SEMANTICALLY ENHANCED PROTOTYPES FOR BUILDING RECONSTRUCTION

Dirk Dörschlag, Gerhard Gröger and Lutz Plümer

Institute of Geodesy and Geoinformation
University of Bonn, Germany
Meckenheimer Allee 172, 53115 Bonn
(doerschlag, groeger, pluemer)@ikg.uni-bonn.de

KEY WORDS: building reconstruction, 3d, semantics, formal grammars, minimum description length, ontology, level of detail, constraints

ABSTRACT:

We present a system for automatic building reconstruction, combining the strengths of grammars to generate varying building models and the principle of minimum description length [MDL] to evaluate results and to control the search process. The reconstruction process is guided by the level of detail, starting with a coarse level and is stepwise improved towards highly detailed models. On each level of detail, the corresponding components are identified by prototypes, provided by the building ontology. The matching between input surfaces and prototypes is supported by constraints representing relevant topological and geometrical relations between these surfaces. When employing MDL, usually an optimal coding is required. In contrast, we use an asymptotic optimal coding which is easier to generate since no a priori knowledge is needed. Instead, a larger amount of input data has to be processed to achieve comparable results.

1 INTRODUCTION

During the last years a lot of activities in the field of 3D city models could be recognized. A lot of cities, especially in Germany like Berlin, Hamburg or Stuttgart are nowadays owner of digital 3D city models and provide these for other users. There are free viewer tools like Google Earth or Aristoteles3D providing an intuitive interface for accessing this data. But not only the number of 3D city models and visualization tools increases significantly, there are also several new standards for city and building models, e.g. the city geography markup language (CityGML) (Kolbe et al., 2005, Gröger et al., 2006) and the Industry Foundation Classes (IFC) (Eastman, 1999).

The first one was developed within the geographic information community and is being standardized within the Open Geospatial Consortium (OGC), one of the most important organizations in this field. The second one was developed within the architectural community. They are intended to provide interfaces to interchange these models, including the semantics and the topology belonging to the geometric models. At the moment, several projects are using these standards to specify their input and interchange format. One of these is part of the testbed for OGC Web Services Phase 4 (OGC, 2006) of the open geospatial consortium which is evaluating the use of service oriented architectures for 3D city models to achieve better response capabilities in the context of homeland security. Another project is initiated by the government of North Rhine-Westphalia (Germany). It currently uses a similar web service architecture to fulfill the requirements within the noise prevention program of the European Union. Each of these projects requires both, a detailed geometry and assigned semantic information. So nowadays city models do not only have to look good they also have to be smart to enable advanced analytic processing. Due to the requirements of users, projects and applications, methods are needed to acquire 3D city models. Requirements for these methods are:

1. they have to be automatic to enable the generation and update of large-area models within short time frames
2. they need to reconstruct both, geometric and semantic properties of urban objects on a high level of detail
3. and they should generate quality information for each reconstructed object.

