GRAMMAR-BASED AUTOMATIC 3D MODEL RECONSTRUCTION FROM TERRESTRIAL LASER SCANNING DATA

Qian Yu,[*] Petra Helmholz, David Belton and Geoff West

Cooperated Research Centre for Spatial Sciences (CRCSI)
Department of Spatial Sciences, Curtin University
GPO Box U 1987, Perth, WA, 6845, Australia
qian.yu9@postgrad.curtin.edu.au, (petra.helmholz, d.belton, g.west)@curtin.edu.au

Commission IV/1

KEY WORDS: model, reconstruction, grammar, building, 3D

ABSTRACT:

The automatic reconstruction of 3D buildings has been an important research topic during the last years. In this paper, a novel method is proposed to automatically reconstruct the 3D building models from segmented data based on pre-defined formal grammar and rules. Such segmented data can be extracted e.g. from terrestrial or mobile laser scanning devices. Two steps are considered in detail. The first step is to transform the segmented data into 3D shapes, for instance using the DXF (Drawing Exchange Format) format which is a CAD data file format used for data interchange between AutoCAD and other program. Second, we develop a formal grammar to describe the building model structure and integrate the pre-defined grammars into the reconstruction process. Depending on the different segmented data, the selected grammar and rules are applied to drive the reconstruction process in an automatic manner. Compared with other existing approaches, our proposed method allows the model reconstruction directly from 3D shapes and takes the whole building into account.

1 INTRODUCTION

In recent years, 3D models have been used in a variety of applications, and the steadily growing capacity in both quality and quantity are increasing demand. In order to cover the requirements and to keep the existing models up to date, automatic reconstruction methods are needed to avoid the labour intensive and time comsuming manual processing workflow. While in recent years a number of paper were published focusing on 2D data sets (Helmholz et al., 2013), there are two major types of model reconstruction approaches in principle (Musialski et al., 2012), which are the data-driven approach and the model-driven approach. A data-driven approach extracts features and elements (e.g. segments) from the input data to create a model. Typically hard-wired rules are used to guide the model building process that are tailored to the task at hand and need to change for different environments. The model-driven approaches use an existing model to hypothesise or predict candidate elements and features and their organisation and then looks for this information in the data in order to verify the presence of the model.

Compared with traditional digital imagery as the only data source, nowadays terrestrial laser scanning provides explicit 3D information, which enables the possibility to rapid and accurately capture the geometry of complex buildings. Instead of starting from 2D data sets to derive 3D building models, a series of rules and grammars are defined to directly describe the 3D geometry structures of buildings and drive the automated reconstruction process. Our proposed grammar-based approach is based on data-driven methods, which constructs a building model as a high level representation from low level data resources i.e. point cloud data. In this way, the same building information can be represented by different levels of data (point cloud data, segmented data, grammar description and model). Therefore the model verification or update can use (all) suitable data resources, especially the grammar description of buildings. In the rest of this paper, we focus on only 3D reconstruction.

Grammar-based methods have been extensively used in architecture modelling. The most well-known examples are the L-systems (Lindenmayer-systems), which were initially developed for modelling plants by (Prusinkiewicz et al., 1990). An L-system consists of an alphabet of symbols that can be used to make strings. The construction is from an initial axiom string from a given symbol set. For each iteration, a collection of production rules expand each symbol into a larger string of symbols. Eventually the generated strings are translated into geometric structures. L-systems have also been used for extracting the buildings and streets from aerial imagery by (Parish and Müller, 2001), in which there is some indication of growth as a city expands by the addition of buildings and roads. However, since L-systems simulate growth in open spaces, and in contrast to plants, buildings do not grow in free space, they are not appropriate for the modelling of individual buildings, especially from street level data. Thus, other types of grammars have been proposed for the modelling. The idea of shape grammars was introduced in 1972 (Stiny and Gips, 1972), which define rules for the specification and transformation of 2D and 3D shapes. A shape grammar includes shape rules and a generation engine that selects and processes rules. Shape rules define how an existing shape can be transformed. Shape grammars have been successfully used in architecture (McKay et al., 2012), however, the applicability for automatic generation of buildings was limited, since the derivation is intrinsically complex and usually done manually, or semi-automatically by computer, with a human deciding on the rules to apply. In 2003, Wonka *et al.* (Wonka et al., 2003) employed a split grammar to generate architectural structures based on a large database of split grammar rules and attributes. In this approach, a split grammar is introduced to allow for dividing the building into parts, and also a separate control grammar is proposed to handle the propagation and distribution of attributes. However, due to the requirement of an excessive amount of splits for complex models, the proposed split grammars have limitations to handle the complexity of architectural details. For instance, this method is difficult to generate large city models artificially down to very

