GENERATIVE MODELS AND MARKOV CHAIN MONTE CARLO TECHNIQUES FOR DETECTION AND RECONSTRUCTION OF STAIRS FROM POINT CLOUDS

Jens Saatkamp, Jörg Schmittwilken

Institute for Geodesy and Geoinformation, University of Bonn
saatkamp@ipb.uni-bonn.de, schmittwilken@ikg.uni-bonn.de

KEY WORDS: Building, Stairs, Detection, Reconstruction, Markov Chain Monte Carlo, MAP Estimation, Generative Model, Point Cloud

ABSTRACT:

The paper describes an approach for the automatical reconstruction of homogeneous straight stairs from point cloud data by using a generative model and Markov Chain Monte Carlo techniques for estimating the parameters. Parameters for a generative model for stairs are presented. The six parameters of this 2D model are determined with a maximum-a-posteriori estimation approach. For all parameters prior probability distributions are chosen. Two types of likelihood functions are introduced. It is shown that four of the parameters under certain conditions can be determined via MCMC. Some results are presented.

1 INTRODUCTION

Task and Motivation Our interest is the semi-automatic acquisition of 3D-data on the transition between building facades and the digital elevation model, the so called building collar. The goal of this research is to increase automation for simplifying and easing the work of the human operator gradually when generating 3D city models. The models to be handled become more and more detailed and require the gradual inclusion of building parts like windows, balconies, doors or stairs.

In this paper we concentrate on stairs being an important link between the building and its surroundings. Stairs on one hand show a rich structure, e. g. may consist of several parts linked by platforms or even may split or merge, which requires a deep modelling and on the other hand contain simple enough parts, like straight flights of stairs, which can be expected to be detected and reconstructed fully automatically.

We present methods for the detection and reconstruction from point clouds. In a first step we restrict to 2D-stair-profiles in order to explore the feasibility of Monte Carlo techniques for this type of problem. In future we will extend the model to handle 3D points of stairs and other objects in the building collar.

Approach for solution Generative models allow to explicitly represent the internal structure of complex objects together with the uncertainty of the model itself as well as the uncertain observation process. They allow to generate simulated data and to evaluate given data with respect to the model. Our stair model is a parametric one, with mixed parameters, thus a mixture of discrete and continuous parameters.

We use maximum-a-posteriori estimation to solve the problem of finding optimal stair parameters. The a-priori-information of the parameters may initially be non-informative, but may be informative as soon as experimental evidence is available.

To find the maxima of the a-posteriori-distribution we use Markov Chain Monte Carlo (MCMC) techniques. Explicitly we use the Metropolis-Algorithm and the principle of Simulated Annealing.

Organization of paper The next section gives an overview about related work taken into account for our research. The general setup of the maximum a posteriori (MAP) estimation is described in Section 3. We give an overview of the parameters of the generative model and describe the distributions for the a-priori-information in detail for each parameter. We show which type of observations we used and how the likelihood function of our MAP-estimation is constructed. We describe the main aspects of the MCMC technique, our target distribution, the Metropolis Algorithm, the Updating Order, principles of Simulated Annealing and the proposal distributions. We also explain some details necessary to understand the implementation. In Section 5 we present the actual results of our approach with 2D data and give a conclusion in the end.

2 RELATED WORK

The approach of modelling objects with a generative model and the determination of model parameters with MCMC techniques was first described by Radford M. Neal in 1993 e.g. in (Neal, 1993).

An approach for architectural models is described in (Dick et al., 2004). Dick, Torr and Cipolla describe the modelling of architecture from images. Their model consists of walls and primitives. They give prior distributions for walls and primitives and a likelihood distribution. The final determination of the parameters is done by reversible jump MCMC (rjMCMC).

Mayer and Reznik (Mayer and Reznik, 2005), (Mayer and Reznik, 2006) describe the detection of windows in building facades with implicit shape models, plane sweeping and rjMCMC technique.

Brenner and Ripperda (Brenner and Ripperda, 2006), (Ripperda and Brenner, 2006) introduce an approach to describe building facades with the help of a facade grammar. A facade is represented by a derivation tree which is generated by rjMCMC.

Literature about MCMC methods can be found in (Gilks et al., 1996) and (Andrieu et al., 2003). A theoretical overview about Markov Chains is given in (Brémaud, 1999).

3 MAP-ESTIMATION FOR DETERMINING THE PARAMETERS OF THE GENERATIVE MODEL FOR STAIRS

In this section we establish a posterior distribution for the determination of the optimal parameters.

