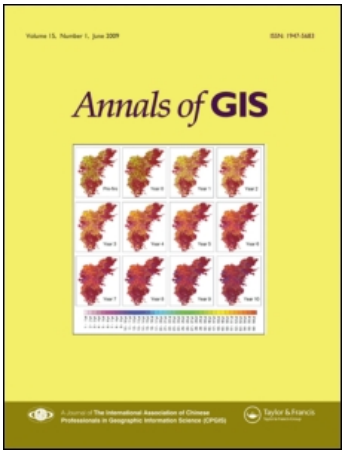

Annals of GIS
Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t909450018

Integration of conditional random fields and attribute grammars for range data interpretation of man-made objects

Jörg Schmittwilken [a]; Michael Ying Yang [b]; Wolfgang Förstner [b]; Lutz Plümer [a]
[a] Department of Geoinformation, Bonn University, Bonn, Germany [b] Department of Photogrammetry, Bonn University, Bonn, Germany

Online publication date: 14 December 2009

To cite this Article Schmittwilken, Jörg, Yang, Michael Ying, Förstner, Wolfgang and Plümer, Lutz(2009) 'Integration of conditional random fields and attribute grammars for range data interpretation of man-made objects', Annals of GIS, 15: 2, 117 — 126
To link to this Article: DOI: 10.1080/19475680903464696
URL: http://dx.doi.org/10.1080/19475680903464696

Annals of GIS
Vol. 15, No. 2, December 2009, 117–126

Taylor & Francis
Taylor & Francis Group

Integration of conditional random fields and attribute grammars for range data interpretation of man-made objects

Jörg Schmittwilken[a]*, Michael Ying Yang[b], Wolfgang Förstner[b] and Lutz Plümer[a]

[a]*Department of Geoinformation, Bonn University, Bonn, Germany;* [b]*Department of Photogrammetry, Bonn University, Bonn, Germany*

(*Received 15 August 2009; final version received 14 October 2009*)

A new concept for the integration of low- and high-level reasoning for the interpretation of images of man-made objects is described. The focus is on the 3D reconstruction of facades, especially the transition area between buildings and the surrounding ground. The aim is the identification of semantically meaningful objects such as stairs, entrances, and windows. A low-level module based on random sample consensus (RANSAC) algorithm generates planar polygonal patches. Conditional random fields (CRFs) are used for their classification, based on local neighborhood and priors from the grammar. An attribute grammar is used to represent semantic knowledge including object partonomy and observable geometric constraints. The AND-OR tree-based parser uses the precision of the classified patches to control the reconstruction process and to optimize the sampling mechanism of RANSAC. Although CRFs are close to data, attribute grammars make the high-level structure of objects explicit and translate semantic knowledge in observable geometric constraints. Our approach combines top-down and bottom-up reasoning by integrating CRF and attribute grammars and thus exploits the complementary strengths of these methods.

Keywords: attribute grammars; conditional random fields; range data; facade interpretation; high- and low-level integration

1. Introduction

In this article, we describe a concept for integrating low- and high-level reasoning for the interpretation of images of man-made objects. Our focus is on interpreting range data of building facades, especially the transition area between the building and the surrounding ground, what we call the 'building collar'. The complexity of man-made objects requires flexible interpretation techniques that can handle the semantic knowledge about the domain as well as the richness of the appearance of all details in the image data. The variability of the number of parts and their hierarchical and neighborhood relations, which occur similarly also in natural language understanding, can be described efficiently with grammars that represent the semantic high-level structure of the scene together with random fields, which can efficiently cope with the fusion of structural knowledge and sensor data.

Our research is motivated by the urgent need to enrich 3D city models, which are mainly used for visualization purposes, by thematic attributes to eventually obtain truly 3D geoinformation systems for complete cities. Because Google and Microsoft provide worldwide access to spatial data, also 3D for an increasing number of cities around the world, the relevance of truly spatial information becomes obvious. Applications are manifold: car and pedestrian navigation, access analysis for fire brigades, location planning for industry, microclimate investigations, or risk analysis.

For a long time, the difficulty of interpreting range and intensity data has been underestimated. The main reason is the high variability of man-made structures and their appearance, and the resulting complexity of the acquired data allowing us to identify objects but also to model object parts, which, in general, are not part of a 3D GIS.

Early attempts in 3D city modeling were based on sets of prototypes or parameterized geometrical models (Fischer *et al.* 1997) with the possibility of aggregation (Fischer *et al.* 1999), on the restriction to roof structures (Brenner *et al.* 2001) made possible by using the ground plans of the buildings from a 2D GIS. Practical approaches were clearly interactive, for example, 'InJect' (Guelch 2001), 'CyberCity Modeler' (Gruen and Wang 1999), with some support by automatic procedures. Modeling the architecture of complete building blocks by using generative models (Dick *et al.* 2004) pushed theoretical research onto a new level. Mobile mapping systems increasingly provide terrestrial data, which changed the focus on facades. Because of their specific structure, models based on grammatical rules were developed, exploiting the long tradition in natural language understanding. Stochastic attribute grammars (Abney 1997) have evolved and today appear as generalizations of Markov random fields (MRFs) and Bayesian networks (cf. Liang *et al.* 2009).

Parallel to these developments aiming at a semantically complete model of a scene, attempts were made to exploit

*Corresponding author. Email: schmittwilken@igg.uni-bonn.de

ISSN 1947-5683 print/ISSN 1947-5691 online

DOI: 10.1080/19475680903464696
http://www.informaworld.com

the neighborhood structure for semantic image partitioning by using random fields. MRFs have been used for image interpretation since 1992 (Modestino and Zhang 1992); their limiting factor that they only allow for local image features has been overcome by conditional random fields (CRFs) (Kumar and Hebert 2003), where arbitrary features can be used for classification, at the expense of a purely discriminative approach.

This article describes an approach attempting to exploit the strengths of both models by integrating CRFs with attribute grammars, thereby exploiting their potential for the interpretation of range images of man-made objects, especially in the area around an entrance with stairs and windows as dominant scene objects.