[*] Corresponding author

This contribution has been peer-reviewed.
doi:10.5194/isprsarchives-XL-4-335-2014

detailed levels. Following this idea, a new Computer Generated Architecture (CGA) grammar (Müller et al., 2006) is presented to generate detailed building architecture in a predefined style, which is demonstrated by a virtual reconstruction of ancient Pompeii. They solely use context-sensitive shape rules to implement splits along the main axes of the façades. Nevertheless, the approach has a limitation in that the created models and the variety of façade structures are restricted to the knowledge base inherent in the grammar rules. More recently, formal grammars have been applied in building façade modelling (Ripperda, 2008, Becker, 2009, Becker and Haala, 2009) to reconstruct building façades from point cloud data. Depending on the structures of façade, the façade model is defined by a formal grammar. Each grammar rule subdivides a part of the façade into smaller parts according to the layout of the façade. A rule selection mechanism is used to guide the derivation process. However façade modelling does not consider the entire building and is limited to certain type of structures in some cases, i.e. symmetric structures.

In this paper, our proposed method aims to derive a structure description of a whole 3D building by using a pre-defined grammar and rules, which are applied in an automated data-driven reconstruction process. Our main contribution is to apply a formal grammar directly on the 3D data set to develop a systematic method for automated 3D building model reconstruction. Compared with conventional ways to derive 3D models from 2D data sources, reconstruction of 3D models directly from the 3D data source i.e. terrestrial laser scanning, can capture more accurate information. We firstly develop an algorithm to convert the segment data in DXF format into the 3D shape, labelling each surface and the adjacent edges. The segment data is extracted from terrestrial laser scanning in the pre-processing stage and is not discussed in this paper. Then a set of grammar and rules are proposed based on generic 3D shapes of building structures i.e. windows and doors. Within the automated model reconstruction, a grammar engine is thereby developed to guide the derivation process by selecting the best fit grammar and rules according to the specific shape structures.

The rest of the paper is organised as follows. In Section 2, we introduce the background related to our work, which include the practical system model and building grammars. In Section 3, the proposed grammar-based model reconstruction method is discussed in detail. In Section 4, we present the first results from a sample data set. Finally, conclusions are made in Section 5.

2 BACKGROUND

Over the past years, the number of available 3D city models has significantly increased. Many applications of these models are widely used in both industry and government sections, such as urban planning and infrastructure management. One of the main challenges is how to keep these existing models up to date in an efficient and accurate manner. One promising solution proposed by us is to develop an automated model assessment framework, which includes *model reconstruction*, *model matching*, *model verification* and *model update*. This framework allows each subsystem to work for one or more data resources in different levels of representation, such as segmented data and grammar description of buildings, which contain different attributes of building information. In this paper, we only explore the grammar-based model reconstruction subsytem.

2.1 System Overview

Figure 1 illustrates the work flow for the proposed building model reconstruction system, where the block diagram in grey indicates it is not part of our work. As the input of system, 3D point cloud data is generated from terrestrial or mobile laser scanning devices, which measures the surface of scanned buildings in the form of a large number of points. The point cloud data then requires further segmentation to extract planar, cylindrical and other surface features, using for instance the approach of principal component analysis (PCA) (Nurunnabi et al., 2012). These two steps have been a focus of a significant amount of research in the past with many techniques and routines available. In this paper, segmented data from 3D point cloud is used as the input for our work directly.