3.1 General Setup

The posterior distribution

$$p(\boldsymbol{x}|\boldsymbol{d}) = \eta p(\boldsymbol{d}|\boldsymbol{x})\, p(\boldsymbol{x}) \tag{1}$$

represents the probability of the stair parameters in $\boldsymbol{x}$ given the measured 3D point data in $\boldsymbol{d}$.

The parameter $\eta = 1/p(\boldsymbol{d})$ normalizes the density function but is not needed in the following because we only need to determine a maximum. Modeling requires knowledge about the a priori probability density $p(\boldsymbol{x})$ of the unknown parameters and the likelihood $p(\boldsymbol{x}|\boldsymbol{d})$ reflecting the observation process.

We will describe the parameters $\boldsymbol{x}$ and their prior $p(\boldsymbol{x})$ and the observation $\boldsymbol{d}$ together with the likelihood in more detail.

Maximizing the a posteriori distribution (1) (MAP- estimation) is equivalent to minimizing the self information $I(\boldsymbol{x}|\boldsymbol{d}) = -\ln p(\boldsymbol{x}|\boldsymbol{d})$ being the negativ logarithm of the corresponding probability:

$$I(\boldsymbol{x}|\boldsymbol{d}) = I(\boldsymbol{d}|\boldsymbol{x}) + I(\boldsymbol{x}) - \ln \eta \tag{2}$$

This expression is often numerically easier to handle. Minimizing $I(\boldsymbol{x}|\boldsymbol{d})$ is actually equivalent to minimizing the description length of $\boldsymbol{x}$ given $\boldsymbol{d}$.

In order to open a choice of an optimiziation technique we use both, the probabilistic and the information theoretic representation.

3.2 Parameters and a-priori probabilities

3.2.1 Parameters A model for straight stairs in 3D is shown in figure 1. For a 3D model 9 parameters are needed. In the following we describe the parameters of a 2D model, where only 6 parameters are sufficient.

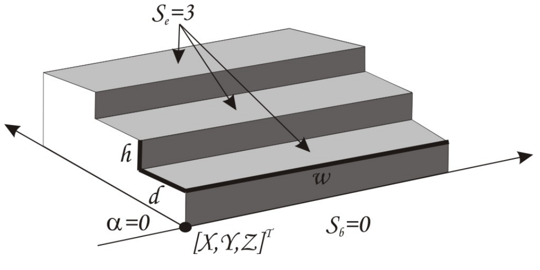

Figure 1: Generative model for a straight stair in 3D with 8 parameters: the number S of stairs, the 3D-position $[X, Y, Z]$ of the reference point, the tread depth t and the riser height h, the step width w and the azimuth α.

The profile of a straight homogeneous stair can be modeled using the following parameters x_i:

- The number $x_1 \doteq S_b \in \mathbb{N}$ of steps before the reference point and the number $x_2 \doteq S_e \in \mathbb{N}$ of steps after the reference point. We use two parameters here because we want to be able to determine the beginning and the end of the stairs independently just referring to the reference point.

- The 2D-coordinates $x_{3:4} \doteq [Y, Z]^\mathsf{T} \in \mathbb{R}^2$ of the reference point of the stairs. In 3D we will have an additional coordinate X.

- The tread depth and the slope $x_{5:6} \doteq [t, h]^\mathsf{T} \in \mathbb{R}^{+2}$, which is assumed to be the same for all steps. The two parameters t and h are assumed to be positive numbers. The direction of the stairs is always assumed to be in the upward direction. The slope h corresponds to the quotient of riser height and tread depth. It is the tangens of the gradient angle of the stairs. So the step height in our model results from the product of the tread depth and this slope.

- The ratio $x_7 = \varepsilon$ of outliers can also be seen as a parameter.

3D-steps in addition require a parameter for the step width. Aditionally we have a parameter α for the azimuth to allow for stairs to go in four principle directions referring to the facade. A stair can lead directly to the facade or one can walk parallel to the facade, respectively upstairs or downstairs.

3.2.2 A-priori Information For each of the parameter groups we might have prior information. This is useful in case the data are weak. Otherwise it should not have a signifcant influence on the estimate, thus will be overwritten by the data.

