DATA DRIVEN RULE PROPOSAL FOR GRAMMAR BASED FACADE RECONSTRUCTION

Nora Ripperda and Claus Brenner

Institute of Cartography and Geoinformatics
Leibniz University of Hannover
{nora.ripperda, claus.brenner}@ikg.uni-hannover.de

KEY WORDS: facade modelling, building extraction, Markov Chain

ABSTRACT:

Today the demands on 3d models are steadily growing. At the same time, the extraction of man-made objects from measurement data is quite traditional. Often, the processes are still point based, with the exception of a few systems, which allow to automatically fit simple primitives to measurement data. The need to be able to automatically transform object representations, for example, in order to generalize their geometry, enforces a structurally rich object description. Likewise, the trend towards more and more detailed representations requires to exploit structurally repetitive and symmetric patterns present in man-made objects, in order to make extraction cost-effective. In this paper, we address the extraction of building facades in terms of a structural description. We extend our former work on facade reconstruction, which is based on a formal grammar to derive a structural facade description in the form of a derivation tree and uses a stochastic process based on reversible jump Markov Chain Monte Carlo (rjMCMC) to guide the application of derivation steps during the construction of the tree. We use measurements to improve the control of the rjMCMC process. This data driven approach reduces the number of false proposals and therefore the execution time.

1 INTRODUCTION

1.1 Motivation

The extraction of man-made objects from sensor data has a long history in research (Baltsavias, 2004). Especially for the modelling of 3D buildings, numerous approaches have been reported, based on monoscopic, stereoscopic, multi-image, and laser scan techniques. While most of the effort has gone into sensor-specific extraction procedures, very little work has been done on the structural description of objects.

Modelling structure though is very important for downstream usability of the data, especially for the automatic derivation of coarser levels of detail from detailed models.

Representing structure is not only important for the later usability of the derived data, but also as a means to support the extraction process itself. A fixed set of structural patterns allows to span a certain subspace of all possible object patterns, thus forms the model required to interpret the scene. Patterns can also guide the measurement process. Especially for man-made structures such as building facades, a large number of regularity conditions hold, which can be introduced into the measurement process as constraints.

Our aim is to extract facade elements from image and range data automatically. This paper extends our former work on the grammar based extraction of facade descriptions (Ripperda and Brenner, 2006) in which the grammar guides the generation of possible facade layouts using a reversible jump Markov Chain Monte Carlo (rjMCMC) process to explore solution space. The rjMCMC algorithm is used for other applications e.g. image segmentation as well. Tu et al. (2005) integrated generative and discriminative methods for image parsing. We present a way to derive distributions of facade attributes like the position of windows. These distributions are used for the rule proposal to evade the large number of wrong proposals which where so far only based on general prior knowledge on facades.

1.2 Related Work

Grammars have been extensively used to model structures. For modelling plants, Lindenmayer systems were developed by Prusinkiewicz and Lindenmayer (1990). They have also been used for modelling streets and buildings (Parish and Müller, 2001; Marvie et al., 2005). But Lindenmayer systems are not necessarily appropriate for modelling buildings. Buildings differ in structure from plants and streets, in that they don't grow in free space and modelling is more a partition of space than a growth-like process.

For this reason, other types of grammars have been proposed for architectural objects. Stiny and Gips (1972) introduced shape grammars which operate on shapes directly. The rules replace patterns at a point marked by a special symbol. Mitchell (1990) describes how grammars are used in architecture. The derivation is usually done manually, which is why the grammars are not readily applicable for automatic modelling tools.

Alegre and Dallaert (2004) use a stochastic context free attribute grammar to reconstruct facades from image data by applying horizontal and vertical cuts.

Wonka et al. (2003) developed a method for automatic modelling which allows to reconstruct different kinds of buildings using one rule set. The approach is composed of a split grammar, a large set of rules, which divide the building into parts, and a control grammar, which guides the propagation and distribution of attributes. During construction, a stochastic process selects among all applicable rules.