The article is organized as follows: In Section 2, the concept of the system's structure and the communication between modules are introduced and the main contributions of the article are presented. Low- and high-level reasoning, the notations of CRF, and attribute grammar are described in Section 3. Section 4 presents the integration of low- and high-level reasoning for range data interpretation. The experimental results are given in Section 5, followed by the concluding remarks.

2. Concept

The range data we address in this article may come from many ways: laser scanning, point clouds generated from videos (Heinrichs *et al.* 2008), and so on. Our focus is on highly detailed 3D building models with focus on facades, that is, the (front) part of the building. As facades are well structured, frequently symmetrical and often even prettily decorated, they combine both good-natured and malicious properties concerning their automatic reconstruction.

In the following, we explain the system's structure, the communication between modules, and present the highlights of characteristics, respectively.

2.1. *System's structure*

It includes three parts (see Figure 1): low-level module, CRF module, and grammar module. The low-level module generates planar patches using the random sample consensus (RANSAC) algorithm (Fischler and Bolles 1981) from 3D point cloud. Here, planar patches refer to sets of points that lie within a plane and are within a certain local range. In the CRF module, a CRF is used as a discriminative neighborhood model of the facade. Patch labels are classified with respect to their local neighborhood. In the grammar module, an attribute grammar is used as a semantic model of the whole building with focus on the facade. Some of the symbols can be interpreted geometrically. The reconstruction of the 3D point cloud is done by a special parser based on the given grammar. The parsing process also uses a RANSAC-based algorithm to estimate the parameters of the symbols with a geometric interpretation.

2.2. *Communication*

The communication between the modules is as follows: Initially, the low-level module generates patches from 3D point cloud, that is, planar polygons and their normal vectors. The CRFs operate on these planar patches. They are used to estimate the class labels of the patches with respect to their local neighborhood and priors from the grammar module. The parser uses the probability of the classified patches to control the reconstruction and to optimize the sampling mechanism of RANSAC. The whole proposed scheme is illustrated in Figure 1.

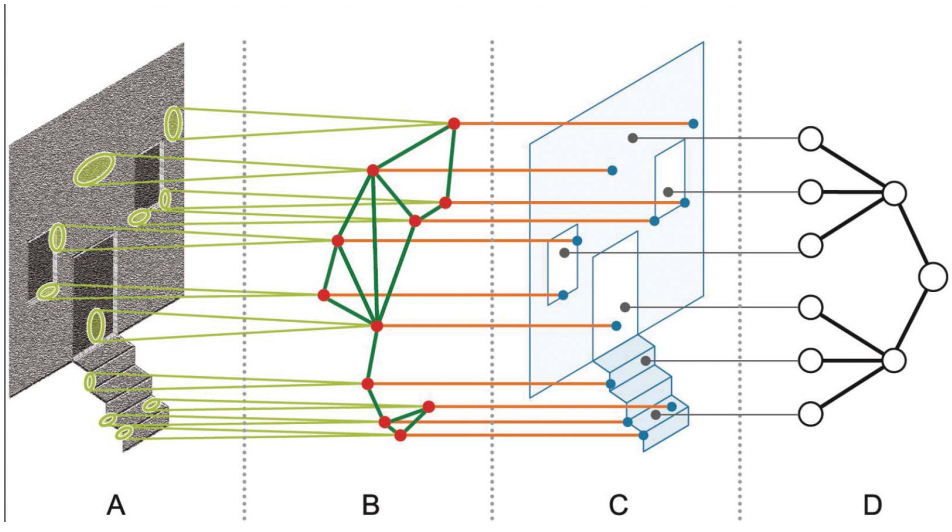

Figure 1. System's structure. (a) Range data with sites referring to planar patches. (b) Graph for conditional random field, referring to sites and their neighborhoods. (c) Reconstructed 3D structure based on the result of the CRF and (d) the derivation tree generated by the grammar.

2.3. Characteristics

We present means for geometric and semantic reconstruction in this article. CRF is close to the data but limited to local neighborhoods, whereas the attribute grammar is close to the semantic model but limited to *a priori* probabilities. Our contribution is highlighted by the integration of these two bottom-up and top-down methods. We make use of the individual strength of each method and remove their individual weakness by the complementary combination.

3. Low- and high-level reasoning

3.1. Conditional random fields

CRFs have been proposed as a discriminative model for taking into account the interactions between neighboring elements during classification. CRFs are used in a discriminative framework to model the posterior over the labels x given the observations $\boldsymbol{y}$ (Lafferty *et al.* 2001), and thus provide full freedom in exploiting observation data. The CRF framework has already been used to obtain promising results in a number of domains where there is interaction between labels, including tagging, parsing, and information extraction in natural language processing (McCallum *et al.* 2003), as well as in the modeling of spatial dependencies in image processing (Kumar and Hebert 2003). In this article, the considered CRF (Kumar and Hebert 2003) is a distribution of the form given below:

$$P(\boldsymbol{x}|\boldsymbol{y}) = \frac{1}{Z}\exp\left(\sum_{i\in S} A_i(x_i, \boldsymbol{y}, \boldsymbol{w}) + \sum_{i\in S}\sum_{i\in N_i} I_{ij}(x_i, x_j, \boldsymbol{y}, \boldsymbol{v})\right) \tag{1}$$

where $\boldsymbol{y} = \{\boldsymbol{y}_i\}_{i\in S}$ are observations, $\boldsymbol{y}_i$ is the observation from the site i, $\boldsymbol{x} = \{x_i\}_{i\in S}$ represents the labels, and $x_i \in L$ is a category label at the site i, $L = \{1, \ldots, C\}$. Furthermore, Z is the partition function for normalization, S a set of sites, and N_i a set of neighbors of the site i. The neighborhoods should be chosen in such a way that the observations may support or contradict the labeling of neighboring sites. The sites provide observations $\boldsymbol{y}$, and the task is to derive the class labels $\boldsymbol{x}$. The two functions A_i and I_{ij} are called 'unary potential' and 'pairwise potential', respectively. They model the relationship between the observations $\boldsymbol{y}$ and the labels. The value of A_i and I_{ij} should be large in case the observation supports the label x_i or the label pair (x_i, x_j). We denote the unknown CRF model parameters by $\boldsymbol{\theta} = \{\boldsymbol{w}, \boldsymbol{v}\}$, the parameters $\boldsymbol{w}$ specifying the classifier for individual sites, the parameters $\boldsymbol{v}$ specifying the classifier for site neighborhoods. They need to be learned from training data.