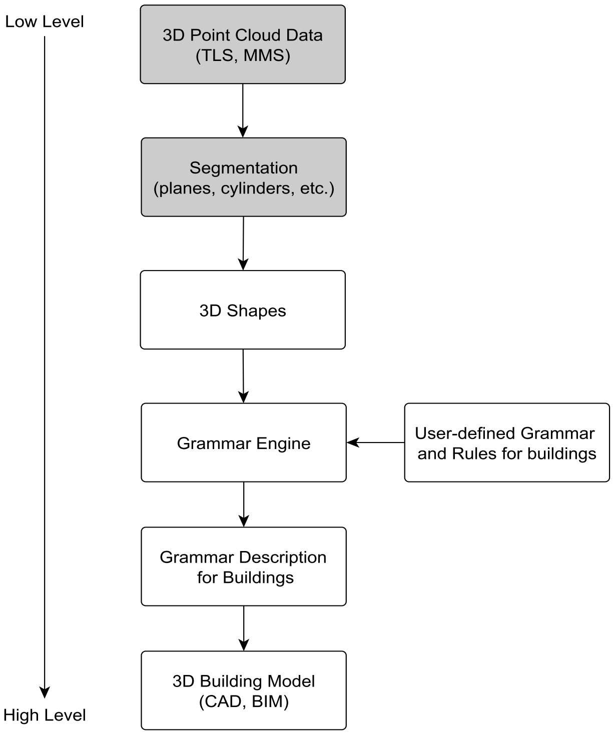


Figure 1: The main flow for proposed building model reconstruction system, where the block diagram in grey indicates it is not part of our work.

As shown in Figure 1, segmented data needs to be converted into 3D shapes in order to support the user-defined grammar and rules, which are derived from 3D building structures. To drive the building reconstruction process, a grammar engine is proposed to apply the appropriate rules for the given shape to break it down into the elementary objects. For example, a complex shape of wall can be divided into a wall panel and many windows, which are very basic building structures. The breaking down process is described via a tree structure, where the root represents the given shape and the leaf indicates the elementary object. A 3D building model thereby is represented by a set of elementary objects according to the derived tree description based on grammars and rules. This grammar-based description of building models can be also used to generate 3D building models in other formats i.e. CAD and BIM.

2.2 Building Grammar

In this part, we firstly explain the concept of grammar in general and then we introduce a grammar more specific to buildings.

Formally a grammar is defined as a four-tuple $G = (T, N, R, I)$ (Wonka et al., 2003). The terminal symbols T and the non-terminal symbols N build the alphabet of the grammar. The non-terminal symbols can be replaced by other non-terminal or terminal children, while terminal symbols cannot be subdivided further. R is a set of production or replacement rules, and I is the initial symbol, a non terminal symbol which defines the initial point for all replacement. A context-free grammar (CFG) is applied for our case, which implies that R contains rules of the form $N \rightarrow (T \cup N)^+$. In other words, a non-terminal symbol on the left side can be replaced by a number of terminal and non-terminal symbols on the right side. The language $L(G)$ of grammar G is defined as all symbols that can be derived from I with rules from R.

A building grammar G_B is derived from the general grammar G and specifies a set of rules R_B particularly applied for building structures. In order to yield a meaningful set of terminals for the grammar G_B, the generic building structure is broken down into some set of elementary parts, which are regarded as indivisible or atomic and therefore serve as building terminals T_B. The elements of building terminals are window elements, door elements, wall elements and roof elements. It is noted that there are two types of wall elements to be distinguished here: wall panel, which represents a basic wall element and an extruded wall, which is a more complex wall element. Extruded walls include structures like windows and doors that are inset within the wall panel, or other attached elementary parts on a wall that stick out from the wall panel. For example, Figure 2 implies the terminal symbols of buildings such as roof element, wall element, window element and door element on the left hand side. An extruded walls including an inset window structure and protruding wall parts is shown on the right hand side. As the building grammar is also a context-free grammar, which means any part of the building's excluded terminal elements T_B can be replaced by a further smaller partition of building and terminal elements. Therefore the language $L(G_B)$ contains all possible building models.

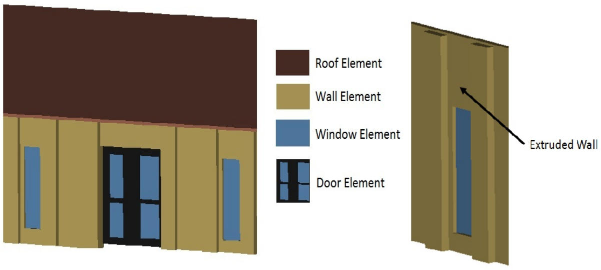


Figure 2: Example for the terminal symbols of buildings - roof element, wall element, window element and door element. Extruded wall that includes inset window structure and stick-out wall parts is shown on the right hand side.