Within the literature on automatic building reconstruction from different data sources, several different methods could be recognized. They could be differentiated by the focused level of detail, the use of semantic information during the reconstruction process and the used input data. For 3D city models, these 3 levels of detail (LoD) are commonly used: LoD1 is the well-known blocks model, LoD2 adds roof structures and LoD3 completes LoD2 by including balconies, roof structures like dormers and chimneys (Kolbe et al., 2005). (Haala, 2005) uses 2D ground plans of buildings and digital surface models (DSM) to derive LoD2 buildings. The method proposed in (Fischer et al., 1998) and (Kolbe, 1999) uses pairs of aerial stereo images and semantically founded geometric constraints to reconstruct LoD2 building models. (Grau, 2000) proposes a multi step method using semantic networks to generate first LoD2 building hypothesis and extending these in a second step to LoD3 models by adding e.g. dormers. Those models are evaluated by least square matching with terrestrial stereo images. Both (Brenner and Haala, 1998) and (Vosselman and Dijkman, 2001) employ 2d ground plans, DSMs and some volume primitives to derive LoD2 building model hypotheses. Both approaches differ in the used evaluation method. The first uses least square matching while the second one uses the minimum description length principle, which will be considered in detail in section 2.4. A method similar to (Brenner and Haala, 1998) is presented by (Stilla and Jurkiewicz, 1999). Beside these automatic methods there exist several approaches for interactive, semi-automatic reconstruction of buildings, e.g. (Grün and Wang, 1999, Rottensteiner, 2001, Gülch et al., 1999). Since the focus of this paper is on the automatic reconstruction of the semantical and geometrical properties of urban objects, these approaches are not considered any further. In our approach we combine the strengths of spatial grammars to generate building models, the principle of minimum description length as evaluation function to control and guide the search process and we employ a multi scale approach starting with a coarse level of detail and stepwise refinement towards highly detailed models. We used building ontology for grammar rule design and the definition of our levels of detail. A widely known type of grammar in the context of plant model generation are L-Systems as discussed in (Prusinkiewicz and Lindenmayer, 1990). This type of grammar was used by (Parish and Mueller, 2001) to generate simple building models. Within the paper (Müller et al., 2006) other types of grammar evolved in the context of architecture and design, such as set grammars (Stiny, 1982) and split grammars (Duarte, 2002, Wonka et al., 2003), are used to generate typical, artificial, fictitious building models for certain contexts, e.g. how the ancient Rome may have looked like (Müller et al., 2005). A crucial problem in building reconstruction is the evaluation of alternative modeling possibilities. A method to deal with this problem is the principle of minimum description length (MDL) (Grünwald et al., 2005). This criterion considers both, the goodness of fit between the model and the data and the complexity of the used model. The use of MDL as evaluation function in our approach is described in section 2.4.

2 METHODS

The method presented in this paper aims at the field of automatic building reconstruction. In contrast to others, we combine the strengths of grammars to generate building models, the principle of minimum description length as evaluation function to control and guide the search process and we employ a multi scale approach starting with a coarse level of detail and stepwise refinement toward highly detailed models. We use building ontology for grammar rule and symbol design and the definition of our specific levels of detail. Important aspects we focus on are formal grammars, the constraint graphs used and finally the model selection criterion MDL.

2.1 Input data

Input data for our approach are terrestrial laser scans of a single suburban building supplemented by an aerial laser scan of the same building.

This cloud of noisy 3D points needs to be preprocessed to remove outliers and to reconstruct the interrelationship between the points. Given the points and the corresponding plane, the boundaries of polygons have to be derived. These may have also interior boundaries.

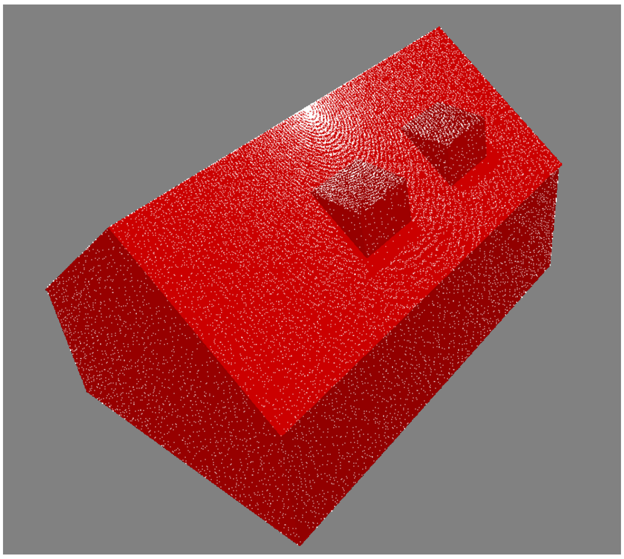

Figure 1: point cloud from a synthetic laser scan and the available 3D polygons

The planes are derived by applying a RANSAC procedure as described in e.g. (Wahl et al., 2005) to the point cloud. After the preprocessing step, we obtain the following:

1. 3d planar polygons, each with one outer boundary
2. the assignment of 3D points to corresponding polygons
3. the assignment of planes to polygons. For each polygon, the primary components and the equation of this planes are stored

Due to the characteristics of laser scanners, the edges of buildings are not observable directly. Instead, they have to be derived by intersection the corresponding planes. This will have consequences for the presented approach, which will be discussed in more detail in section 2.2.