The unary potential A_i represents relationships between labels and local features. It predicts the label x_i based on the local features at the site i. Various local features are useful to characterize site i. For instance, the CRF (Shotton *et al.* 2006) uses shape-texture, color, and location features. The pairwise potential I_{ij} represents relationships between labels of neighboring sites. It models compatibility between neighboring labels. If neighboring sites have similar features, I_{ij} suggests the same category label for them. If the sites have dissimilar features, they might be assigned different category labels. Both A_i and I_{ij} can be modeled as arbitrary unary and pairwise classifiers (Kumar and Hebert 2004). Features extracted from each site are used as observations in the CRF.

3.1.1. Unary potential

The unary potential A_i independently predicts the label x_i based on the observations $\boldsymbol{y}$: $A_i(x_i, \boldsymbol{y}) = \log P(x_i|\boldsymbol{y})$. The label distribution $P(x_i|\boldsymbol{y})$ is calculated by a classifier. We employ the *multiple logistic regression model*, $P(x_i = c|u_c(\boldsymbol{y})) = \exp(u_c(\boldsymbol{y}))/\sum_c \exp(u_c(\boldsymbol{y}))$, where $u_c(\boldsymbol{y}) = \boldsymbol{w_c}^{\mathrm{T}}\boldsymbol{h}_i(\boldsymbol{y})$, with $\boldsymbol{w}_c = [w_0, w_1, \ldots, w_M]$ $M + 1$ unknown parameters per class, and $\boldsymbol{h}_i = [1, h_1, \ldots, h_m, \ldots, h_M]^{\mathrm{T}}$ containing M features, h_m depending on observations $\boldsymbol{y}$. $\boldsymbol{w} = \{\boldsymbol{w}_c\}_{c=1,\ldots,C}$ are the model parameters. Consequently, the unary potential is

$$A_i(x_i, \boldsymbol{y}) = \sum_c \delta(x_i = c)\log P(x_i = c|u_c(\boldsymbol{y})) \tag{2}$$

3.1.2. Pairwise potential

The pairwise potentials I_{ij} describe category compatibility between neighboring labels x_i and x_j, which here take the form of a contrast-sensitive Potts model (Potts 1952):

$$I_{ij}(x_i, x_j, \boldsymbol{y}) = \boldsymbol{v}^T\boldsymbol{\mu}_{ij}(\boldsymbol{y})\delta(x_i \neq x_j) \tag{3}$$

where the function $\boldsymbol{\mu}_{ij}(\boldsymbol{y})$ is the pairwise relational vector for a site pair (i, j), and v are the model parameters. Note that in the case of object detection, the vector $\boldsymbol{\mu}_{ij}(\boldsymbol{y})$ encodes the pairwise features that are required for forcing geometric and possibly photometric consistency for the pair of parts.

3.2. Attribute grammars

Grammars have received increased attention in computer graphics, image interpretation, and reconstruction within the last years (Marvie *et al.* 2005, Müller *et al.* 2006, Zhu and Mumford 2006, Huang and Mayer 2007, Müller *et al.* 2007, Ripperda 2008, Han and Zhu 2009, Ripperda and Brenner 2009).

In the following, we present a concept that is based on attribute grammars (Knuth 1968, 1971, Abney 1997), which extend context-free grammars (Chomsky 1956, 1959). In contrast to other approaches such as shape or split grammars, they have specific advantages in making semantic assumptions explicit and formally specifying geometric constraints of aggregated objects (cf. Schmittwilken *et al.* 2009).

Table 1. Selected production rules (top), attributes and semantic rules (bottom).

No.	Production rule
P_1	`Building` → `Roof Facade Left Right Back`
P_2-P_4	`Facade` → `IFacade` | `LFacade` | `TFacade` | `UFacade`
P_5	`IFacade` → `FacadePart`
P_6	`LFacade` → `FacadePart FacadePart`
P_7	`TFacade` → `FacadePart FacadePart FacadePart`
$P_{8,}P_9$	`FacadePart` → `WindowFacadePart` | `EntranceFacadePart`
P_{10}	`WindowFacadePart` → `windowGrid facadeRectangle`
P_{11}	`EntranceFacadePart` → `Entrance windowGrid facadeRectangle`
$P_{12,}P_{13}$	`Entrance` → `stair door` | `door`
No.	**Semantic rule**
$R_1(P_9)$	`EntranceFacadePart.direction` = `FacadePart.direction`
$R_2(P_{11})$	`Entrance.direction` = `EntranceFacadePart.direction`
$R_3(P_{12})$	`stair.direction` = `Entrance.direction`
$R_4(P_{12})$	`door.direction` = `Entrance.direction`
$R_5(P_{12})$	`door.y` = `stair.y` + `stair.numberOfSteps` * `stair.treadDepth`
$R_6(P_{12})$	`door.z` = `stair.z` + `stair.numberOfSteps` * `stair.rise`
$R_7(P_{12})$	`door.width` = `stair.width`

Table 1 gives an excerpt of the grammar. Upper case initials indicate nonterminals and lower case initials indicate terminals. For production rules, | is used to separate concurrent production rules. For semantic rules, the '.' notation is used for the attributes of specific symbols: `Symbol.attribute`.

In general, production rules describe the relation between aggregated objects and their parts. Semantic rules and attributes specify the constraints that govern these aggregations. Together, they specify the partonomy of objects in an explicit and precise way. For instance, production rule P_{12} aggregates a `door` and a `stair` to an `entrance`. The semantic rules R_5 and R_6 constrain the relative position of both objects considering the stair's shape parameters.