A typical process to derive the 3D building model from 3D shapes is to refine them over a number of successive steps. The defined rules are applied for each non-terminal part of the building to split it into elementary parts, which are associated with the model components. This partitioning process further develops the building model by providing more detailed information. Therefore a building model can be eventually described by the set of terminal structures derived using the defined grammar and rules.

Figure 3 illustrates an example of the derivation steps using a tree structure. It shows a small part of the building, which is subdividable. Based on the split rules, the derivation step is represented as follows:

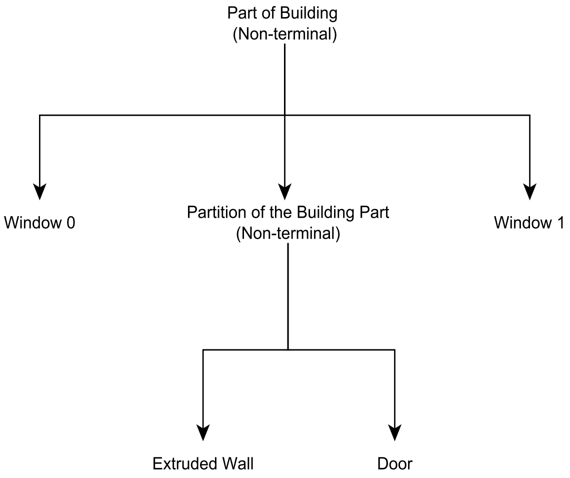


Figure 3: Example of a tree structure from building model derivation according to the split rules.

$$part\ of\ building \rightarrow window_0 + window_1 + partition\ part$$

and

$$partition\ part \rightarrow extruded\ wall + door$$

where $window$, $extruded\ wall$ and $door$ are all terminal symbols.

The set of building rules R_B may vary according to the different specific buildings. However, there are a few common rules in the G_B. If the building has more than one floors, $Building \rightarrow \sum_i Building_{floor_i}$. It divides the multi-floors into single floors and reconstructs each floor respectively. If there are repetition symbols (either terminal or non-terminal), the repeated parts are replaced by $Repeat\ Part$. The $Repeat\ Part$ is modelled once and the additional number of replacements is stored. If the building has symmetric structures, only one side of the symmetric part is modelled and is used to replace the other side automatically. The other important rule that is the terminal symbols are allowed to be replaced by other terminal symbols if the later symbol is deemed to be more appropriate.

3 GRAMMAR-BASED 3D MODEL RECONSTRUCTION

As shown in Figure 1, the proposed grammar-based 3D model reconstruction include four major steps, which are shown as follows:

- Step 1 - Convert segmented data into 3D shapes
- Step 2 - User-defined grammar and rules
- Step 3 - Grammar Engine
- Step 4 - Generate a building model from grammar-based model description

In this section, we introduce these four steps in details.

3.1 Step 1 - Convert segmented data into 3D shapes

We have used building grammars for primitive shape objects, which are consists of planes, adjacent edges and vertices with labels. However the segmented data from 3D point cloud available for us is formatted in DXF for example, which represents the building structures with native geometry components such as poly-lines. As it is difficult to capture the high level structure information from primitive low level data directly, we proposed an algorithm to convert the segmented data stored in DXF format into shape objects.