Though we have six parameters we only use $U = 5$ priors because only for the sum S of the steps S_a and S_e information is available. In this simple model of a projection of a homogeneous stair we may assume all $U = 5$ prior parameters to be independent, thus

$$p(\boldsymbol{x}) = \prod_{u=1}^{U} p(x_u) \tag{3}$$

In all cases we might distinguish three degrees:

1. No prior. Then the a-priori-probabilities are flat and we do not have a factor $p(x_u)$.

2. Weak prior, e. g. a mean value, a tolerance or a mean value and a variance.

3. Strong information, e. g. from reference data. Then in general we have some irregular distribution.

At the moment we use weak priors only.

Priors for the number of steps We model a prior for the parameter $S = S_a + S_b$, the sum of the two parameters described above, because the values for S_a and S_b are not unique for a particular stair. In case we only know the mean, a reasonable choice for the distribution is an exponential distribution:

$$P(S) = \frac{1}{\mu_S} \mathrm{e}^{-S/\mu_S} \qquad I(S) = \frac{\ln \mu_S}{\mu_S} S \tag{4}$$

Thus the prior information increases linearly with S. Stairs with less steps are preferred.

Prior for the position of the stairs The 2-vector specifying the position of the stairs requires a definition of the coordinate system.

In our application it appears to be useful to relate the stairs to the building. Therefore the origin of the stairs should refer to a point next to the facade. In case the stair direction is perpendicular to the facade the position of the origin will be close to the facade and approximately in the height of the surrounding terrain. However, the accuracy of this assumption is low.

In general assuming a mean and a standard deviation for each of the coordinates appears to be sufficient. This leads to the a-priori-model

$$p(Y,Z) \quad = \quad g(Y;\mu_X,\sigma_X^2)\; g(Z;\mu_Z,\sigma_Z^2) \qquad (5)$$

with

$$g(x;\mu_x,\sigma_x^2) = \frac{1}{\sqrt{2\pi}\sigma_x}\mathrm{e}^{-\frac{1}{2}\left(\frac{x-\mu_x}{\sigma_x}\right)^2} \qquad (6)$$

The prior information prefers stairs close to the mean $[\mu_Y,\mu_Z]$.

Prior for the tread depth and the riser height In a first approximation we assume the tread depth and the rise height to be normally distributed.

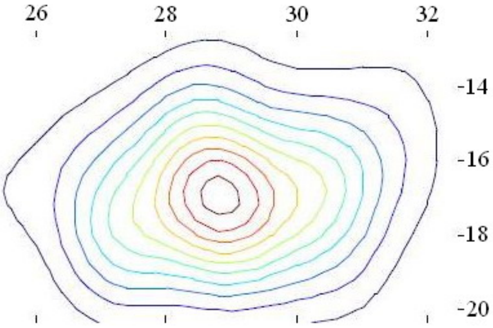

Figure 2: Contour lines of smoothed histogram of tread t (horizontal) and rise h (vertical): the two parameters of a step appear to be uncorrelated and may be approximated by a normal distribution. Contourlines taken from 117 samples.

Prior for the percentage of outliers The outlier rate ε is quite dependent on the observation process. In a first instance we fix this value.

Parameters of the prior distributions The parameters of the prior distributions are mostly taken from (Schmittwilken et al., 2007).

number S of the stairs:

$$\mu_S = 5.7$$
$$\sigma_S = 1.7$$

slope h:

$$\mu_h = 0.60$$
$$\sigma_h = 0.08$$

tread depth t:

$$\mu_d = 30.8$$
$$\sigma_d = 2.7$$

For the reference point there is no data available because its position depends on the coordinate system and is somehow arbitrary. We choose zero as expectation value. Furthermore we take quite high standard deviations because the starting positions may be far away from the real values of the parameters.

$$\mu_Y = 0$$
$$\sigma_Y = 200$$
$$\mu_Z = 0$$
$$\sigma_Z = 100$$

Of course these priors may be improved or adapted if necessary or in case more information is available.

3.3 Observations and Likelihood Function

3.3.1 Observations The initial observations are points $\boldsymbol{q}_i$, $i = 1,...,I$, derived by photogrammetric matching or by LIDAR. These data are characterized by two parameters:

1. Geometric accuracy of the points, represented by a covariance matrix.

2. The density of points.

Density and accuracy in general may depend on various parameters, e. g. distance of the sensor to the object, angle between surface and viewing direction, surface texture.

3.3.2 Likelihood function In order to derive a reasonable likelihood function we assume the following: The accuracy of the points is the same for all points and represented by $\sigma^2 I_2$. The density of the points does not vary too much. Reasons for variations may be manifold, like occlusions, large angles between normal and viewing direction, low texture. However, we require that the first and last step contain enough points so that support is not by points outside the stairs alone. We therefore take the number of points on the first and last step as observations and relate it to some robust average, say the median. These assumptions may be incorporated into the likelihood function.