Dick et al. (2004) introduce a method which generates building models from measured data, i.e. several images. This approach is also based on the rjMCMC method. In a stochastic process, 3D models with semantic information are built. Mayer and Reznik (2006) also use a MCMC method for the facade reconstruction from images.

2 FACADE RECONSTRUCTION USING A GRAMMAR AND MCMC

2.1 The facade grammar

A grammar is used to model facade structure. The facade is presented by the derivation tree of the word of the language of the grammar, which corresponds to the facade. The grammar is built in a way that the derivation describes a recursive partition of space. We obtain a partition from the application of a derivation rule of the split grammar. A derivation tree represents the overall facade partitioning. Each node of this tree corresponds to one of the symbols of the grammar. There are two kinds of symbols, nonterminals and terminals. The terminal symbols represent facade geometry and cannot be subdivided further. Geometrically, nonterminals do not represent facade geometry directly but serve as containers, which hold other objects, represented in the derivation tree by nonterminal or terminal children.

Some of the containers imply that their children have identical properties while others don't (see figure 1). SYMMETRICFACADE indicates symmetries in the facade and can be replaced by SYMMETRICFACADESIDE which represents the left side and the mirrored right side of the facade and an optional SYMMETRICFACADEMIDDLE. Implicitly, left and right side have the same content. In contrast, FACADE implies nothing about its children. In figure 1 on the right hand side a FACADE is subdivided in two PARTFACADEs, the upper and lower part, that have no similarities.

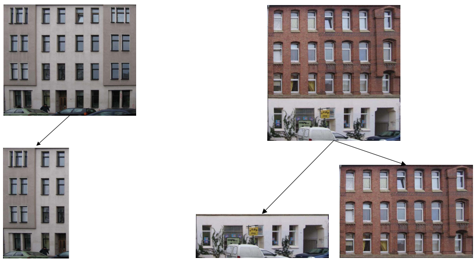

Figure 1: Symbols with (left: SYMMETRICFACADE) and without (right: FACADE) implications to their children.

The start symbol is the symbol FACADE. Starting from it, the subdivision can be made by rules similar to the ones introduced by Wonka et al. (2003). The model is expressed as a derivation tree with FACADE being the root. Derivation rules have a left side, which consists of one symbol, and a right side, which may comprise several symbols in a certain spatial layout. As an example, a grammar rule splits FACADE into GROUNDFLOOR and PARTFACADE. Figure 2 shows two examples of the subdivision of facades. In both cases the facade is subdivided into GROUNDFLOOR and the upper floors represented by PARTFACADE. The GROUNDFLOOR is partitioned in different FACADEELEMENTS that contain a DOOR or a WINDOW each. The upper floors are modelled in different ways. In the first case it is a SYMMETRICPARTFACADE with an IDENTICALFACADEARRAY of WINDOWs inside. In the second case two different IDENTICALFACADEARRAYs with different types of WINDOWs are derived.

The model is described by a parameter vector θ which contains the derivation tree and the attributes of the symbols. E.g. the parameter vector of the configuration in figure 1 right is represented by the hierarchic structure

$$\theta = Facade(0, 0, w, h, (PartFacade(0, 0, w, h_s),$$

$$PartFacade(0, h_c, w, h - h_s))),$$

where w and h are the width and hight of the facade and h_s is the height of the split.