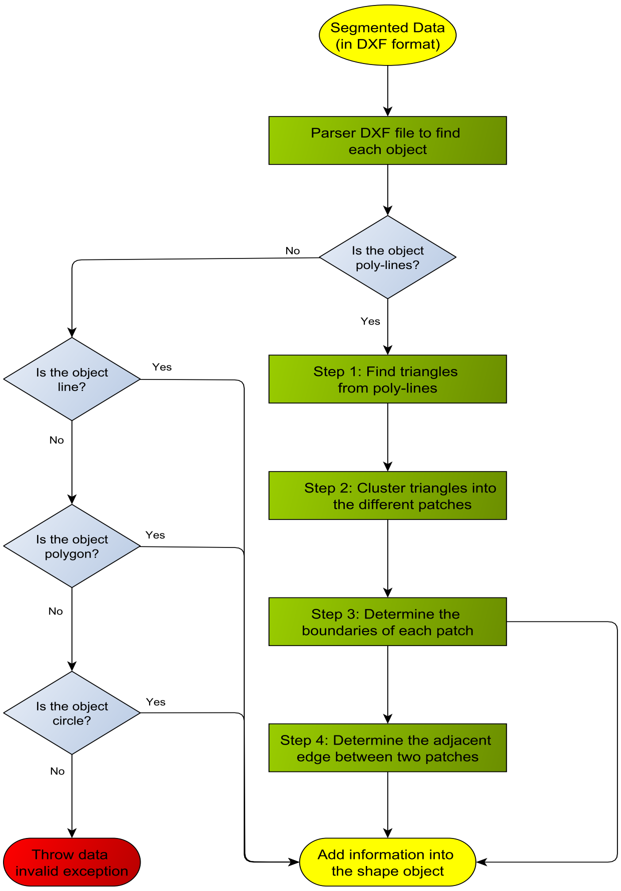

Segmented Data
(in DXF format)
Parser DXF file to find each object
Is the object poly-lines?
No
Yes
Is the object line?
Yes
No
Step 1: Find triangles from poly-lines
Step 2: Cluster triangles into the different patches
Is the object polygon?
Yes
No
Step 3: Determine the boundaries of each patch
Is the object circle?
Yes
No
Step 4: Determine the adjacent edge between two patches
Throw data invalid exception
Add information into the shape object

Figure 4: The main flow of the Segmentation-to-Shape algorithm.

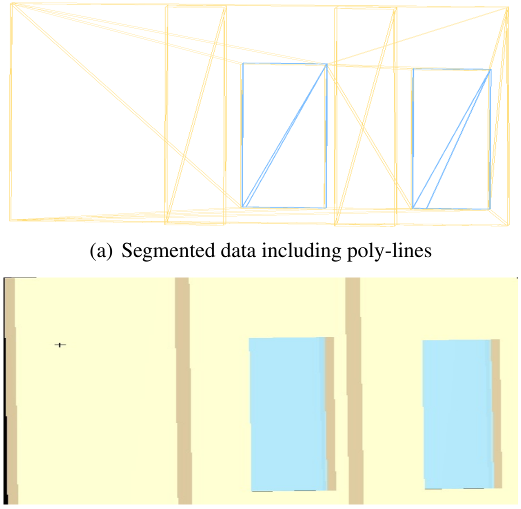
Figure 4 shows the main flow of the Segmentation-to-Shape algorithm. Depending on the DXF specification (DXF Reference, 2012), DXF file can be parsed into individual objects such as line, polygon, circle and poly-lines. For those objects within the same plane, it is labelled as one patch of the shape.

However, poly-lines may not be within the same plane, which require more steps to convert them into a shape object. In *Step 1*, triangles are extracted from the poly-lines in order to find the possible shape surface, as each triangle only belongs one shape surface based on DXF specification (DXF Reference, 2012). In *Step 2*, all triangles thereby need to be clustered into different shape surfaces. The equation of a plane is used to determine whether two triangles are within the same plane. From linear algebra theory (Poole David, 2006), the equation of a plane with non-zero normal vector $\mathbf{n} = (a, b, c)$ through the point $\mathbf{x}_0 = (x_0, y_0, z_0)$ is:

$$\mathbf{n} \cdot (\mathbf{x} - \mathbf{x}_0) = 0 \qquad (1)$$

where $\mathbf{x} = (x, y, z)$. Therefore the normal vectors from two planes are compared to check whether they are in parallel at first. If normal vectors are not in parallel, the two planes must be intersected and the adjacent edge is the line within both planes. If normal vectors are in parallel, then check any known point that lies on the opposite plane to determine whether the two planes are the same. In this way, all shape surfaces can be found out. The last two steps are used to find the boundaries of each shape patch and the adjacent edge between different patches, which have been discuss in *Step 2*.

Figure 5 gives an example. The sub-figure 5(a) shows poly-lines from segmented data, and sub-figure 5(b) illustrated the shape objects with the labelled surfaces in different colours. Blue indicates the patches that are inset into the wall.



(a) Segmented data including poly-lines

(b) Shape objects with labelled surfaces in different colours. Blue indicates the patch that are inset into the wall.

Figure 5: An example to demonstrate the proposed Segmentation-to-Shape algorithm.

3.2 Step 2 - User-defined grammar and rules

Grammar and rules have been discussed in 2.2. Depending on the features of different building, rules can be changed accordingly. In practice, a grammar parser will be implemented to interpret the pre-defined grammar and rules, which are stored in files.