Furthermore, attributes define parameters, for example, location parameters and shape parameters. Semantic rules then describe the propagation of attributes within the derivation, even between remote symbols. For instance, R_1–R_4 propagates the direction of the facade to the stair and the door. Hence, all these objects are parallel. Semantic rules are also used to constrain attribute values by thresholds or probability distribution functions.

Finally, we apply specific semantic rules that we call *guards* (Ueda 1986, Schmittwilken *et al.* 2007). Generally, production rules with the same nonterminal on the left-hand side represent different models. Guards support the selection of the best model with regard to the given data, see Section 4.3.

If we take facades as an example, L-, T-, or U-shaped buildings (production rules P_2–P_4) may be differentiated by the occurrence of different parallel and/or coplanar facade faces. L-shaped buildings for instance consist of two parallel principal planes, T- and U-shaped buildings of three parallel principal planes. Therefore, we introduce the terminal `facadeRectangle` that represents the contour rectangle of each part of the facade. The guards for L-, T-, and U-shaped buildings can be outlined as given in 4–6 where i denotes the index of the plane.

$$\texttt{facadeRectangle}_\texttt{i}\texttt{.geometry} = \left\{ \underbrace{a_i, b_i, c_i, d_i}_{\text{plane}}, \underbrace{x_i, y_i}_{\text{location}}, \underbrace{w_i, h_i}_{\text{shape}} \right\} \tag{4}$$

$$\texttt{facadeRectangle}_\texttt{i}\texttt{.precision} = p_i \tag{5}$$

$$\texttt{Facade.precision} = \sum_i P_i \; ; \tag{6}$$

The precision $p_i = 1/\sigma_i^2$ is related to the variance of the plane fitting. The guard calculates the Hessian normal parameters of the planes a_i, b_i, c_i, d_i and the location and shape parameters x_i, y_i, w_i, h_i. The best model is given by the rule that is given the highest precision by the guard.

4. Integration of low- and high-level reasoning for range data interpretation

4.1. Preparation

Three-dimensional data labeling is done mostly point based (Anguelov *et al.* 2005). For example, for every point to be labeled, a fixed number of neighboring points is randomly picked: three points are taken randomly in a fixed radius

sphere and another three points in a fixed radius cylinder. However, individually classifying neighboring points that belong to the same class is unnecessary and a large amount of computation is required for high-resolution data.

As we focus on building facades, planar patches are generated and then classified, instead of classifying individual points. In the following, we show how we generate planar patches. RANSAC is applied to extract planes. The basic idea is to estimate the model parameters using the minimum number of data possible and then to check which of the remaining data points fit the model estimated.

Based on the observation that RANSAC may find wrong planes if the data have a complex geometry, we use the following scheme for planar patch extraction: first, the point cloud is partitioned into small rectangular blocks to make sure that there will be a maximum of three planes in one block; second, RANSAC is applied to extract planes in each block; third, the minimum description length principle is used to decide how many planes are in each block (cf. Pan 1994). Eventually, there are zero to three planar patches in each block. Each planar patch is considered to be a site in the considered CRF model. The rectangular block serves as bounds to build neighboring features.

4.2. CRFs with priors

According to our CRF model, the interaction between part labels has to use observed data (e.g., the location of patches). Because in CRFs the pairwise potential I_{ij} is a function of the observed data, these fields allow for a way of solving the problem in a random-field framework. On the contrary, in conventional MRFs, the conditional distribution over labels is modeled as $P(\boldsymbol{x}|\boldsymbol{y}) \propto P(\boldsymbol{x},\boldsymbol{y}) = P(\boldsymbol{y}|\boldsymbol{x})\ P(\boldsymbol{x})$, where $P(\boldsymbol{x})$ is used for modeling the label interaction. Because $P(\boldsymbol{x})$ does not allow us to use data $\boldsymbol{y}$ while modeling label interactions, conventional forms of MRFs cannot model the geometric consistency simultaneously with appearance.

On the contrary, although CRFs explore neighborhood relations, they cannot model long-range interactions. For instance, if there are three windows on the facade, the relation between every pair of them can hardly be modeled by CRFs. The grammar model representing the semantic high-level structure serves as priors for the CRFs. These priors include different distributions for different classes and different distributions for object parameters. If there is no prior for a class, it is assumed to be uniformly distributed in the whole domain.

We compute two different types of feature vectors at each site i. First, a single-site feature vector $\boldsymbol{h}_i(\boldsymbol{y})$ is computed from the geometric property of the data $\boldsymbol{y}_i$ at the site. Obviously, this vector does not take into account the influence of the data in the neighborhood of that site. Next, $\boldsymbol{\mu}_{ij}(\boldsymbol{y})$ is calculated, which explicitly considers the dependencies in the data of neighboring sites. In the following, we describe the details of feature extraction at each site as follows:

(1) Range of point coordinates on the planar patch: Δx, Δy, Δz;
(2) Mean position of the planar patch: $\bar{x}, \bar{y}, \bar{z}$;
(3) Number of points on the planar patch;
(4) Angle between planar patch normal and Z-axis;
(5) Priors for `stair`, `door`, `window`, and `facade`;
(6) Angle between neighboring planar patch normals.

The parameters $\boldsymbol{\theta}$ of the CRF model are learned in a supervised manner. Hence, we use training data and the corresponding ground-truth labeling. We use the standard maximum likelihood approach and thus, in principle, aim at maximizing the conditional likelihood $P(\boldsymbol{x}|\boldsymbol{y},\boldsymbol{\theta})$ of the CRF model parameters. However, this would involve the evaluation of the partition function Z, which is in general NP-hard. To overcome this problem, one may either use sampling techniques or resort to some approximation, for example, mean-field or pseudo-likelihood, to estimate the parameters.