An impression of the input data is shown in figure 1.

2.2 Constraint graph and prototypes

Within any geometric representation of objects, there exists additionally several geometric and topological constraints between modeling primitives like nodes, edges and faces.

The most important observation in this context is that there exist constraints which are invariant against certain types of transformations like scaling, translation and rotation of objects. E.g. given an object consisting of two planes which are parallel, if one of the transformation mentioned before is applied to both planes, then these two still remain parallel. This characteristic of constraints is very important and one of the main reasons why a certain constraint is used in the following.

The modeling primitives and the constraints can be represented in a graph structure. Within such a structure, the modeling primitives are represented by nodes and the constraints are represented as edges between the two nodes they apply to. This graph structure is called constraint graph in the following. This concept was used by (Kolbe, 1999) to match geometric shapes within a 2D context and is part of the weak CSG primitive concept introduced by (Brenner, 2004). In contrast to both papers, the constraint graphs used and discussed here are always embedded in the 3d space and describe e.g. geometric constraints between 3D faces like parallelism between planes.

A constraint graph representation is derived for the given input data. This graph is called data constraint graph (DCG). Within the derived structure any node represents 3D polygons, and the edge represent the ascertainable constraints. An extract of the 3D data constraint graph for a scanned L-shaped building is shown in figure 2. In this figure, any edge represents a 3D polygon and the constraint holds for the polygons or the planes they belong to or both.

The different compartments of the ontological structure of buildings, like storeys or dormers, have to be linked to geometry observable within the input data. These compartments have to be bounded by closed and topologically correct surfaces. Following the idea of the constructive solid geometry, that complex solid can be represented by a set of parameterized instances of solid primitives and a set of boolean operations on them, and be represented in a tree structure (CSG tree), for any compartment typical simple solid were identified. This association of a geometric primitive and a semantic interpretation is called semantically enhanced prototype or prototype for short. An extract of the 3D data constraint graph for a prototype with a cuboid as geometric primitive is shown in figure 3. In this figure, any edge represents a 3D polygon and the constraint holds for these polygons or the planes they belong to or both. Due to the fact that the geometric primitives of the prototypes are decomposable into modeling primitives and a set of constraints that holds between them, it is possible to derive a constraint graph representation for any prototype. The semantic interpretation of a prototype furthermore allows to predict the constraints that have to be introduced if one

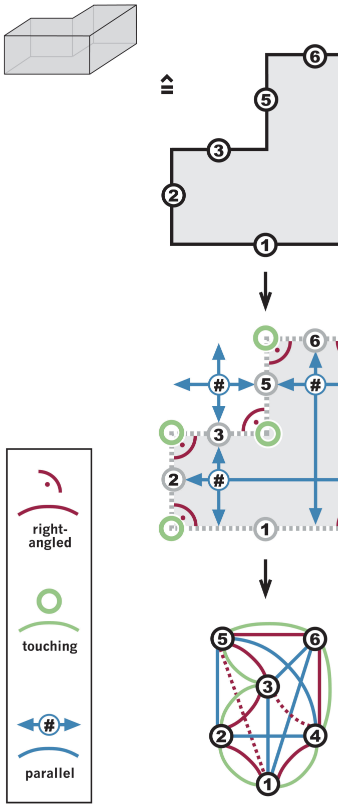

Figure 2: Constraint Graph reconstructible for the wall surface polygons of a house

or more prototype constraint graphs (PCG) have to be integrated into an existing aggregate of PCGs. Because of the special capabilities of constraints mentioned above, each of these prototype constraint graphs (PCG) could be compared with subgraphs of the DCG. If both match, the identified part of the DCG represents an instance of the geometric primitive associated with this PCG and the parameter values for this primitive could be estimated by using the geometry associated with the DCG nodes. Figure 4 illustrates this matching procedure. In this figure, any node represents a vertical 3D polygon and the constraint holds for these polygons or the planes they belong to or both.

One important observation that will be from interest later is that this holds for the matching of more complex constraint graphs with the DCG, too.