3.3 Step 3 - Grammar Engine

Figure 6 illustrates internal flow of the proposed grammar engine, which is used for driving the building model reconstruction based on 3D shapes and user-defined grammars. The block diagrams in grey indicate internal modules, and connection between internal modules uses black lines. The external modules are indicated in the block diagram in white and uses red lines to connect with internal modules. Blue block diagram shows a database.

In the grammar engine, there are mainly two working paths. The first path is through a grammar manager to select the best fit grammar and rules to perform the building partition depending on the input shape objects and their features. The second path through a tree manager is to store and update the interpretation tree of the reconstructured building model, which is based on the building partition results from the first working path. A building terminal database is developed to support the building partition process and interprthe etation of the tree update. This database can be dynamically updated if there are variations of terminal

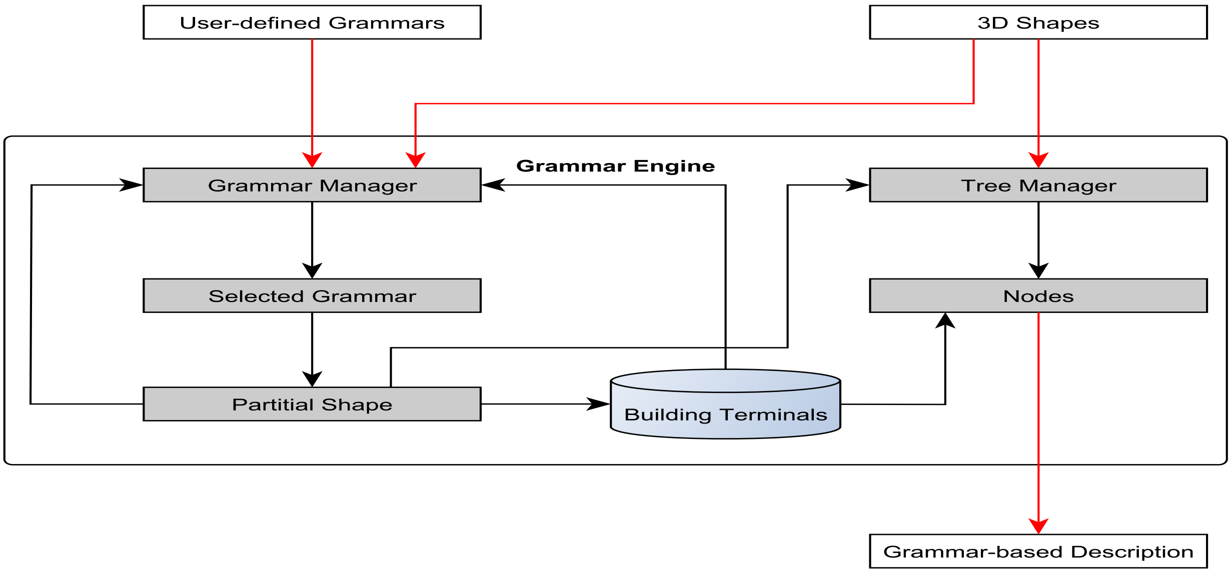


Figure 6: The main flow for proposed grammar engine is given. The block diagrams in grey indicate internal modules, and connection between internal modules uses black lines. The external module is in white block diagram and uses red lines to connect with internal modules. The blue block diagram shows a database.

symbols found during the reconstruction process, such as different shapes of window elements.

For instance, a typical reconstruction process is described as follows if one 3D shape is given to the grammar engine. The grammar manager examines the shape features and selects the appropriate rules to divide the shape into smaller partitions if it is not a building terminal. This operation is performed recursively until the 3D shape is fully divided into building terminals. These building terminals are then passed to a tree manager to create an interpretation tree, of which each leaf node represents a terminal symbols. Meanwhile the building terminal database is updated according to those generated building terminals. For certain rules i.e. repetition rules that are applied by the grammar manager, appropriate building terminals are extracted from the building terminal database accordingly.

Grammar selection is another important concern especially when the grammar and set of rules is very large. In this paper, we use a exhaustive search method to select the most likely rules among all candidates due to a small number of rules

3.4 Step 4 - Generate a building model from grammar-based model description

The grammar-based model description is represented by an interpretation tree from the model reconstruction process. However, as it is not suitable for visualisation, and hence a building model therefore needs to be generated from the interpretation tree.

