Institut für Kartographie und Geoinformatik | Leibniz Universität Hannover

Simulation and Markov Chain Monte Carlo III

Claus.Brenner@ikg.uni-hannover.de

Leibniz Universität Hannover

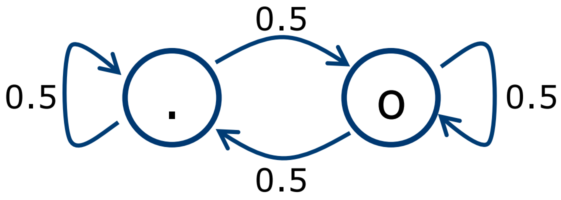
Simulation and Markov chain Monte Carlo

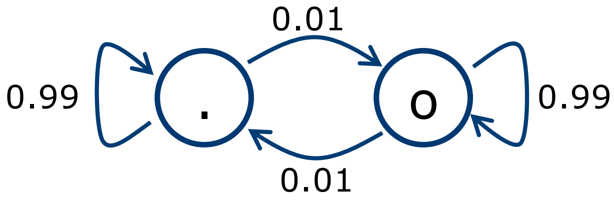
- “Mixing” of a Markov chain – the role of the proposal density
- Integrating simulated annealing with MCMC
- Reversible Jump MCMC
- Examples
  - Façade reconstruction (Nora Ripperda)
  - Building reconstruction (Hai Huang)
  - Forest point processes for the automatic extraction of networks in raster data (Alena Schmidt)

Mixing

Mixing of Markov chains

- Compare those two Markov chains:



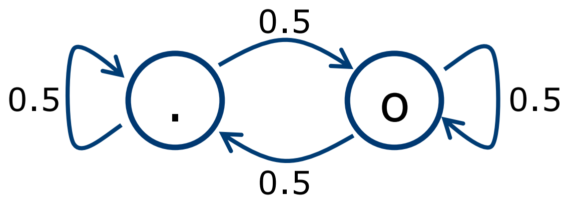


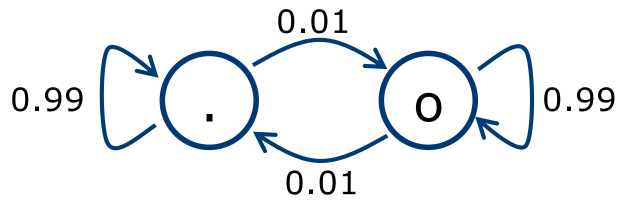
- Evolution of T^n:

```
T_10 =                                    T_10 =
           [[ 0.5  0.5]                             [[ 0.9085364  0.0914636]
           [ 0.5  0.5]]                              [ 0.0914636  0.9085364]]
T_50 =                                    T_50 =
           [[ 0.5  0.5]                             [[ 0.68208484  0.31791516]
            [ 0.5  0.5]]                             [ 0.31791516  0.68208484]]
T_100 =                                   T_100 =
           [[ 0.5  0.5]                             [[ 0.56630978  0.43369022]
            [ 0.5  0.5]]                            [ 0.43369022  0.56630978]]
T_500 =                                   T_500 =
           [[ 0.5  0.5]                             [[ 0.50002051  0.49997949]
            [ 0.5  0.5]]                             [ 0.49997949  0.50002051]]
```

Mixing of Markov chains

- Compare those two Markov chains:





- Example for the first 1000 random draws:

```
...oooo.ooo.o.....oooo.o.o..o..o.oooooo...oo.....
oo.....o....ooooooo..o..ooo..oo.ooooo...oo..ooooo
o.o.....o.oo.o.oo.o..oo.oo.o..o.o.o.ooo..o.o.ooo.
.o..ooo.ooo.ooo.ooo.....o.oo..o.o......oo.ooo....
.o..ooo.ooo..ooooo.oo.oo.o..o...ooooo.o....ooo..o
..o.o..oo.ooo.ooo....o.o......o.o...o..o.ooo.ooo.
....o...o.o....ooo.ooo.o.o.oo.oo....oo.o..o..oo.o
...o...o..o.ooo.....o..oo....oo.....ooo...o..o.o.
o.o..oo..o.oo...ooo....o........o..o...........oo
.oooooo.ooo..o..o..o.oo..oooo.o.oo..o.oooo...o...
o.o.oo.oo......oo.o..ooo.ooo...o.o..o.oo....oo.o.
ooooo....oo...ooo.o...o...oooo..oo.o.o...o..o.oo.
....o..o..ooo...oo.o...o.....oo..ooo.o....oo.....
..ooooo.oo.oo.o.oo..o.....o.o...o..o...oo.o..oooo
.o.o.oooooooooo.oo..oo..o...ooo.oo.o..ooooo..oo..
o...o...o.o.oo.o.o..oo.o.ooo.oo..ooooo..oooo...oo
oooooooooooo.o.o..ooo...o..o.oooo...oo..o.o....o.
.o..ooo....ooo..o.....o......o..o..oooo.oooo..ooo
.o.oooo.....o..ooo..ooo.o...o.o......o..oo....oo.
o...oooo..oo..o..oo.o.o.o......ooo..oo.oo...o....
...ooooo.o...oo...o.
```

```
.................................................
.................................................
.................................................
..............................ooooooooooooooooo..
.................................................
.................................................
.................................................
.........................ooooooooooooooo.........
.................................................
.................................................
.................................................
.................................................
.................................................
.................................................
.................................................
.................................................
.................................oooooooooooooooo
ooooooooooooooooooooooooooooooooooooooooooooooooo
ooooooooooooooooooooooooooooooooooooooooooooooooo
ooooooooooooooooooooooooooooooooooooooooooooooooo
oooooooooo..........
```

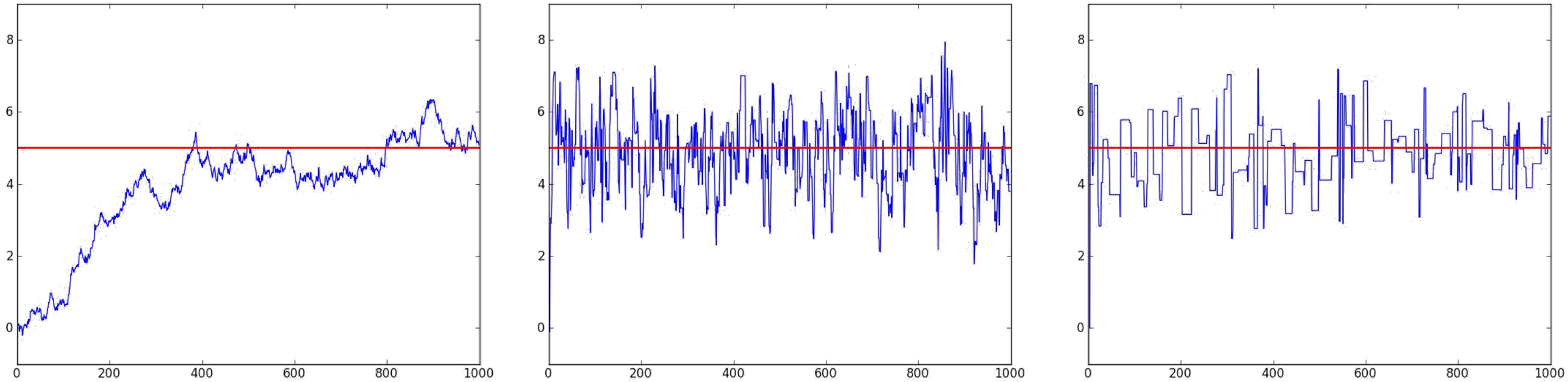
Mixing of Markov chains

- Both chains have the same stationary distribution [0.5, 0.5], to which they converge
- But the first one has better mixing properties
- The same effect arises in MCMC simulations (since these are Markov chains!)
- The effect is controlled by the proposal density
  - The proposal density has to fulfill the requirements from the last lecture, apart from that it can be chosen arbitrarily
  - BUT the choice influences mixing!