Due to the uncertainties of the input data the methods used to derive the data constraint graph (DCG) are modified to be able to handle these uncertainties. Other effects of the uncertainties and the general setting during data acquisition influence the calculation of the matching quality between the prototype constraint graphs and the data constraint graph. The handling of these influences will be discussed in section 2.4, where the principle of minimum description length (MDL) as evaluation function will be presented.

2.3 A grammar for building generation

During the last decades, the connection between grammars and spatial design was addressed by several research groups. Within this section a brief introduction to grammars is given. Several spatial grammars, like set grammars (Stiny, 1980, Stiny, 1982), shape grammars (Stiny, 1980) and structure grammars (Carlson et al., 1991), are available and discussed in the context of the generation of spatial objects.

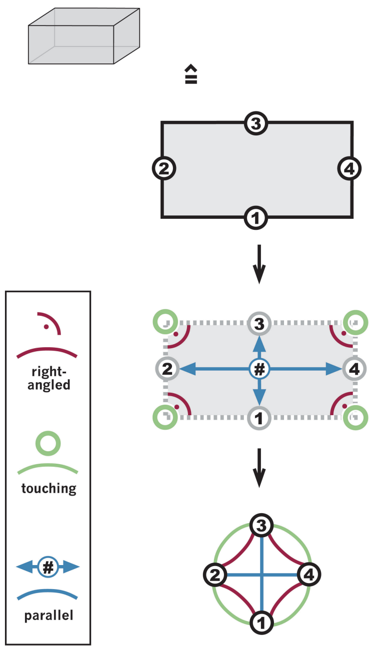

Figure 3: constraint graph within a solid primitive of a cuboid

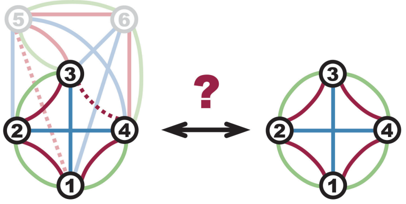

Figure 4: Mapping the constraint graph of the cuboid into the one of the house

The grammar used in our approach (called BG grammar) is a context-free grammar as defined by (Chomsky, 1959). This was extended to an attributed grammar in the way given by (Knuth, 1968, Knuth, 1971). The following definitions are the basis for the grammar we developed:

Definition 2.1 A **formal grammar** G consists of the quad-tuple (N, T, S, P). N is the finite set of nonterminal symbols, T the finite set of terminal symbols with $T \cap N = \emptyset$, $S \in N$ a distinguished start symbol and P a finite set of production rules of the form $(T \cup N)^* N (T \cup N)^* \to (T \cup N)^*$[1]. The alphabet V of the grammar G is defined with $V = N \cup T$. The language of a formal grammar $G = (N, T, P, S)$, denoted as $L(G)$, is defined as all those strings over T that can be generated by starting with the start symbol S and then applying the production rules in P until no more nonterminal symbols are present.

Definition 2.2 A **context-free grammar** is a formal grammar in which the left-hand side of each production rule consists of only a single nonterminal symbol.

Definition 2.3 An **attributed grammar** is a context-free grammar, were for any symbol $x \in V$ with $V = N \cup T$ a set

---

[1] S^* terms any final concatenations of the set S including the empty symbol ϵ, S^+ terms any final concatenations of the set S without the empty symbol ϵ

of attributes $\alpha(x)$ exists. For every production rule $p \in P$ there exists a set of semantic rules $R(p)$ of the form $x_i.a = f(x_j.b, ..., x_k.c)$, were $x_i, ..., x_k$ are the occurring symbols within the production rule and $a...c \in \alpha(x)$. For each x within a derivation at least one semantic rule $R(p)$ is applicable.

After recapitulating the basic notions, we now are in a position to present our grammar for the reconstruction of buildings. First we define how the result of the reconstruction process is represented:

Definition 2.4 A **reconstructed constraint graph (RCG)** is a constraint graph , which consists of a set of PCGs connected at common nodes, and two partial functions v, e mapping the vertices of the RGC to the vertices of the DCG, and the edges of the RCG to the edges of the DCG.