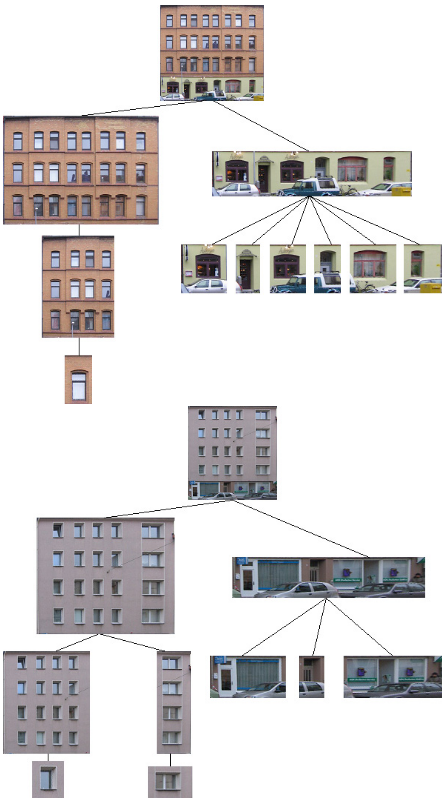

Figure 2: Example subdivision of facades.

2.2 Exploration of the Derivation Tree Using RjMCMC

We obtain the model of the facade using a stochastic process. We are searching for the model given by parameter vector θ with the highest probability $p(\theta|D_S D_I)$ under given scan (D_s) and image data (D_I) where the parameter vector θ encodes the current state of the derivation tree, including attributes.

We use a Markov Chain simulation to obtain the value of θ. This simulates a random walk in the space of θ. The process is led by a transition kernel $J(\theta_t|\theta_{t-1})$ and converges to a stationary distribution $p(\theta|D_S D_I)$.

The transition kernel $J(\theta_t|\theta_{t-1})$ assigns a probability to each rule and is made up from the commonness of the result in a dataset of facade images and some functions of the processed facade which will be described later. With the transition kernel in each iteration a rule is proposed. This is accepted with the acceptance probability

$$\alpha = \min(1, \frac{p(\theta_t|D_S D_I) \cdot J(\theta_{t-1}|\theta_t)}{p(\theta_{t-1}|D_S D_I) \cdot J(\theta_t|\theta_{t-1})}). \qquad (1)$$

This depends on the unknown distribution $p(\theta_t|D_S D_I)$. Using Bayes' law, this is proportional to $p(D_S D_I|\theta_t) \cdot p(\theta_t)$, a product of likelihood and prior of the facade.The acceptance probability decides whether the rule is applied or not.

During the simulation, facade elements are added, deleted or changed. The first two operations change the number of elements on the facade and thus the dimension of the parameter vector θ.

The basic Markov Chain Monte Carlo method does not support dimension changes of θ and therefore we use rjMCMC instead. This method allows a change in the dimension of the parameter vector θ and thereby the number of facade elements can vary during the simulation. The rjMCMC method requires reversibility. For each change from state θ_1 to state θ_2 there must exist a reverse change from θ_2 to θ_1.

2.3 Jumping Distribution

A change is proposed depending on the jumping distribution $J_t(\theta_t|\theta_{t-1})$ that expresses the likelihood for each change. Each state change is in one of the following categories:

- Application of a split rule from the grammar. Facade elements are divided horizontally, vertically or in both directions and each part becomes a new symbol.The split indicates a change in the facade. If the ground floor differs from the rest of the facade, a split is applied.

  In fact, one grammar rule comprises a set of changes to the parameter vector θ, since the associated attributes have to be chosen, such as the number and size of children. For example a rule divides FACADE into several PARTFACADEs, the general rule stands for all rules of this kind with any number and position of columns. The number of columns and their width is determined randomly.

- Changes in structure. Even after derivation of new containers according to the previous step, a second set of state changes allows to modify parameters, e.g. the number of rows or the position of the parting lines between rows.The same can be done starting from a child symbol. The position and extent of a symbol may change. In this case, the neighbour symbols, which are involved in the change, have to be changed as well.

- Replacement of symbols. This allows to interchange one symbol in the derivation tree by another symbol. In this case, the geometry stays the same, but the denotation changes. This is for example used if a FACADE is declared symmetric. FACADE → SYMMETRICFACADE

To ensure reversibility, each change can be applied from left to right and vice versa. This is a difference to the way split grammars are used, but is a requirement for the rjMCMC approach.