In this article, we use an alternative way to learn the parameters $\boldsymbol{\theta}$. We set $\boldsymbol{v}$ to 0 and put neighboring features into the single-site feature vector, which reduces CRF parameter learning to efficient logistic regression parameter learning. In future work, we plan to apply mean-field approximation to learn the CRF parameters $\boldsymbol{\theta}$. Therefore, we modify feature *angle between neighboring planar patch normals* to *mean angle within one block with respect to* Z*-axis*, and add one feature: the mean angle with neighboring blocks with respect to Z-axis. Hence, for each site i, a 15D feature vector $\boldsymbol{h}_i$ is obtained. We apply bounded logistic regression by solving logistic regression as a convex optimization problem with constraints (cf. Roscher and Förstner 2009). The 3D data are classified into five classes: `facade`, `window`, `door`, `stair`, and `unclassified`. It is important to select appropriate features that are capable of differentiating the different classes. For `stair`, `door`, `window`, and `facade`, we only use location priors here, that is, `window` and `facade` are assumed to be uniformly distributed over the whole blocks, and `stair` and `door` over a part of the blocks. Within one block, there is the same prior probability for the sites.

4.3. Parsing range data with attribute grammars

The modeling of 3D objects with attribute grammars was presented in a previous section. Now we discuss how the attribute grammar is used for the reconstruction of 3D objects. The planar patches that have been classified by the CRF are used as input data for the parsing algorithm.

Parsing range data with attribute grammars has its own intrinsic complexity. Parsing techniques from natural and formal language processing cannot be applied for several

reasons: (1) Words of formal languages are 1D and have a natural left to right ordering. However, there is no such ordering in 3D. (2) Formal language processing assumes that there is a set of symbols that is correct and complete. In 3D reconstruction, we start from noisy observations that are neither correct nor complete. To find the best model in the set of possible interpretations, an appropriate search strategy is needed.

Generally, parsing aims at finding the most likely derivation of a given grammar. The parsing algorithm presented is based on the exploration of an AND-OR tree (Nilsson 1980, Zhu and Mumford 2006). The latter supports the guided search of the most plausible derivation. Figure 2 shows the slightly trimmed AND-OR tree of the grammar defined in Table 1. The AND branches are connected by arcs. Two exemplary derivations of the grammar are highlighted. However, the parsing algorithm will only explore one of these derivations, namely, the most evident one.

Based on the type of the respective node, two different kinds of decisions have to be made for AND or OR nodes, both of which are introduced in the following:

(1) Starting at the root node of the tree, that is, the start symbol of the grammar, the algorithm selects at each **AND** node, the most evident one from the literals that are known so far. Because literals like `FacadePart`, `door`, and `window` correspond to different patches, the grouping of the latter gives an estimation which of the literals are best supported by the data. The selection of literals has no effect on the final models, but on the performance of the algorithm. Choosing a 'good' literal promises to start the reconstruction with the object that is strongly supported by the data and can therefore be reconstructed more accurately. At the same time, those objects imply constraints and derive parameters that can make the reconstruction of the remaining objects easier. The facade, for instance, will generally have the strongest support. Its derivation yields parameters for the orientation and location that simplify the identification of windows, entrances, and stairs.

(2) At each **OR** node, the parsing algorithm evaluates the guards of the applicable production rules to select the best production rule, that is, giving the highest precision. The latter is applied to the selected literal. The guards use low-level operators such as RANSAC to estimate the parameters of the derived symbols. The low-level operators include some semantic knowledge about the objects. First, the single parts of the objects are estimated and afterwards their precision is simply added to receive the total precision, for example, the precision of the facade and its parts in Equation (6). The guards consider the set of global constraints generated by all production rules that have been applied so far.

The pseudocode of the parsing algorithm `parse3d` is given in Table 2. The search strategy for exploring the AND-OR tree is similar to A* (Russel and Norvig 2003). The subprocedures `selectRule` (decision at OR nodes) and `selectLiteral` (decision at AND nodes) implement heuristic functions. These subprocedures are the sensitive parts of the algorithm. As explained above, they affect the efficiency of the search and the quality of the results. The selection of an inappropriate literal with weak support in the data makes the search for the best rule more difficult. The selection of an inadequate rule will yield the wrong model. To simplify the presentation, the AND-OR search algorithm assumes that an informed heuristic is available. This is the case whenever the evaluation of the guards is guaranteed to predict the best model. In this case, an irrevocable search strategy such as AO* (Nilsson 1980) is complete. Otherwise, a tentative search strategy has to be applied with either backtracking or parallel evaluation of different choices (Russel and Norvig 2003).

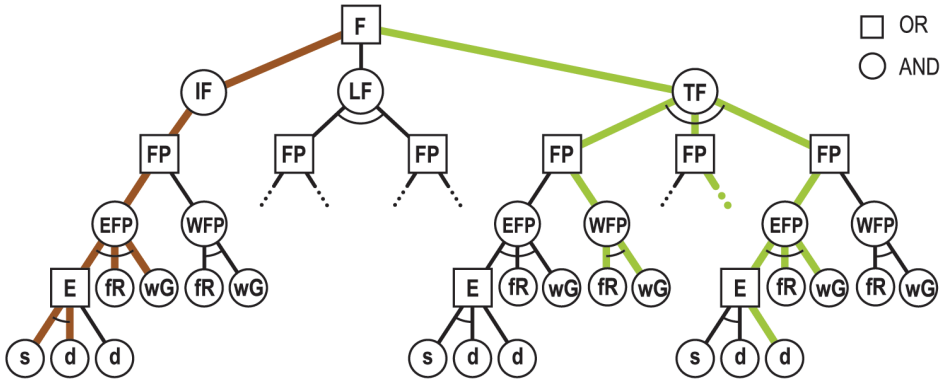

Figure 2. Slightly trimmed AND-OR tree of the grammar given in Table 1. The following acronyms are used: F: `Facade`, FP: `FacadePart`, IF: `IFacade`, LF: `LFacade`, TF: `TFacade`, fR: `facadeRectangle`, E: `Entrance`, wG: `windowGrid`, d: `door`, and s: `stair`. AND branches are connected by arcs.