The RCG represents the building reconstructed so far and consists of PCGs as subgraph, where each PSG represents a prototype. The functions v associates faces in the RCG with the faces observed in the data, while the function e assign the information whether a constraint in the RSG is observable in the data. Since not all faces and not all constraint in the RCG are observable in the data, both v and e have to be partial functions. The grammar which produces a RCG is a special attributed grammar called BG grammar (building generating grammar) which is specified as follows:

Definition 2.5 A **BG grammar** is a attributed grammar, were the set N of nonterminal symbols is is partitioned in the set PCG of prototype constraint graphs, the set J of junctions and a start symbol $\{S\}$. The set T of terminal symbols is is partitioned in the set TJ of terminal junctions and the set TP of prototypes. The production rules have one of the following three forms:

1. $S \rightarrow PCG$
2. $PCG \rightarrow TP(J)^*$
3. $J \rightarrow TJ$
4. $J \rightarrow PCG$

A junction in the set J represents a part of the hull of the building reconstructed so far, which is denotes a discrepancy between the RCG and the DCG. A junction indicates a missing prototype, which is usually identified in one of the next production steps. If a junction can not be assigned finally, it is replaced by a symbol from the set TJ by applying a rule of the third form.

The reconstruction process using a BG grammar is illustrated now by applying it to the L-shaped building which was already given in figure 2. The process starts with the matching of a PCG (see bottom of figure 3) with the DCG (see bottom of figure 2), by applying a production $S \rightarrow PCG_c$ of the first form. PCG_1 represents the constraint graph of the first detected prototype, e.g. a cuboid. The result is a RCG reflecting the detected polygons and constraints. The corresponding derivation tree is depicted in figure 5. The current string produced by the grammar so far is PCG_1. To this string, a rule $PCG_c \rightarrow cRJ$ of the second form is applied, where c is a cuboid. The resulting string is c_1RJ, where the RJ indicates a missing prototype. In the next step, a rule $RJ \rightarrow PCG_c$ is applied, which yields the string c_1PCG_c. Finally, the rule $PCG_c \rightarrow c$ completes the process. The generated string is c_1c_2, where the geometric information is represented in the attributes of c_1 and c_2. If during the reconstruction process more than one area is not observable, more than one junction is needed. In the rule of the second form, this is reflected by using the star notation.

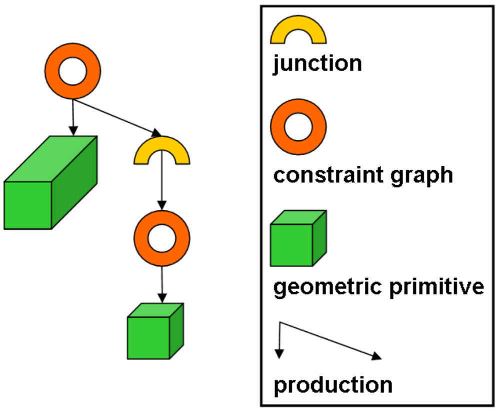

Figure 5: Derivation tree for the first reconstruction level of a L shaped building

In the general case typically several rules are applicable to one derived string. Here a selection method is required, which chooses the "best" rule. This selection method is topic of the following section.

2.4 MDL for decisions

As already stated in section 2.3, the process of applying production rules is nondeterministic, since in one step more than one rule may be applied. It is possible to exhaustively generate all possible building models, but this of course is too time consuming. Thus decision mechanisms are required. Within the literature, several selection criteria are discussed (Akaike, 1974, Schwarz, 1978, Rissanen, 1978). For our approach we have chosen the principle of minimum description length [MDL] (Grünwald et al., 2005). The suitability of MDL for building reconstruction was already demonstrated by (Kolbe, 1999, Vosselman and Dijkman, 2001). MDL is an information theoretic model selection criterion. It incorporates the goodness-of-fit between the observed data and the model, and the complexity of the fitted model. Especially the last point is important for the intended use, since we are operating on biased real world data and do not want to transfer the bias into model parts. The formulation of MDL used within our approach is due to (Vosselmann, 1992):

$$\hat{M} = \arg\max_{M_i} I(D; M_i) - I(M_i) \qquad (1)$$

where $\hat{M}$ is the selected model, $I(D; M_i)$ the mutual information between data D and model M_i and $I(M_i)$ the information of the model M_i. This formulation of the MDL criterion provides several advantages for the use within our approach. The most important one is, that wild card assignments do not have any effect on the criterion (Vosselmann, 1992). This means, they neither support the mapping between the model and the data nor contradict it. A wild card assignment is required for this approach because the input data cannot be assumed as complete.