We have to define two kinds of distributions. The first one is the probability to choose a rule and the second one defines the parameter like the position of a split line or the number of windows. At the moment, the probability for rules is assigned manually depending on an assumed likelihood of the result. For example, a change FACADE → IDENTICALFACADEARRAY is more likely than FACADE → FACADEARRAY because facades build regular structures of similar elements. Some hints for the assumptions are taken from a database of facade images from Hannover.

To determine the parameter for the rules we need information about the distribution of colour or depth on the facade to control the split operation and to determine the distribution of the windows. Both depend on regularities and differences. For window grids we use autocorrelation and for splits a function based on a norm.

For splitting the facade into parts a change in colour or depth on a large part of the facade is needed. Other indications are breaks in regularity. The changes of colour and depth occur in different scales. We search for changes, which influence a great part of the facade, or separately changes caused by windows. Smaller artefacts in the facade may disturb the result. So we have different ways to score splits but in each we have to mask the small changes, which falsify the result. One way to suppress such unwanted changes is to use a scale space image (see figure 3). Another possibility is to cluster the facade depending on the colour value and in another step depending on the depth value. The results are shown in figure 4. From these images we can derive a probability for the splits. Therefore we compute the norm of two regions next to the split line (see figure 5), the upper region R_u and the lower region R_l. To evaluate the split line we compute the norm of the difference of both regions

$$||R_u - R_l||_2 = \sqrt{\sum_{x,y}(R_u(x,y) - R_l(x,y))^2},$$

where $R_u(x,y)$ is the rgb value at position (x,y).

The results are shown in figure 6. For a better visual understanding the original facade image is overlaied to the resulting graph. With the cluster image (blue line) we achieve better results than with the scaled image (red line) because on the scale image lines at top edges of windows are scored better than colour changes throughout the entire facade.

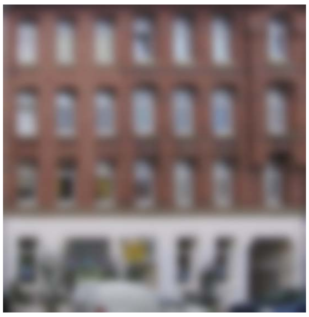

Figure 3: Image with lower scale maintains only large changes in facade structure.

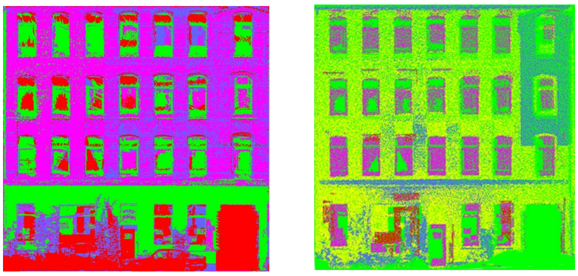

Figure 4: Clustered facade calculated by colour value and depth.

Using autocorrelation, we can predict the distribution of windows. We correlate the overlapping parts of the facade image and a copy of it which we shift horizontally resp. vertically. Figure 7 shows the result. In the case of a regular window grid the correlation values show peaks in a regular distance. The number of peaks is the number of window rows resp. columns plus one for the identical image plus one for the case when the overlap tends towards zero. In the example the horizontal correlation shows seven peaks because of the seven window columns plus two for identical and border cases. This pattern is not so clear for the vertical correlation because of the different ground floor.

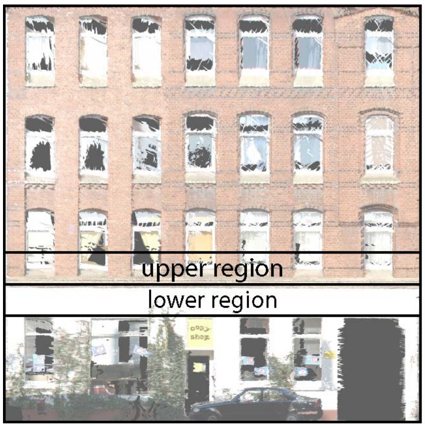

Figure 5: Two regions above and below the tested split line were moved over the facade.