A model generation algorithm is developed to efficiently place the leaf nodes from the interpretation tree in the correct order. The detailed flow chart is shown in Figure 7. As wall elements represent the fundamental structures of the building, it is important to arrange the wall elements in the right sequence so that the building framework is decided as the first priority. After the wall elements of the building is decided, associated windows and doors are placed into the extruded wall accordingly. Finally, roof elements are added into the building model.

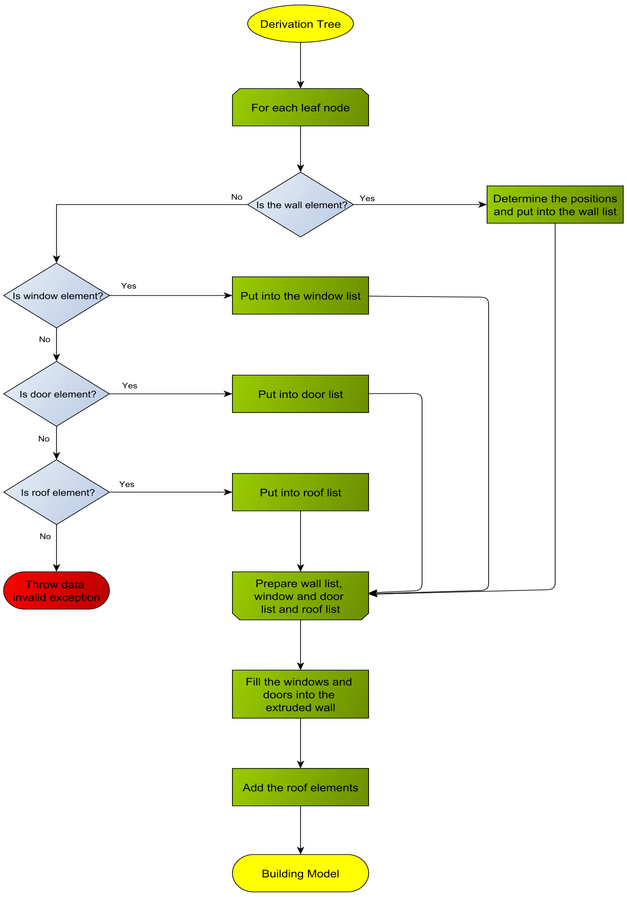


Figure 7: The main flow to generate the building model from the grammar-based interpretation tree.

In order to generate other types of building models i.e. CAD and BIM from the grammar-based model description, a format conversion algorithm needs to be developed, which is not yet considered at this stage.

4 RESULTS AND DISCUSSIONS

The proposed algorithms are implemented by JAVA and run on a dual core PC at 2.2G Hz with 8G memory.

Without loss of generality, we consider a single floor building structure. Figure 8 shows the segmented building structures formatted in DXF. Figure 9 illustrates the expected reconstructed building model from the grammar-based method. The different types of terminal elements are shown in different colours. Window elements are in blue, wall elements are in gold and the roof is in brown.

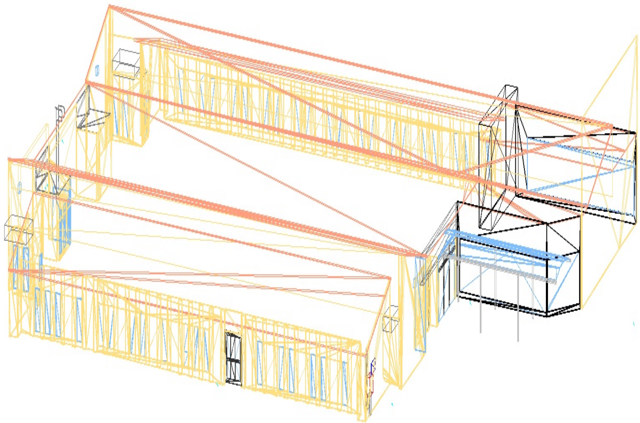


Figure 8: 3D visualisation of segmented building structures in DXF.