The data is not complete because parts of the surfaces of the prototypes are occluded by other prototypes, a situation which leads to the junctions defined within the grammar section (2.3). Another reason for their occurrence lies in the surveying situation. For each of these cases there exist nothing mappable for some compartments within the model, which then will be mapped to virtual objects called wild cards.

Within our approach we use MDL as a criterion to measure the quality of the mapping between the models generated by our grammars and the input data. One possible way is to derive the probabilities of occurrence $P(x)$ of any character x of an alphabet X, since the information $I(x)$ is defined as (Cover and Thomas, 2006):

$$I(x) = -log P(x) \tag{2}$$

Within the constraint graphs mentioned in section 2.2 the probabilities of occurrence for any edge and node in all occurring graphs have to be derived. Due to the large variety of buildings and a high impact of regional characteristics of building types, it is a difficult and cumbersome process to learn the required information from training data. Since any set of training data is only related to a specific region and not globally valid, these values are always biased. Under the assumption of optimal coding the probability can be replaced by the code length $L_C(x)$ of x encoded with the code C (Cover and Thomas, 2006):

$$L_C(x) = I(x) = -log P(x) \tag{3}$$

Up to this point the problem of deriving the a priori probabilities correctly is the same as in equation 2, because it is required to build an optimal code. A possibility to face this problem is the use of asymptotic optimal codes. One compression technique producing asymptotic optimal codes is the Lempel-Ziv-Welch-compression introduced by (Welch, 1984). A code is asymptotic optimal, if the redundancy approaches zero whenever the source code length tends to infinity (Lelewer and Hirschberg, 1987). While LZW-compression is a codebook based compression technique, an optimization of the initial codebook can be used to reduce the source code length required for initial learning. An initial codebook for the approach discussed within this paper contains the different geometric components and constraints forming the constraint graphs and it contains the constraint graphs of the primitives in use. These elements have to be entered in that way that the most frequent element appears first and the most infrequent appears last.

Within our approach the LZW-compression will be used to derive the required code length $L_C(x)$ of the model and its matched parts.

2.5 System architecture

In the last sections we have presented the components of the building reconstruction process. Now these components are integrated to obtain the whole procedure.

The process is composed of several steps, which correspond to scales. Starting with the coarsest scale, form one step to another the scale becomes more detailed. Usually we consider four scales: In the coarsest scale no. 1 the storeys except attics are detected. In scale no. 2 the attics are reconstructed, in scale no. 3 the larger building characteristics like dormers and balconies, and in scale no 4 doors and windows are detected.

For each scale, the components defined so far are specialized. A BG grammar is specified for each scale, obtaining different grammars BG_{scale} which differ in the prototypes and their semantics. Furthermore, the data constraint graph is specialized for each scale. Specific scale-dependent criteria are used to derive these constraint graphs.

The scale oriented procedure is depicted in figure 6; The procedure starts with scale 1 by matching the specialized PCG_{scale} with the specialized DCG_{scale}. Then the rules of the grammar BG_{scale} are applied as described in section 2.3, until there is no nonterminal symbol left. Now the scale is incremented and the procedure is repeated, until all scales are considered.

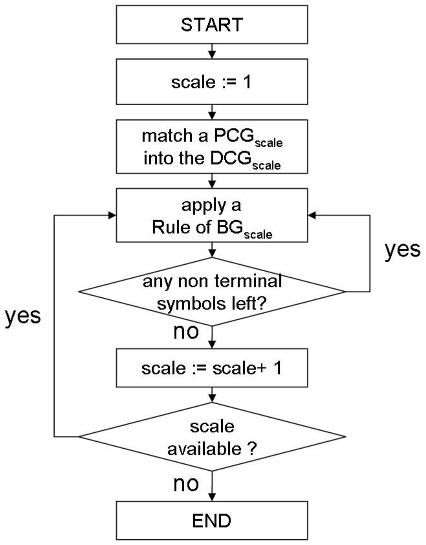

Figure 6: The building reconstruction procedure.