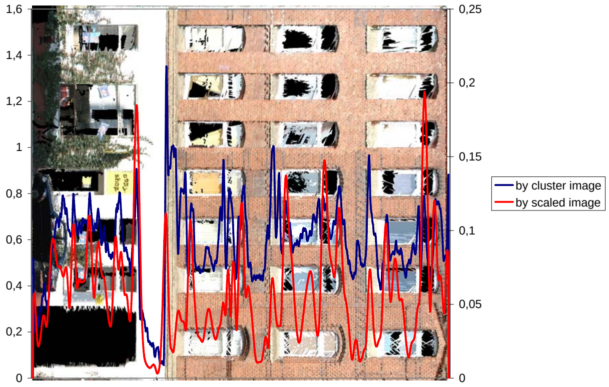

Figure 6: Facade image overlayed with the probability of splits evaluated by a scaled image and cluster image.

Another operation we can use to determine the window distribution is the planar segmentation of the scan. We use the segmentation descibed in (Dold and Brenner, 2004). Figure 8 shows the detected facade planes, but also sone smaller planes detected in the windows.

More information about the windows is given by point clouds from different standpoints. The laser beam penetrates the glass partly and is reflected from inside the building. If we compare two point clouds from different standpoints the differences mean windows or points, which can be seen only from one standpoint (see figure 9). The latter should not occur if we limit the point cloud to the facade.

To determine the differences we need the registration of the point clouds. This is the transformation matrix from the coordinate system of one standpoint to the one of another. We transform one point cloud in the coordinate system of the other and transform the cartesic coordinates into polar coordinates. The point cloud of one standpoint is stored as a raster addressed by polar and azimuth angle. Therefore with the received polar and azimuth angle the corresponding scan point can be read. A difference in the range value means a different point and therefore a window hypothesis. In figure 10 white pixel mean window hypothesis, black pixels have no corresponding pixel in the second scan and grey pixels are others.

2.4 Scoring Functions

The scoring functions affect the acceptance probability (eq. 1) in the term $p(D_S D_I|\theta_t) \cdot p(\theta_t)$ respectively $p(D_S D_I|\theta_{t-1}) \cdot p(\theta_{t-1})$.

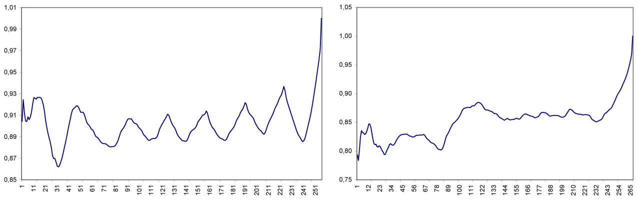

Figure 7: Autocorrelation coefficent in horizontal and vertical direction for the facade in figure 5.

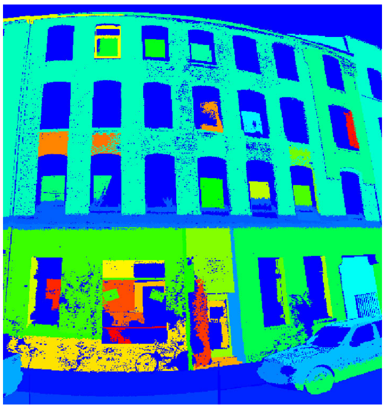

Figure 8: Segmentation of the scan leads to different planes for facade and windows.

For the evaluation we use different methods, which can be divided into two groups. The first group contains methods, which test the general plausibility of the model of the facade corresponding to the factor $p(\theta_t)$. They depend on the alignment, the extent and the position of the facade elements. Here we use the same scoring functions as given in (Dick et al., 2004), which where described in (Ripperda and Brenner, 2006) as well.

