are significant difference. The total SVF, that is the percentage of visible sky considering the all windows facade, is 30.6% for LoD0, 15.1% for LoD1 and 14.8% for LoD2. We observe a small variation between the LoD1 and LoD2 comparing to the appreciated difference with LoD0. Thus, LoD1 may be a feasible approximation when computing daylight diffuse components. This result confirms the one obtained in [RBPB], where it is shown that the thickness of the wall plays an important role in the SVF and in most cases could be simplified to a window frame model with no more details. One of the benefits of our system is that it allows to easily analyze and configure the correct level of detail according to the required specific simulation.

To analyze the neighbor geometry impact we consider the same previous computation but just with the generated facade with detailed geometry (LoD2), taking out all the other buildings. For the same SVF computed as the one computed in Figure 5 we obtained 29.1% which represents a large difference with respect to the 14.8% originally obtained. We note a considerable difference in the simulations (see Figure 6), that enhances the importance of the neighbor scale influence, even when only a local building is being analyzed. This motivates the use of a distance criteria for simplification from a POI reference.

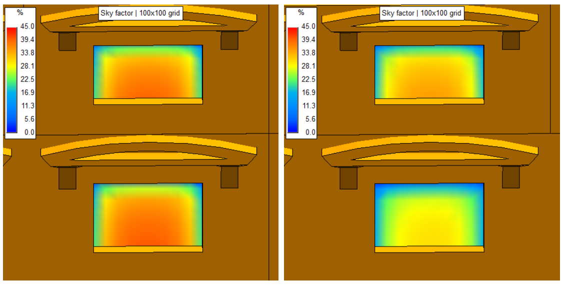


Figure 6: *SVF at the top windows of the facade considering only the isolated building (left) and with the whole city (right).*

The Point of Interest LoD can be used for a local analysis study introducing a reference point (see Figure 7). While no significant simulation results are observed comparing to the simulation with the full model resolution model, there is an important reduction in geometry size. Considering only the windows geometry that is being replaced, we obtained a simplification of 12% over the full detailed windows. In simulation applications, involving numerical methods like the finite element approach for global illumination or heat transfer, it is important to to manage the balance between the amount of data and the accuracy.

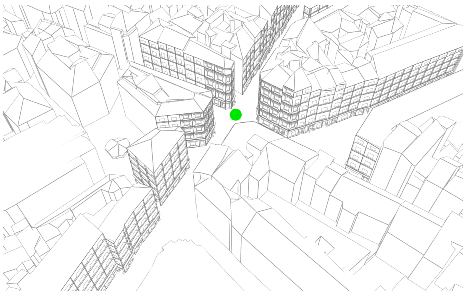


Figure 7: *Urban model resulted from the POI LoD using the green ball as reference.*

5.1. Discussion

Our system does no use simplification reduction routines as the ones presented in [BP13b] and [HBT*12]. We consider that all element models that are eligible for being simplified should be provided by the modeler. According to this strategy, our technique can be classified as a discrete LoD. The main advantage of the present approach is that we provide more control over the reduction, making it more feasible for analysis.

Our methodology is independent of the application and could be included in any ruleset procedural engine. Aside of the flexibility of the configurable parameters shown in the examples for the simplification, a procedural strategy also provides the benefit of easily modifying modeling parameters in order to explore with changing element sizes or shapes. This advantage could also improve simulation application where different elements should be explored in order to analyze the impact of a given configuration.

Our future work includes the application of the method to structures at all levels like facades, blocks and districts. Also, more detailed simulation analysis specific for global illumination and heat transfer will be performed in order to analyze the geometry needs and sensitivity.

Acknowledgements

This work was funded by TIN2010-20590-C02-02 project (Min. de Ciencia e Innovación, Spain).

References

[And05] ANDERS K.-H.: Level of detail generation of 3d building groups by aggregation and typification. In *Proceedings of the XXII International Cartographic Conference* (2005). 2