Table 2. AND-OR search algorithm for the parsing of range data.

```
function parse3d (grammar, crfPatches) returns derivation
    inputs
        grammar      // set of production rules
        crfPatches   // classified and grouped 3D points
    output
        derivation   // the most evident derivation, i.e. set of
                     // terminal symbols
    local variables
        openLiterals ← {}       // non terminals that still have
                                // to be rewritten
        constraintStore ← {}    // set of globally valid constraints incl.
                                // derived parameters and their distributions
        terminals ← {}          // so far derived terminals of the grammar

    openLiterals ← openLiterals ∪ start symbol of the grammar
    while openLiterals is not empty do
        activeLiteral ← selectLiteral (openLiterals, crfPatches)
        [precision, rule constraints] ← selectRule(
              activeLiteral, grammar, crfPatches, constraintStore)
        openLiterals ← (openLiterals ∪ non-terminals in the body of rule) \
              activeLiteral
        terminals ← terminals ∪ terminals in the body of the rule
        constraintStore ← constrainStore ∪ constraints
    return derivation

function selectRule (literal, grammar, crfPatches, constraints) returns [p, r, c]
    output
        p   // precision estimated by the guard
        r   // the most evident production production rule
        c   // the constraints raised by the guard

    ask the guards of all applicable production rules for their precision
    return [precision, rule, constraints] of the strongest guard

function selectLiteral (literals, crfPatches) returns literal
    output
        literal   // the most evident literal

    return that literal with the highest support by the crfPatches
```


5. Results

In the following we give a short overview of the performance of the implementation of our concept. First, we demonstrate the classification of synthetic data with CRF and show the results of the application to real-world range data acquired from a terrestrial laser scanner afterwards.

The CRF model is trained with nine synthetic point clouds. In total the training samples consist of ~3.5 Mio. data points, and the total training time is around 3 hours with an Intel© Core 2 Duo 2.50 GHz CPU and 2 GB of RAM. We validate our algorithm with several synthetic data sets. Each data set is of a different size, but each point cloud is divided in nonoverlapping 32 × 32 rectangular blocks.

The synthetic data are derived from the attribute grammar by a random-based derivation – just the other way around than parsing. Because the derivation of a grammar gives the semantic structure, the point clouds can be labeled with their class attributes.

5.1. *Synthetic data*

We have tested our algorithm on several synthetic data sets that led to similar results. Therefore, we exemplarily present only one of the tested synthetic data sets (see Figure 3 top left).

The 0.4 Mio. original points are reduced to 2165 planar patches. Figure 3 bottom shows the classified planar patches of a window and the entrance. The plane normal vectors are also shown. Because of semitransparent rendering, light areas are caused by covered planar patches.

Figure 3 top right shows the classified points that inherit their classification from the planar patches. The computation time for feature extraction is around 7 minutes, and

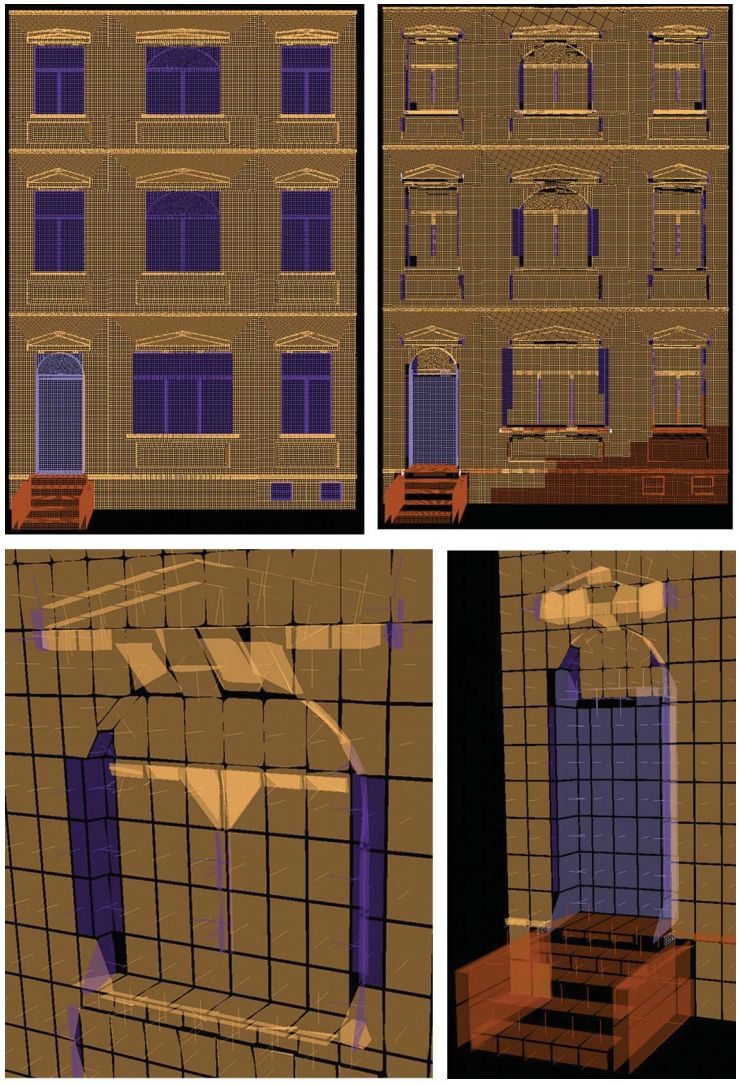

Figure 3. *Top Left*: Input point cloud of facade, with color-coded ground truth. *Top Right*: Point cloud with color-coded classification inherited from planar patches (CRF). *Bottom Left*: Zoom-in part of `window` planar patches from CRF classification. *Bottom Right*: Zoom-in part of `stair` and `door` planar patches from CRF classification. *Color-coding*: `facade`: light gray; `window`: dark gray; `door` and `stair`: medium gray; and `unclassified`: white.

CRF inference takes 0.06 seconds. The classification accuracy is around 67%.

5.2. *Real data*

We have applied our concept to real 3D range data of facades. We present some of the results of a data set of about 1.3 Mio. points (see Figure 4 top left). Low-level processing generated about 1600 planar patches.