The second group evaluates how good the model fits the data by comparing it to range and image data corresponding to the first term $p(D_S D_I|\theta_t)$. In any case, the evaluation functions return a score, which builds an acceptance probability for the change. To determine $p(D_S D_I|\theta_t)$ we have different possibilities which use scan and image data. We develope measures for depth and colour and use correlation, entropy and variance as well.

Depth In the first case, the fact that window points typically lie behind the facade is exploited. The average $\bar{d}$ of the facade depth is calculated. The variation of the points inside the proposed window constitutes the measure

$$\alpha_d = \frac{\sum |d - \bar{d}|}{A},$$

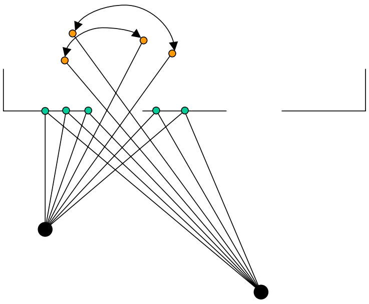

Figure 9: Principal sketch for window hypothesis.

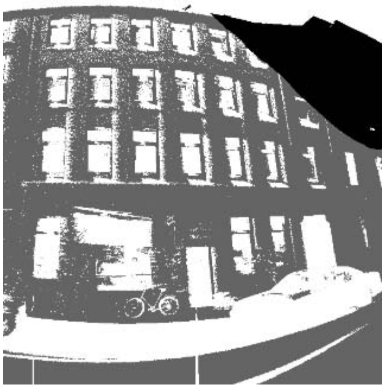

Figure 10: Window hypothesis from different standpoints.

where A is the total number of points. α_d is typically close to zero for facade points and large for window points.

Colour In the second case, colour has been used since windows typically appear darker than the surrounding facade (or in some cases brighter because of reflections). Here we use the clustered images as well. We consider one region for the window and a boundary region (see figure 12 left). Let N_{max} be the number of pixels of the largest cluster inside the proposed window region, N_0 the number of unclassified pixels, A_{win} the area of the window, A_{bound} the area of the boundary and N_{bound} the number of pixels of the boundary which belong to the largest cluster inside the window. α_C gives a measure for the window.

$$\alpha_C = \frac{1 + \frac{N_{max}+N_0}{A_{win}} - \frac{N_{bound}}{A_{bound}}}{2}$$

In colour and depth cases, the information is used for the subdivision of the facade. A proposed split of a container demands that the children have different properties.

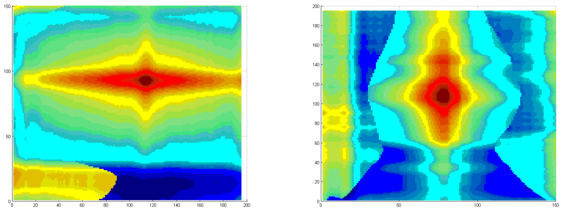

Figure 11: Score function with the depth cluster method (left) and color cluster method (right).

Correlation In the case of similarity we use the correlation function (see sec. 2.3). For example, upon division into rows, the resulting row strips are correlated to determine whether the split is accepted or not.

Entropy To score arrays of windows we used entropy and variance for homogeneity measure. Entropy is

$$I = \sum_{i=1}^{n} \frac{|C_i|}{A} \log_2 \frac{A}{|C_i|},$$

where n is the number of clusters, A the total area and $|C_i|$ the number of points in the i-th cluster. We divide the facade with a mask like in figure 12, right, according to the proposed array of windows. Entropy respectivly variance are calculated for white and gray areas separately.

We test the entropy for different grid positions. The grid has six degrees of freedom but for a better visualisation we fix the number of grid points and the distance between them. The results are shown in figure 13 and 14 on the left hand side. In both figures the yellow surface is the score of the window part of the facade, the blue one of the boundary part. In the diagramm for entropy the boundary part produces the better result because the window part isn't as homogeneous as the facade without windows. To obtain a probability we use the boundary part. The maximum possible result is $\log_2 n$ so we normalize the function with this factor. The probability (see figure 13, right) is $\alpha_I = 1 - \frac{I}{\log_2 n}$.