3 CONCLUSIONS AND FUTURE WORK

In this paper we have presented the outline of a procedure for automatic building reconstruction. It enables the extraction of semantic building models from heterogeneous input data which passed a preprocessing step. While the input data could be points from laser scanning or existing building models without semantic information, the preprocessing step produces 3D planar polygons and connected planes from the input data. The reconstruction itself is a multi step process where each step corresponds to a level of detail. Each reconstruction step uses a specific set of symbols, production rules and semantic rules. To guide the generation process and reduce the search space, the principle of minimum description length is employed to select the best alternative and it is used as a termination criterion. These steps are embedded in a control structure, which decides which step has to be performed next. Semantic information, gathered from ontologies are used to define the objects and get a priori information about their specific attributes. The system offers a mechanism to reconstruct buildings and their structural elements on a high level of detail in a generic and flexible way. Future work will include the implementation and evaluation of the presented approach. Another aspect is the extension of the set of available solid primitives within the grammars to be able to reconstruct e.g. pillars or downpipes. More reconstructible semantical parts of buildings have to be specified. Another point of future work will be the derivation of typical value ranges for the attributes and spatial extends of the reconstructed components. This knowledge seems to be helpful to improve the efficiency of the outlined process.

4 ACKNOWLEDGEMENTS

This research was supported by the German Research Foundation (Deutsche Forschungsgemeinschaft, DFG) as part of the bundle project entitled "Abstraction of Geographic Information within the Multi-Scale Acquisition, Administration, Analysis and Visualization" (FO 180/10-1).

REFERENCES

Akaike, H., 1974. A new look at the statistical model identification. IEEE Transactions on Automatic Control 19(6), pp. 716–723.

Brenner, C., 2004. Modelling 3d objects using weak csg primitives. nternational Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 35(3), pp. 1085–1090.

Brenner, C. and Haala, N., 1998. Fast production of virtual reality city models. IAPRS 32, pp. 77–84.

Carlson, C., Woodbury, R. and McKelvey, R., 1991. An introduction to structure and structure grammars. Environment and Planning B: Planning and Design 18(4), pp. 417 – 426.

Chomsky, N., 1959. On certain formal properties of grammars. Information and Control 2, pp. 137–167.

Cover, T. M. and Thomas, J. A., 2006. Elements of Information Theory. 2 edn, Wiley-Interscience.

Duarte, J., 2002. Malagueira Grammar - towards a tool for customizing Alvaro Siza's mass house at Malagueira. PhD thesis, MIT School of Architecture and Planning.

Eastman, C. M., 1999. Building Product Models: Computer Environments, Supporting Design and Construction. CRC.

Fischer, A., Kolbe, T. H., Lang, F., Cremers, A. B., Förstner, W., Plümer, L. and Steinhage, V., 1998. Extracting buildings from aerial images using hierarchical aggregation in 2d and 3d. Computer Vision and Image Understanding: CVIU 72(2), pp. 185–203.

Gülch, E., Müller, H. and Läbe, T., 1999. Integration of automatic processes into semi-automatic building extraction. In: Proceedings of ISPRS Conference "Automatic Extraction Of GIS Objects From Digital Imagery".

Grau, O., 2000. Wissensbasierte 3D-Analyse von Gebäudeszenen aus mehreren frei gewählten Stereofotos. PhD thesis, Universität Hannover.

Gröger, G., Kolbe, T. H. and Czerwinski, A. (eds), 2006. OpenGIS City Geography Markup Language (CityGML), Implementation Specification Version 0.3.0, Discussion Paper, OGC Doc. No. 06-057. Open Geospatial Consortium.