Figure 4 top right shows the classified points. The color-coded classification is inherited from the planar patches that are derived from the CRF. The computation time for feature extraction is around 8 minutes and 0.02 seconds for CRF inference. Training using synthetic data leads to some misclassification because of the noisy real data. The CRF and the grammar have problems with the large ground plane because both the CRF and the grammar do not include the class `ground` yet.

We also applied the parsing algorithm to the same data set. Figure 4 bottom shows the stair estimated in front of the door and the facade. The four sample points used by RANSAC are visualized by spheres. The spheres have been used for the estimation of treads (horizontal parts) and risers (vertical parts). The rise parameter (height differences) has been estimated correctly: 17 cm. The estimation of tread depth was not well performed because there are still some points not classified as stair on the vertical parts of the steps. Because of the missing class `ground`, we estimated and eliminated the points of the ground plane before parsing. Because the ground is not a plane, some points remain and

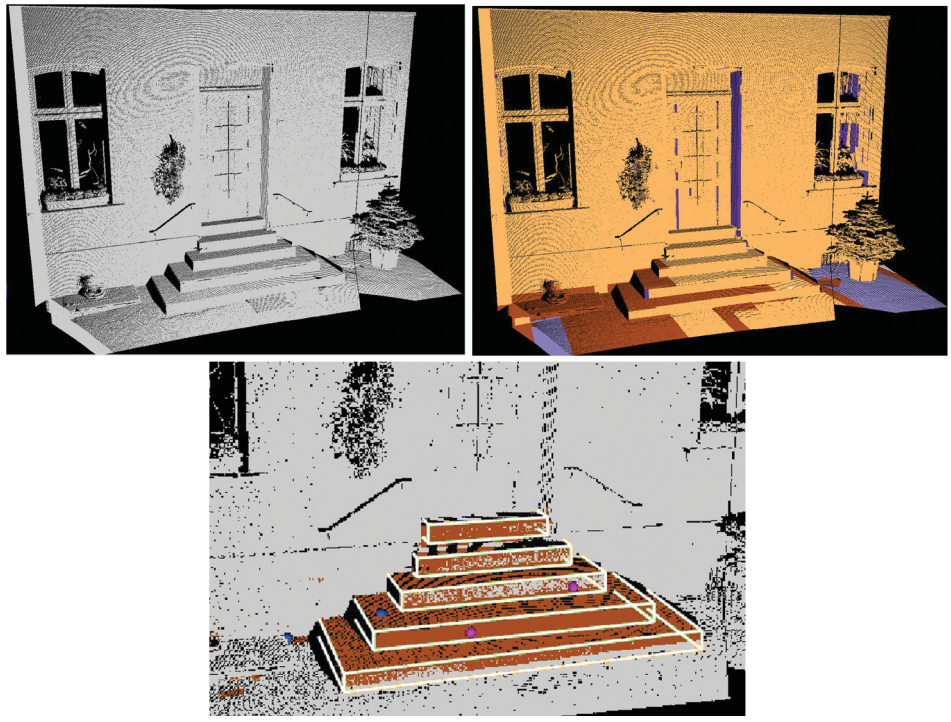

Figure 4. *Top left*: Unclassified point cloud of the facade. *Top right*: Point cloud with color-coded classification inherited from planar patches (CRF). *Bottom*: Result of the grammar-based parsing: skeleton of the reconstructed stair (white) superimposed on the improved classification of stair. *Color-coding*: `facade`: light gray; `window`: dark gray; `door` and `stair`: medium gray; and `unclassified`: white.

are falsely classified as `stair` (see dark gray points to the left of the stair). An improved guard could yield a better estimation of the width of the stair and therefore avoid such misclassification.

6. Conclusions

We have proposed a novel approach for the interpretation of 3D range data and the semantically meaningful reconstruction of man-made objects. It combines top-down and bottom-up reasoning by integrating CRFs with attribute grammars. Although MRFs have been used for image interpretation for more than a decade, they are limited to local features. This has been overcome by CRFs where arbitrary features can be used for classification. We use CRFs to classify planar polygonal patches derived by RANSAC. CRFs are close to the data. They represent the bottom-up part and start the reconstruction.

Attribute grammars specify meaning and structure of objects. Their specific advantage is the explicit representation of semantic assumptions and the precise specification of geometrical constraints. They translate semantics into observable features. Attribute grammars represent the top-down part of the reconstruction process.

The approach starts by grouping of 3D points into patches and classifying patches by CRFs. An uninterpreted point cloud is thus transformed into a set of meaningful symbols. From the grammars' perspective, this set of symbols, however, is neither complete nor (fully) correct in a logical sense. This makes parsing different from formal language processing. We use AND-OR trees to specify the parser and to control the search. The critical part of the search is the selection rule that is meant to predict the best model. We introduce the concept of guards to compare competing models and to select the one with the highest precision yielding an informed heuristic as part of an A* search strategy. More sophisticated control rules, however, are conceivable.

The methods developed so far allow further generalizations. The learning for the CRF, which up to now has been performed per class, will be generalized to a joint learning scheme, which is expected to yield better results. The grammar presented for stairs will be extended to the main facade elements such as windows, doors, and balconies. The guards will also be improved to obtain more accurate parameter estimation. The single pass from the data to the grammatical inference through the CRF will be extended to an iterative procedure where the posteriors of both methods are used as priors for the other. This will require the development of a random field containing a mixture of generative and discriminative model parts. Finally, the stochastic model of the attribute grammar will be enriched by adequate priors, which will be learned from annotated data sets.

Acknowledgments

The work is funded by Deutsche Forschungsgemeinschaft (German Research Foundation) FO 180/14-1 (PAK 274) within the Sino-German joint collaboration on 'Interpretation of 3D Urban Geoinformation'.

References

Abney, S., 1997. Stochastic attribute-value grammars. *Computational Linguistics*, 23, 597–618.

Anguelov, D., *et al*., 2005. Discriminative learning of Markov random fields for segmentation of 3D scan data. *In: Proceedings of IEEE computer society conference on computer vision and pattern recognition*, 169–176.