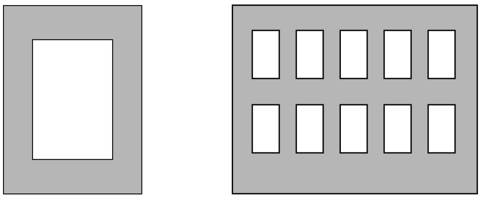

Figure 12: Mask for a single window (left) and an array of windows (right). The window area is white and the boundary area gray.

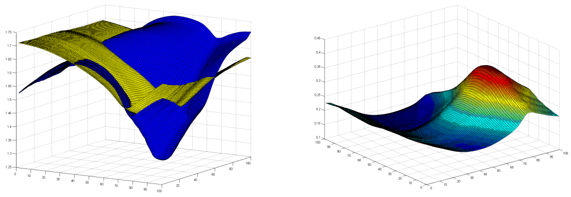

Figure 13: Entropy of window (blue) and boundary (yellow) and the probability derived from the window entropy.

Variance For another homogeneity measure, the variance, we use the original facade image because cluster labels are artificial numbers which would weight the differences arbitrarily. With this measure the boundary part of the facade leads to good results while the variance of the window part is higher than the one of the boundary part or mixed parts. Using $\alpha_V = 1 - \frac{\sqrt{V}}{255}$ we get a probability (see figure 14, right).

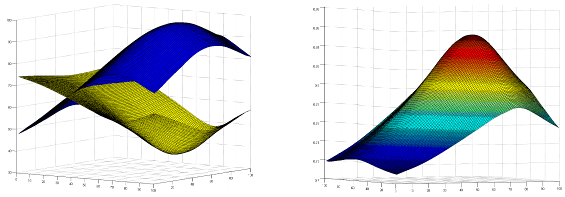

Figure 14: Variance of window (blue) and boundary (yellow) and the probability derived from the boundary variance.

3 RESULTS

We've tested the method on facades of dwelling houses. The input data are the point cloud and an orthophoto, which is generated with the RiScanPro software. The other required data are computed in a first step.

For a better understanding we first test parts of the modelling process separately. Therefore we cut out a single window. For this small data set we compute the score for each value and compare the result of the MCMC process (see figure 15) with the distribution given by the score function (see figure 11).

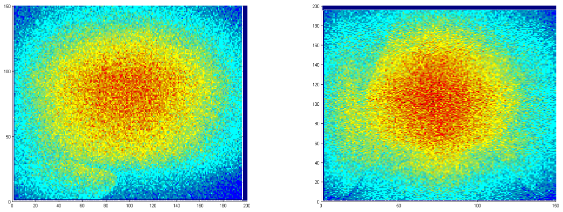

Figure 15: Sampled points with the depth cluster method (left) and color cluster method (right).

In the complete process, windows may not be modelled at the correct position in early derivations. Figure 16 shows two interim results of facade models. Not all modelled windows fit to the real ones. The reason is the assumption that the windows are arranged in a regular grid, which is not true. After further derivation steps the facade part is subdivided and single parts contain a regular grid. It is also possible that the grid pattern changes. Figure 17 shows a model of the right facade from figure 16. This final model uses a grid of window pairs and reproduces the facade in a better way.

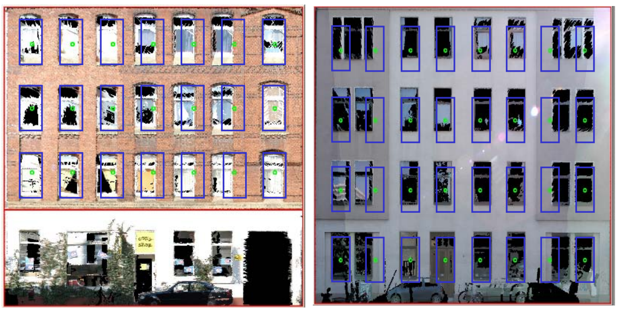

Figure 16: Displaced windows because of the wrong assumption of a regular grid.