Grün, A. and Wang, X., 1999. Cybercity modeler, a tool for interactive 3-d city model generation ,. In: Fritsch and Spiller (eds), Photogrammetrische Woche 1999, Photogrammetrische Woche 1999.

Grünwald, P. D., Myung, I. J., Pitt, M. A., Balasubramanian, V., Lanterman, A. D., Hanson, A. J., Fu, P. C., P-Vitányi, Barron, A., Liang, F., Foster, D. P., Stine, R. A., Yamanishi, K., Rissanen, J., Tabus, I., Comley, J. W., Dowe, D. L., Jörnsten, R., Yu, B., Kontkanen, P., Myllymäki, P., Buntine, W., Tirri, H., Lee, M. D., j. Navarro, D., Charter, N. and Su, Y., 2005. Advances in Minimum Description Length. MIT Press.

Haala, N., 2005. Multi-Sensor-Photogrammetrie - Vision oder Wirklichkeit? PhD thesis, Universität Stuttgart.

Knuth, D. E., 1968. Semantics of context-free languages. Theory of Computing Systems 2(2), pp. 127–145.

Knuth, D. E., 1971. Top-down syntax analysis. Acta Informatica 1(2), pp. 79–110.

Kolbe, T. H., 1999. Identifikation und Rekonstruktion von Gebäuden in Luftbildern mittels unscharfer Constraints. PhD thesis, Hochschule Vechta.

Kolbe, T. H., Gröger, G. and Plümer, L., 2005. Citygml – interoperable access to 3d city models. In: Geo-information for Disaster Management. Proc. of the 1st International Symposium on Geo-information for Disaster Management.

Lelewer, D. A. and Hirschberg, D. S., 1987. Data compression. ACM Computing Surveys 19(3), pp. 261–296.

Müller, P., Vereenooghe, T., Ulmer, A. and Gool, L. V., 2005. Automatic reconstruction of roman housing architecture. In: International Workshop on Recording, Modeling and Visualization of Cultural Heritage, Balkema Publishers (Taylor & Francis group).

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

OGC, 2006. Ogc web services phase 4. Technical report, Open Geospatial Consortium. http://www.opengeospatial.org/projects/initiatives/ows-4 (last visited 30.03.2007).

Parish, Y. and Mueller, P., 2001. Procedural modeling of cities. In: E. Fiume (ed.), Proceedings of ACM SIGGRAPH 2001, ACM Press / ACM SIGGRAPH, pp. 301–308.

Prusinkiewicz, P. and Lindenmayer, A., 1990. The Algorithmic Beauty of Plants. Springer.

Rissanen, J., 1978. Modeling by the shortest data description. Automatica 14, pp. 465–471.

Rottensteiner, F., 2001. Semi-automatic extraction of buildings based on hybrid adjustment using 3D surface models and management of building data in a TIS. PhD thesis, Technische Universität Wien.

Schwarz, G., 1978. Estimating the dimension of a model. Annals of Statistics 6(2), pp. 461–464.

Stilla, U. and Jurkiewicz, K., 1999. Reconstruction of building models from maps and laser altimeter data. In: P. Agouris and A. Stefanidis (eds), Integrated spatial databases:Digital images and GIS, Springer, pp. 34–46.

Stiny, G., 1980. Introduction to shape and shape grammars. Environment and Planning B 7, pp. 343–361.

Stiny, G., 1982. Spatial relations and grammars. Environment and Planning B 9, pp. 313–314.

Vosselman, G. and Dijkman, S. T., 2001. 3d building model reconstruction from point clouds and ground plans. International Archivs Photogrammetry and Remote Sensing XXXIV part 3/1, pp. 339–345.

Vosselmann, G., 1992. Relational matching. Vol. 628, lect. not. comp. sci. edn, Springer.

Wahl, R., Guthe, M. and Klein, R., 2005. Identifying planes in point-clouds for efficient hybrid rendering.

Welch, T. A., 1984. A technique for high-performance data compression. Computer 17(6), pp. 8–19.

Wonka, P., Wimmer, M., Sillion, F. and Ribarsky, W., 2003. Instant architecture. ACM Transactions on Graphics 22, pp. 669–677.