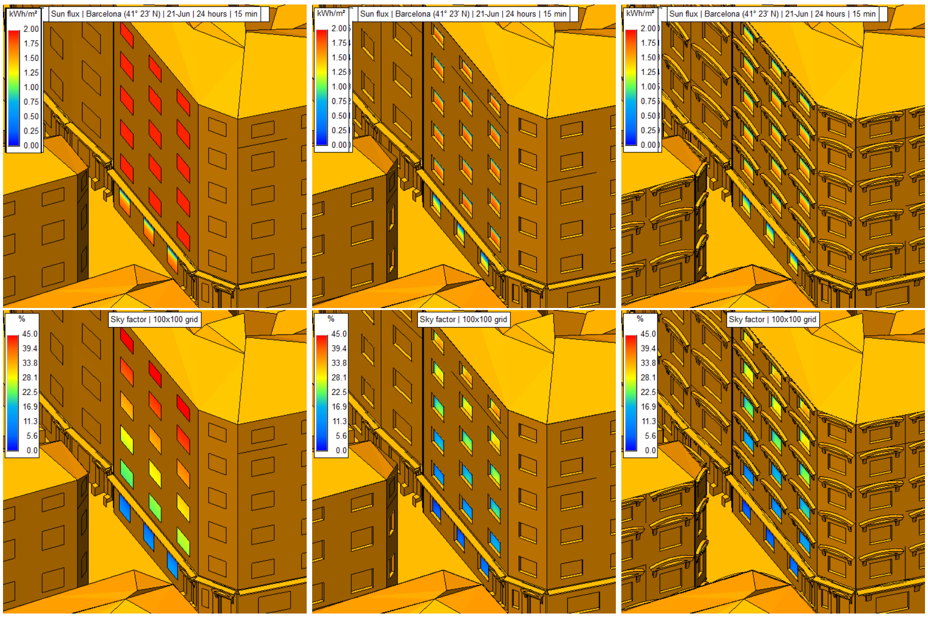


Figure 5: *Daylight simulation at windows for the selected facade: direct irradiation per area (top) and SVF (bottom) for LoD0 (left), LoD1 (middle) and LoD2 (right).*

[Bec12] BECKERS B.: *Solar Energy at Urban Scale*. ISTE. Wiley, 2012. 1, 2

[BM12] BECKERS B., MASSET L.: Heliodon, 2012. www.heliodon.net. 4

[BP13a] BESUIEVSKY G., PATOW G.: City-level level-of-detail. In *XXIII Spanish Computer Graphics Conference* (2013), pp. 29–36. 2

[BP13b] BESUIEVSKY G., PATOW G.: Customizable lod for procedural architecture. *Computer Graphics Forum 32*, 8 (2013). 2, 5

[Esr12] ESRI: Cityengine, 2012. www.esril.com. 2

[HBT*12] HE S., BESUIEVSKY G., TOURRE V., PATOW G., MOREAU G.: All range and heterogeneous multi-scale 3d city models. In *Usage, usability, and utility of 3D city models*. Edp sciences, 2012, p. 16p. 1, 5

[HWM*10] HAEGLER S., WONKA P., MÜLLER S., VAN GOOL L., MÜLLER P.: Grammar-based encoding of facades. *Computer Graphics Forum 29* (2010), 1479–1487. 2

[Kol09] KOLBE T. H.: Representing and exchanging 3d city models with citygml. In *Lecture Notes in Geoinformation and Cartography* (2009), Springer Verlag, p. 20. 2, 3

[KPK10] KRECKLAU L., PAVIC D., KOBBELT L.: Generalized use of non-terminal symbols for procedural modeling. *Computer Graphics Forum 29*, 8 (2010), 2291–2303. 2

[LWC*02] LUEBKE D., WATSON B., COHEN J. D., REDDY M., VARSHNEY A.: *Level of Detail for 3D Graphics*. Elsevier Science Inc., New York, NY, USA, 2002. 1

[LWW08] LIPP M., WONKA P., WIMMER M.: Interactive visual editing of grammars for procedural architecture. *ACM Trans. Graph. 27* (August 2008), 102:1–102:10. 2

[MWH*06a] MÜLLER P., WONKA P., HAEGLER S., ULMER A., VAN GOOL L.: Procedural modeling of buildings. *ACM Trans. Graph. 25* (July 2006), 614–623. 1, 4

[MWH*06b] MÜLLER P., WONKA P., HAEGLER S., ULMER A., VAN GOOL L.: Procedural modeling of buildings. *ACM Trans. Graph. 25*, 3 (2006), 614–623. 2

[Pat12] PATOW G.: User-friendly graph editing for procedural modeling of buildings. *IEEE Computer Graphics and Applications 32* (2012), 66–75. 2

[PM01] PARISH Y. I. H., MÜLLER P.: Procedural modeling of cities. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques* (2001), Press, pp. 301–308. 2

[RBPB] RODRIGUEZ D., BESUIEVSKY G., PATOW G., BECKERS B.: Procedural models to better compute solar flux at the neighbourhood scale. In *Proceedings of Flow modeling for urban development*. 2, 5

[RCT*] RAU J.-Y., CHEN L.-C., TSAI F., HSIAO K.-H., HSU W.-C.: Lod generation for 3d polyhedral building model. In *Advances in Image and Video Technology*, Lecture Notes in Computer Science. 2

[Rob11] ROBINSON D.: *Computer Modelling for Sustainable Urban Design: Physical Principles, Methods and Applications*. Earthscan Publications, Limited, 2011. 1

[Ung09] UNGER J.: Connection between urban heat island and sky view factor approximated by a software tool on a 3d urban database. *Int. J. Env. and Pol. 36*, 1-2-3 (2009), 59–80. 4

[WAMV11] WONKA P., ALIAGA D., MÜLLER P., VANEGAS C.: Modeling 3d urban spaces using procedural and simulation-based techniques. In *ACM SIGGRAPH 2011 Courses* (2011), SIGGRAPH '11, pp. 9:1–9:261. 2

[WWSR03] WONKA P., WIMMER M., SILLION F., RIBARSKY W.: Instant architecture. *ACM Transaction on Graphics 22*, 3 (July 2003), 669–677. Proceedings ACM SIGGRAPH 2003. 2