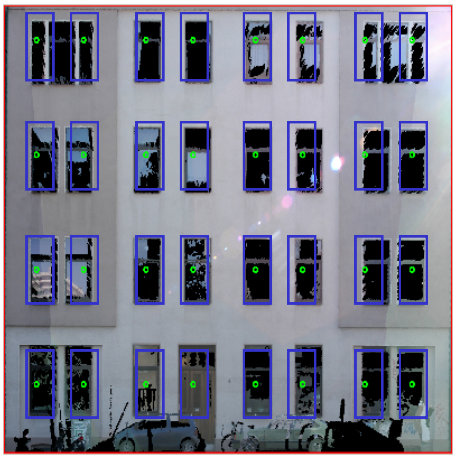

Figure 17: Facade model with a grid of window pairs.

4 CONCLUSION AND OUTLOOK

In this paper, we have presented an advancement of our previous work on grammar based facade reconstruction. It also combines the generation of artificial facade structures using grammars, and the reconstruction of facades using rjMCMC. Compared to existing grammar-based approaches, we gain the ability to reconstruct facades based on measurement data. Compared to existing rjMCMC approaches, by using a grammar, we obtain a hierarchical facade description and the ability to evaluate superstructures such as regularity and symmetry at an early stage, i.e., before terminal symbols such as WINDOW are instantiated.

We presented several measures to improve the rule proposals. These are no longer based only on general prior knowledge of facades. The measured facade influences the process not only in the scoring part but also in the proposal part.

Acknowledgements This work was done within in the scope of the junior research group “Automatic methods for the fusion, reduction and consistent combination of complex, heterogeneous geoinformation”, funded by the VolkswagenStiftung, Germany.

References

Alegre, F. and Dallaert, F., 2004. A probabilistic approach to the semantic interpretation of building facades. In: International Workshop on Vision Techniques Applied to the Rehabilitation of City Centers.

Baltsavias, E. P., 2004. Object extraction and revision by image analysis using existing geodata and knowledge: current status and steps towards operational systems. ISPRS Journal of Photogrammetry and Remote Sensing 58, pp. 129–151.

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

Dold, C. and Brenner, C., 2004. Automatic matching of terrestrial scan data as a basis for the generation of detailed 3D city models. In: International Archives of Photogrammetry and Remote Sensing, Vol. XXXV, Part B3, Proceedings of the ISPRS working group III/6, Istanbul, pp. 1091–1096.

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), pp. 329 – 339.

Mayer, H. and Reznik, S., 2006. Building facade interpretation from uncalibrated wide-baseline image sequences. ISPRS Journal of Photogrammetry & Remote Sensing 61, pp. 371–380.

Mitchell, W. J., 1990. The Logic of Architecture : Design, Computation, and Cognition. Cambridge, Mass.: The MIT Press.

Parish, Y. and Müller, P., 2001. Procedural modeling of cities. In: E. Fiume (ed.), ACM SIGGRAPH, ACM Press.

Prusinkiewicz, P. and Lindenmayer, A., 1990. The algorithmic beauty of plants. New York, NY: Springer.

Ripperda, N. and Brenner, C., 2006. Reconstruction of faade structures using a formal grammar and rjmcmc. In: K. Franke, K.-R. Mller, B. Nickolay and R. Schfer (eds), Pattern Recognition, Proceedings of the 28th DAGM Symposium, pp. 750–759.

Stiny, G. and Gips, J., 1972. Shape Grammars and the Generative Specification of Painting and Sculpture. Auerbach, Philadelphia, pp. 125–135.

Tu, Z., Chen, X., Yuille, A. and Zhu, S., 2005. Image parsing: Unifying segmentation, detection, and recognition. International Journal of Computer Vision 63(2), pp. 113–140.

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