Brenner, C., Haala, N., and Fritsch, D., 2001. Towards fully automated 3D city model generation. *In*: *Proceedings of Third International Workshop on Automatic extraction of man-made objects from aerial and space images III*. Ascona, Switzerland, 47–56.

Chomsky, N., 1956. Three models for the description of language. *Information Theory, IEEE Transactions*, 2 (3), 113–124.

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

Dick, A., Torr, T., and Cipolla, R., 2004. Modelling and interpretation of architecture from several images. *International Journal of Computer Vision*, 60 (2), 111–134.

Fischer, A., Kolbe, T., and Lang, F., 1997. Integration of 2D and 3D reasoning for building reconstruction using a generic hierarchical model. *In*: W. Förstner and L. Plümer, eds. *SMATI '97, workshop on semantic modeling for the acquisition of topographic information from images and maps*. Basel, Switzerland: Birkhäuser-Verlag, 159–180.

Fischer, A., Kolbe, T., and Lang, F., 1999. On the use of geometric and semantic models for component-based building reconstruction. *In*: W. Förstner, C.E. Liedtke, and J. Bückner, eds. *SMATI '99, workshop on semantic modeling for the acquisition of topographic information from images and maps*. Institut für Theoretische Nachrichtentechnik und Informationsverarbeitung, Universität Hannover, 101–119.

Fischler, M. and Bolles, R., 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24, 381–395.

Gruen, A. and Wang, X., 1999. CyberCity Modeler, a tool for interactive 3-D city model generation. *Photogrammetric Week*, 99, 1–11.

Guelch, E., 2001. New features in semi-automatic building extraction. *In*: *Proceedings of ASPRS 2001 conference*, St. Louis, MO.

Han, F. and Zhu, S.C., 2009. Bottom-up/top-down image parsing with attribute grammar. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31 (1), 59–73.

Heinrichs, M., Hellwich, O., and Rodehorst, V., 2008. Robust spatio-temporal feature tracking. *In*: W. Förstner and H. Mayer, eds. *ISPRS congress Beijing 2008, proceedings of commission III*, *International Society for Photogrammetry and Remote Sensing*, 51–56.

Huang, H. and Mayer, H., 2007. Extraction of the 3D branching structure of unfoliaged deciduous trees from image sequences. *Photogrammetrie - Fernerkundung - Geoinformation*, 6, 429–436.

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

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

Kumar, S. and Hebert, M., 2003. Discriminative random fields: a discriminative framework for contextual interaction in classification. *In*: *Ninth IEEE Proceedings of international conference on computer vision*, 2, 1150–1157.

Kumar, S. and Hebert, M., 2004. Discriminative fields for modeling spatial dependencies in natural images. *In*: S. Thrun, L. Saul and B. Schölkopf, eds. *16th Annual conference on neural information processing systems*.

Lafferty, J., McCallum, A., and Pereira, F., 2001. Conditional random fields: probabilistic models for segmenting and labeling sequence data. *In*: *Proceedings of international conference on machine learning*, Montreal, Canada, 282–289.

Liang, P., Jordan, M.I., and Klein, D., 2009. Probabilistic grammars and hierarchical dirichlet processes. *In*: T. O'Hagan and M. West, eds. *The handbook of applied Bayesian analysis*. Oxford: Oxford University Press.

Marvie, J.E., Perret, J., and Bouatouch, K., 2005. The FL-system: a functional L-system for procedural geometric modeling. *The Visual Computer*, 21 (5), 329–339.

McCallum, A., Rohanimanesh, K., and Sutton, C., 2003. Dynamic conditional random fields for jointly labeling multiple sequences. *In*: *16th Annual conference on Neural Information Processing Systems workshop on syntax, semantics and statistic*.

Modestino, J.W. and Zhang, J., 1992. A Markov random field model-based approach to image interpretation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14 (6), 606–615.

Müller, P., *et al.*, 2006. Procedural modeling of buildings. *ACM Transactions on Graphics*, 25 (3), 614–623.

Müller, P., *et al.*, 2007. Image-based procedural modeling of facades. *ACM Transactions on Graphics*, 26 (3), 85.

Nilsson, N.J., 1980. *Artificial intelligence. Symbolic computation*. Berlin: Springer-Verlag.

Pan, H., 1994. Two-level global optimization for image segmentation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 49, 21–32.

Potts, R.B., 1952. Some generalized order-disorder transformations. *Proceedings of the Cambridge Philosophical Society*, 48, 106–109.

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

Ripperda, N. and Brenner, C., 2009. Application of a formal grammar to facade reconstruction in semiautomatic and automatic environments. *In*: *Proceedings of 12th AGILE conference on GISience*, Hannover, Germany, 2009.

Roscher, R. and Förstner, W., 2009. Multiclass bounded logistic regression–efficient regularization with interior point method. *In*: *Technical report*, Department of Photogrammetry, University of Bonn, 1–10.

Russel, S. and Norvig, P., 2003. *Artificial intelligence – a modern approach*. Upper Saddle River, NJ: Pearson Education.

Schmittwilken, J., Dörschlag, D., and Plümer, L., 2009. Attribute grammar for 3D city models. *In*: A. Krek, *et al*., eds. *Urban and regional data management*. Boca Raton: CRC Press, 49–58.

Schmittwilken, J., *et al.*, 2007. A semantic model of stairs in building collars. *Photogrammetrie, Fernerkundung, Geoinformation*, 2007 (6), 415–427.

Shotton, J., *et al*., 2006. Textonboost: joint appearance, shape and context modeling for multi-class object recognition and segmentation. *In*: *Proceedings of ninth European conference on computer vision*, Graz, Austria, 1–15.

Ueda, K., 1986. *Lecture Notes in Computer Science*, Vol. 221/1986. *In*: *Guarded horn clauses*. Berlin/Heidelberg: Springer, 168–179.

Zhu, S.C. and Mumford, D., 2006. A stochastic grammar of images. *Foundations and Trends in Computer Graphics and Vision*, 2, 259–362.