We can derive the likelihood function

$$p(\boldsymbol{d}|\boldsymbol{x}) = p(n_1|\boldsymbol{x})p(n_S|\boldsymbol{x})\prod_{i=1}^{I} p(\boldsymbol{q}_i|\boldsymbol{x}) \qquad (7)$$

1. $p(\boldsymbol{q}_i|\boldsymbol{x})$ only depends on the distance e_i of a point $\boldsymbol{q}_i$ to the stairs. We assume each step to consist of a vertical and a horizontal segment with length h and t resp. The likelihood is mixture of Gaussians with mean 0 and standard deviation $\sigma_G = \sigma$ for the good points and $\sigma_B >> \sigma_G$ for the bad points

   $$p(\boldsymbol{q}_i|x) = p(e_i|\boldsymbol{x}) = (1-\varepsilon)g(e_i|0,\sigma_G^2) + \varepsilon g(e_i|0,\sigma_B) \qquad (8)$$

   In case we have knowledge about an anisotropic distribution of the measured point coordinates $\boldsymbol{q}_i$ we may introduce this, then the likelihood for each point in addition to its covariance matrix depends on the direction of the normal.

   The points which do not lie on the surface are classified as bad points. They are either outlier of the measurement process or do not belong to the stairs at all, e. g. to the house wall or the ground. For control purposes it is useful to link this standard deviation to the measuring accuracy σ by $\sigma_B = f_B\sigma_G$. For our tests we specified $f_B = 80$.

   The outlier rate ε is quite dependent on the observation process. Initially we fix this value.

   The structure of the likelihood function can be seen in figure 3.

2. n_1 and n_S are the numbers of points on the first and the last step. This requires to identify the step s each point **i** belongs to in order to obtain the numbers n_s, and especially n_1 and n_S. Only if the number of points on the first and the last step is high enough, can we assume these two steps to be the two ends of the stairs. This number should be related to a robust estimate of the number of points per step, e. g. the median of the number of steps and should be larger than this reference value. Thus we need a likelihood function depending on

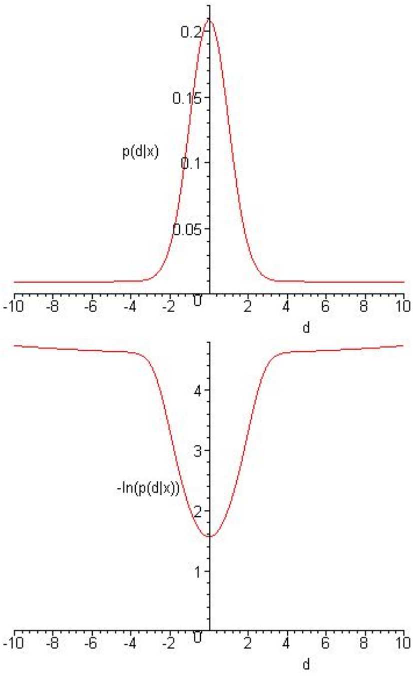

Figure 3: Likelihood function and negative log-likelihood function for point observations: $\sigma_1 = 1, \sigma_2 = 20, \varepsilon = 0.5$

$n_1/\text{med}(n_s)$ and $n_S/\text{med}(n_s)$, which is skew, supporting the values of this ratio that are larger than 1 and punishing values that are lower than 1.

This can be achieved with the the following distribution (cf. figure 4)

$$p\left(\frac{n_i}{\text{med}(n_s)}|\boldsymbol{x}\right) = \frac{3b}{\pi}\frac{x^{10}}{b^{12}+x^{12}} \quad i = 1, S \qquad (9)$$

The parameter b is to be so chosen that low ratios yields low support and the probability that the ratio is larger than 1 is high. In case $b \approx 0.815$ the ratio $n_i/\text{med}(n_s)$ is larger than 1 with probability 75 %. The probability that half this ratio is smaller than 1/2 is only 1.4 %, that it is larger than 2 is still 39 %.

3.4 Maximum-a-posteriori Estimation

The a-posteriori-distribution (1) is quite irregular, because, starting from the optimum, shifting the position $[Y, Z]$ of the stairs by one step $[t, h]$ leads to a local optimum. Also, decreasing or increasing the number of stairs leads to a local optimum. Therefore simple gradient descent techniques will not be able to yield the global optimum.