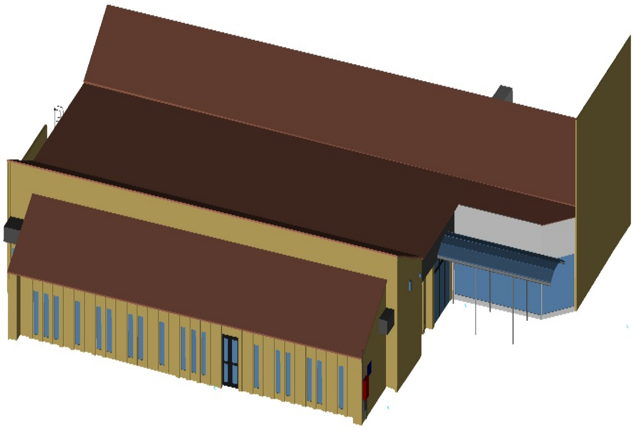


Figure 9: Expected reconstructed building model from segmented data. Different terminal elements are shown in different colours. Window elements are in blue, wall elements are in gold and the roof is in brown.

5 CONCLUSIONS

In this paper, we proposed a novel method to automatically reconstruct 3D building models from segmented data based on a pre-defined formal grammar. Our work is not limited with any specific parts of building but takes the whole building model into account. Instead of manipulating data in 2D, the proposed reconstruction process is directly derived building model from 3D shapes, which is converted from segmented data. A grammar engine is also developed to manage the automated 3D model reconstruction process by integrating the building shapes with the user-defined grammar.

As this 3D model reconstruction system is yet to be completed, there is still some potential to improve this concept. One of our interests is to deploy a more complex grammar selection algorithm in the grammar engine, which is capable of handling a large set of grammar and rules. Another interesting topic is to investigate the model assessment based on grammar-based building description i.e. interpretation tree.

ACKNOWLEDGEMENTS

The work has been supported by the Cooperative Research Centre for Spatial Information, whose activities are funded by the Australian Commonwealth's Cooperative Research Centres Programme.

REFERENCES

Becker, S., 2009. Generation and application of rules for quality dependent facade reconstruction. ISPRS Journal of Photogrammetry and Remote Sensing 64(6), pp. 640–653.

Becker, S. and Haala, N., 2009. Grammar supported facade reconstruction from mobile lidar mapping. ISPRS Workshop, CMRT09 - City Models, Roads and Traffic (2009) pp. 229–234.

DXF Reference, 2012.

Helmholz, P., , Belton, D. and Moncrieff, S., 2013. Appoach for the semiautomatic verification of 3d building models. IntArchPhRs 2013 pp. x–x.

McKay, A., Chase, S. C., Shea, K. and Chau, H. H., 2012. Spatial grammar implementation: From theory to useable software. AI EDAM (Artificial Intelligence for Engineering Design, Analysis and Manufacturing) 26, pp. 143–159.

Müller, P., Wonka, P., Haegler, S., Ulmer, A. and Gool, L. V., 2006. Procedural modeling of buildings. Proceedings of ACM SIGGRAPH 2006 / ACM Transactions on Graphics 25(3), pp. 614–623.

Musialski, P., Wonka, P., Aliaga, D. G., Wimmer, M., Gool, L. V. and Purgathofer, W., 2012. A survey of urban reconstruction. EUROGRAPHICS 2012 State of the Art Reports pp. 1–28.

Nurunnabi, A., Belton, D. and West, G., 2012. Robust segmentation for multiple planar surface extraction in laser scanning 3d point cloud data. 21st International Conference on Pattern Recognition (ICPR), 2012 pp. 1367–1370.

Parish, Y. I. H. and Müller, P., 2001. Procedural modeling of cities. Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques pp. 301–308.

Poole David, 2006. Linear algebra a modern introduction. Thompson Higher Eduction.

Prusinkiewicz, Przemyslaw, Lindenmayer and Aristid, 1990. The Algorithmic Beauty of Plants. Springer-Verlag.

Ripperda, N., 2008. Grammar based facade reconstruction using rjmcmc. Photogrammetrie Fernerkundung Geoinformation 2, pp. 83–92.

Stiny, G. and Gips, J., 1972. Shape grammars and the generative specification of painting and sculpture. Information Processing '71 pp. 1460–1465.

Wonka, P., Wimmer, M., Sillion, F. and Ribarsky, W., 2003. Instant architecture. ACM Trans. Graph. 22(3), pp. 669–677.