There are various methods which may cope with this situation. Two of them appear to be promising and have been investigated:

- Simulated annealing is a Monte Carlo technique which tries to find an optimum score under very general conditions by generating random samples of the parameter vector, evaluating the score using the optimization function, accepting it in case the new score is better and with slowly decreasing probability accepting parameter vectors which a worse score. As the annealing schedule guaranteeing a global optimum is very slow, one may decrease the acceptance probability faster, facing the risk of getting trapped in a local minimum or of refining the sampling scheme.

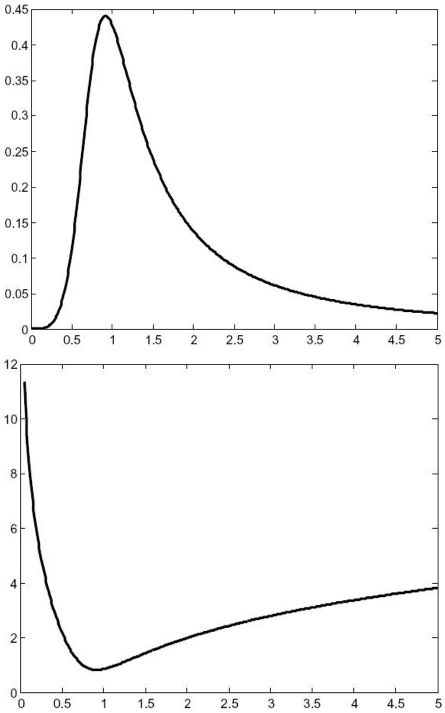

Figure 4: Likelihood function for relative number $n_i/\text{med}(n_s)$ points for first and last step. Low numbers yield low support.

- Markov Chain Monte Carlo techniques aim at a faster sampling scheme, by making the new sample dependent on the previous one using a intermediate distribution. Under certain conditions, the distribution of the chosen samples follows the target distribution, in our case the posterior distribution.

We use both techniques for finding the maximum of the a-posteriori density (1).

4 NUMERICAL OPTIMIZATION OF A-POSTERIORI-DISTRIBUTION

4.1 Target Distribution / Scoring Function

The target distribution is composed by the multipication of the respective priors and the likelihood term.

With respect to Bayes' theorem the target distribution, i.e. the a posteriori density is of the form:

$$\pi(\boldsymbol{x}) \doteq p(\boldsymbol{x}|\boldsymbol{d}) = \prod_{i=1}^{I} p(d_i|\boldsymbol{x}) \cdot \prod_{k=1}^{6} p(\boldsymbol{x}_k) \cdot \eta \qquad (10)$$

The distribution for information terms accordingly:

$$I(\boldsymbol{x}) = I(\boldsymbol{d}|\boldsymbol{x}) + \sum_{i=1}^{6} I(\boldsymbol{x}_i) - \text{lb}\eta \qquad (11)$$

The data $\boldsymbol{d}$ are given, and the parameters $\boldsymbol{x}$ are to be found. So the task is to find a maximum in $\pi(\boldsymbol{x})$ or a minimum in I by fitting the parameters $\boldsymbol{x}$.

In the following we use the distribution for probabilities:

$$\pi(x) = \eta e^{-I(\boldsymbol{x}|\boldsymbol{d})} \cdot \prod_{i=1}^{6} e^{I(\boldsymbol{x}_i)} \qquad (12)$$

The factor η is not relevant here. We have to maximize this target function. To do this we use Markov Chain Monte Carlo (MCMC).

4.2 Starting Values

To start the Markov Chain process for all dimensions starting values have to be defined. Though the starting values do not have influence on the samples after the burn-in it is recommended to choose reasonable values to achieve faster convergence.

The starting values can be determined with respect to building codes or norms like ISO. For the translation one can use zero for all three dimensions.

4.3 Proposal Distributions / Jumping Distributions

The jumping distributions or proposal distributions $q(.|X_t)$ define transition probabilities of the Markov chain for the transition from a state X_t to a new state X_{t+1}. The distribution q has to be chosen so that the Markov chain has rapid convergence and rapid mixing.

Here we have chosen a normal distribution $N(X_t, S)$ for most of the parameters, because we assume that the parameters are somehow normally distributed in the search range. This is a very rough assumption.

The variances have to be fitted to the expected dimension of the parameters. If the possible variance of a parameter is bigger, the variance of the proposal distribution must be bigger too.

Alteration of proposal distributions To find a maximum optimally one can adjust the variance of the proposal distributions. We start with a higher variance and during the run decrease the proposal distribution. So one can make bigger steps in the beginning to get near the maximum and make smaller steps in the end to get the maximum more exact in a smaller number of steps. The variance can be changed for every step during a run of the Markov Chain, before another run etc.

4.4 Metropolis-Algorithm

How the rejection method of the MCMC process works is described in the following:

- The Markov chain is initialized by a starting value.
- A point Y is sampled from the proposal distribution.
- A value U is sampled from a random uniform(0,1) variable.
- A factor α is generated with respect to the actual and the former state of the Markov chain and the proposal distributions.
- If $U \leq \alpha$
  Y is accepted as $X_t + 1$
  else
  the Markov Chain does not change: $X_{t+1} = X_t$.
- The last 4 steps are repeated.

In our case X and Y are 6D-points of the 6D-distribution $\pi(\boldsymbol{x})$. Each point stands for a combination of the 6 parameters of the stairs.

In the following the variables α, X and Y correspond to those described above. The factor α can be determined in different ways. The standard formula is used in the Metropolis-Hastings algorithm:

$$\alpha(X, Y) = \min\left(1, \frac{\pi(Y)q(X|Y)}{\pi(X)q(Y|X)}\right) \tag{13}$$

We use a more simple form (Metropolis-Algorithm):

$$\alpha(X, Y) = \min\left(1, \frac{\pi(Y)}{\pi(X)}\right) \tag{14}$$

Here does not appear any proposal distribution in the formula. Because of the symmetry of the normal distributions it is assumed that $q(X|Y) = q(Y|X)$.

We implemented this algorithm by use of $\ln \pi(\boldsymbol{x})$, see below.

4.4.1 Blocking / Updating Order

Update -en bloc- One can update the dimensions of the multivariate Markov chain in one step en bloc. One tests or rejects a totally new candidate with all components generated new in an update step. So eventually some dimensions have values which converge better to the scoring function while others have values which converge worse.

If one has many parameters the process in this way will converge and mix very slowly. So it could be necessary to update the components one-by-one.

Single-Component Update With the single-component-update one updates each component of the multidimensional Markov Chain in a single step. One only changes one dimension in an update step. With this one step one moves along the direction of the coordinate axes of this dimension. So one can test and reject a candidate with respect to a single dimension.

Our approach Because of the low number of parameters we decided to use Update -en bloc-. Single-Component Update did not lead to a significant enhancement of the MCMC- process. We decided to estimate different groups of parameters separately. In a first run the two parameters for the position $[Y, Z]$ and the slope are determined. In a second run we correct the slope and find the value for the tread depth. The number of steps is determined in a special process (cf. below). To improve the position results one could make a third run to improve the two position parameters. This step has not been tested up to now.

4.4.2 Simulated Annealing To determine the maximum of our scoring function we use a principle of Simulated Annealing. This means that the acceptance rate is decreased by calculating another α before the comparison with U.

The following is the standard approach for Simulated Annealing with Metropolis-Hastings-Algorithm :

$$\alpha(X, Y) = \min\left(1, \frac{\pi(Y)^{\frac{1}{T_i}}q(X|Y)}{\pi(X)^{\frac{1}{T_i}}q(Y|X)}\right) \tag{15}$$

Practically this was implemented with the Logarithm of π which also makes the calculation here easier.

Hereby T_i is decreasing with a cooling schedule. It converges to zero. T_i can be determined with two parameters, e.g. C and T_0 cf. (Andrieu et al., 2003), p. 20:

$$T_i = (C \ln(i + T_0))^{-1} \tag{16}$$

To achieve that $T_1 = 1$ we set $C = 1/\ln(1 + T_0)$. So we start with standard Metropolis-Algorithm and increase the rejection rate during the run.

4.5 Details for the Implementation

Metropolis-Algorithm The calculation of the algorithm was not done by using the values of π but by using those of $\log(\pi)$. This was necessary because of numerical problems.

The α was determined with

$$\ln \alpha = \ln \pi(Y) - \ln \pi(X)$$

With the use of Simulated Annealing:

$$\ln \alpha = \frac{1}{T_i}(\ln \pi(Y) - \ln \pi(X))$$

In the actual configuration the range of the values of the scoring function varies strongly. The Metropolis-Algorithm rejects every result which leads to a lower value for the target distribution than the result before. So the Markov Chain often does not move. This could in future be improved by adjusting the values of π which means to change the model so that π is better adjusted to the distribution U. Therefore in the actual configuration simulated annealing has no influence on the results.

At the moment the estimation of the number of stairs with MCMC is not realized yet. We use another method by classifying the tread and riser edges by the number of 3D points near them. It is in progress to integrate the likelihood term described above into the approach.

Parameters

Starting Values, cf. Section 4.2

- number of steps: 10; 5 below and 5 above the reference point
- reference point: $\begin{bmatrix} 0 \\ 0 \end{bmatrix}$ cm
- tread depth: 29 cm
- slope: $17/29$

Proposal Distributions, cf. Section 4.3 normal distributions $N(X_t, \sigma^2)$, standard deviations given in the table:

- reference point X resp. Z coordinate: from $\sigma_{start} = 10$ to $\sigma_{end} = 2$ cm
- tread depth: $\sigma_{start} = 1.5$ to $\sigma_{end} = 0.3$ cm
- slope: $\sigma_{start} = 0.03$ to $\sigma_{end} = 0.01$

The standard deviation is exponentially decreased during a run.

Number of steps in the runs We use 80 steps in the first run (reference point and slope) and 30 steps in the second run (slope and tread depth)

5 RESULTS

We generated test data in the 2-dimensional YZ plain. For every step tread and riser here are represented by edges. Next to every tread and riser about ten points were generated regularly along the edge and randomly (normal distribution) differing from it. The profile before and after the steps was also generated. Every point here was roughly positioned dependent from the two points next to it in direction of the stairs and exactly positioned randomly.

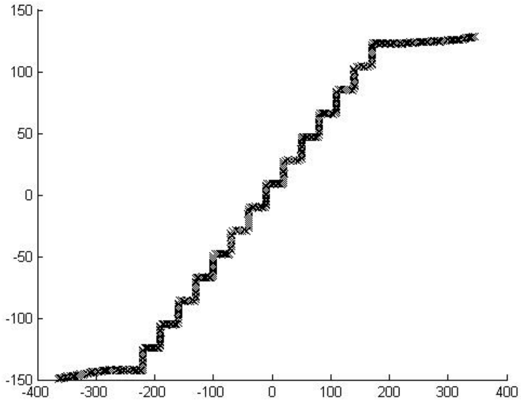

Figure 5: Test data randomly generated from a generative model with specified parameters noised with a normal distribution with $\sigma = 0.25$cm. Different scales for the two dimensions.

Test Data We tested two examples of generated stairs, one with a good approximation, one with a bad approximation.

Good example, parameters, given below, close to the starting values (400 2D points):

- number of steps: 14; 7 below and 7 above the reference point
- reference point: $\begin{bmatrix} -10 \\ -10 \end{bmatrix}$ cm
- tread depth: 30 cm
- slope: $19/30$

Bad example, parameters, given below, afar from the starting values (400 2D points):

- number of steps: 4; 2 below and 2 above the reference point
- reference point: $\begin{bmatrix} -50 \\ 70 \end{bmatrix}$ cm
- tread depth: 26 cm
- slope: $14/26$

We also tested real data. The 3D points were generated with a stereophotogrammetric approach with support of a software from TU Berlin (Heinrichs/Rodehorst). The 3D points were manually rotated and projected into the YZ-plain. A scale was determined by manual measurements as well. For the projection only points in the area of the stairs itself and not those in the area right or left aside were taken into account.

Real example, parameters determined manually, given below (300 2D points):

- number of steps: 3
- reference point: $\begin{bmatrix} 0 \\ -0.5 \end{bmatrix}$ cm
- tread depth: 29.7 cm
- slope: $15.1/29.7$

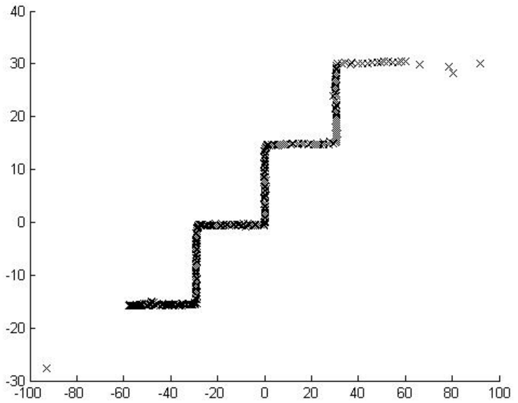

Figure 6: Real data: 3D points determined with software from TU Berlin, projected into the YZ-plain. 300 randomly chosen points of originally more than 1200 points were used here. The example contains no outliers because there were no objects standing on the stairs. The platforms below and above the stairs are not represented here because there were no matches found by the software.

Results The MAP Estimation as described above was made 30 times.

We get good results for the first example, here average and the standard deviation.

- reference point $\begin{bmatrix} -10.75 \\ -9.73 \end{bmatrix} \pm \begin{bmatrix} 0.82 \\ 0.65 \end{bmatrix}$ cm
- tread depth: 30.10 ± 0.14 cm
- slope: $19.01/30 \pm 0.17/30$

For the second example the estimation does not work well. The reason is the bad approximation for both tread depth and slope. So the model converges during the first run to an approximation which is quite good only for some steps of the stairs. If the reference point is not on one of these steps, it cannot be determined correctly. A solution for this probably is the change of the reference point to one of the good approximated steps. This could not be tested yet. A bad approximation for the translation is no problem here.

We also get good results for the real data. The starting values are a good approximation here.

- reference point $\begin{bmatrix} 0.61 \\ -0.36 \end{bmatrix} \pm 0.430.39$ cm
- tread depth: 29.88 ± 0.26 cm
- slope: $15.15/29.7 \pm 0.17/29.7$

The difference between the starting value for the slope and its real value (2cm) does not cause any problems here.

6 CONCLUSION

The paper is an investigation into the reconstruction of stairs from 2D point clouds with a generative model and MCMC techniques. We construct a generative model with 6 parameters for 2D reconstruction which can be extended to a 9 parameter model for 3D reconstruction.

We use a MAP estimation to find the optimal parameters. The a-posteriori distribution is used as scoring function for the MCMC algorithm. Four of the six parameters of the 2D model are determined with the help of the first likelihood function. The determination of these four parameters is tested.

If the approximation for tread depth and slope is not too bad we get quite exact results for the stair parameters. Admittedly the scope here rather is not precision, but reliability. The reliability has yet to be analysed more precisely. If the distance between the starting values for tread depth and slope is too high improvements are necessary. An approach could be to change the position of the reference point or another rejection criteria during the MCMC process which accepts more results which lead to a lower value for the scoring function. For the test data the prior probability distributions have almost no influence on the results. So more tests with weak data consisting of just a few points are recommended to examine the priors.

The second likelihood function to determine the number of stairs in a third run should be no problem. The approach can be extended to 3D if the step width is taken into account as parameter. For this parameter a special likelihood function is needed because the likelihood used at the moment is not sensitive if the width is too high.

We have not yet tested to find the parameters with other common methods. Therefore we cannot say if the MCMC approach is really necessary and if it is better than other approaches. This will be future work.

In the future there are many approaches possible to determine different kinds of stairs with special models. For those detections more criteria are needed to find an optimum.

REFERENCES

Andrieu, C., de Freitas, N., Doucet, A. and Jordan, M., 2003. An Introduction to MCMC for Machine Learning. Machine Learning 50, pp. 5–43.

Brémaud, P., 1999. Markov Chains: Gibbs Fields, Monte Carlo Simulation, and queues. Springer.

Brenner, C. and Ripperda, N., 2006. Extraction of Facades using rjMCMC and Constraint Equations. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences 36 part 3, pp. 155–160.

Dick, A., Torr, P. and Cipolla, R., 2004. Modelling and Interpretation of Architecture from Several Images. International Journal of Computer Vision 60(2), pp. 111–134.

Gilks, W. R., Richardson, S. and Spiegelhalter, D. J. (eds), 1996. Markov Chain Monte Carlo in Practice. Chapman & Hall/CRC.

Mayer, H. and Reznik, S., 2005. Building Facade Interpretation from Image Sequences. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences 36-3/W24, pp. 55–60.

Mayer, H. and Reznik, S., 2006. MCMC Linked With Implicit Shape Models and Plane Sweeping for 3D Building Facade Interpretation in Image Sequences. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences 36 part 3, pp. 130–135.

Neal, R., 1993. Probabilistic Inference Using Markov Chain Monte Carlo Methods. Technical report, Department of Computer Science, University of Toronto.

Ripperda, N. and Brenner, C., 2006. Reconstruction of facade structures using a formal grammar and rjMCMC. Lecture Notes in Computer Science 4174, pp. 750–759.

Schmittwilken, J., Saatkamp, J., Förstner, W., Kolbe, T. and Plümer, L., 2007. A Semantic Model for stairs. PFG - Photogrammetrie, Fernerkundung und Geoinformation.