Example: simulation of a normal distribution

- Normal distribution to be simulated: $x \sim \mathrm{N}(\mu = 5, \sigma^2 = 1.0^2)$
- Metropolis using initial value x_0=0, 1000 iterations and the following proposal densities:

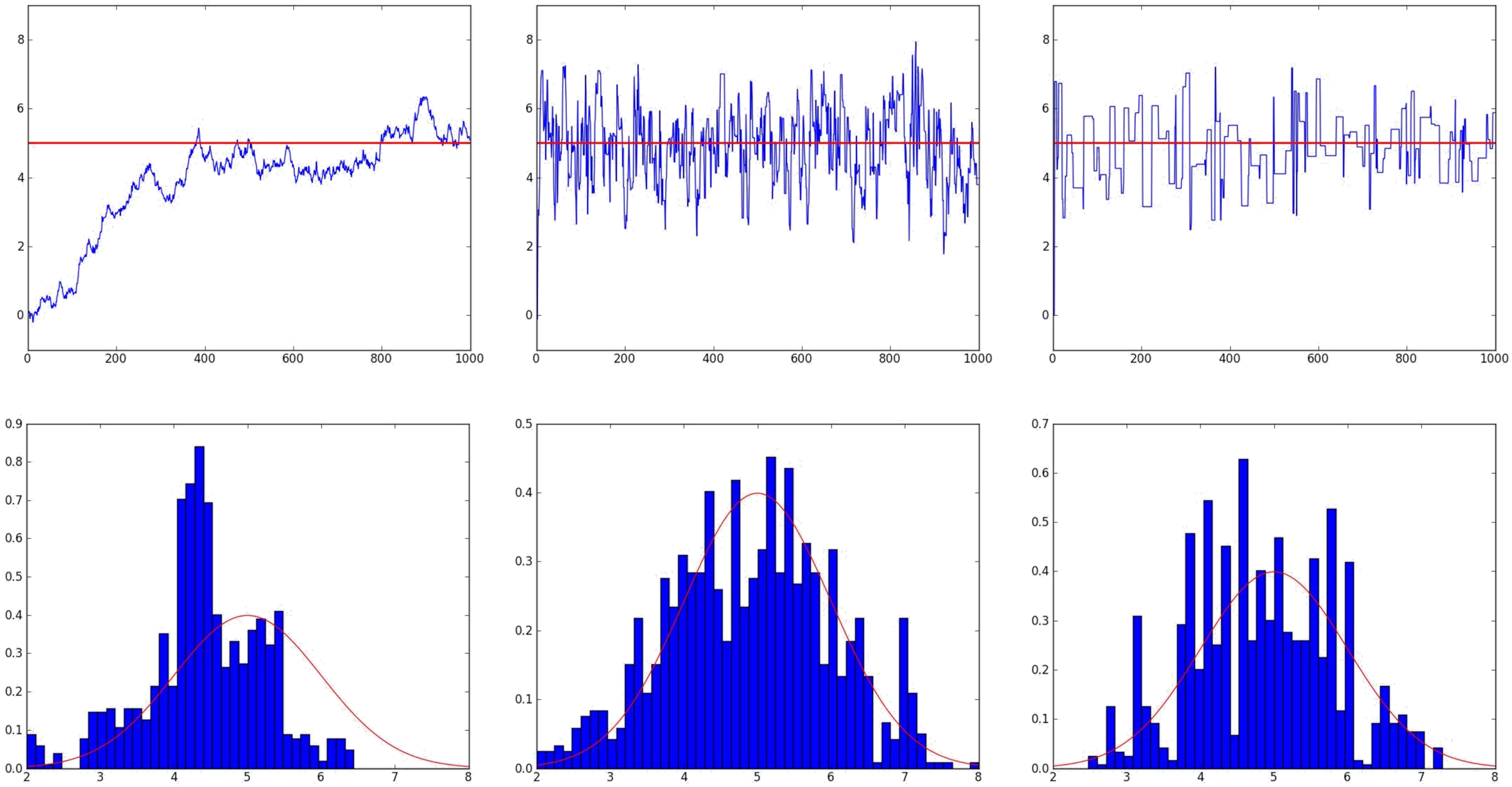
$y \sim \mathrm{N}(\mu = x_t, \sigma^2 = 0.1^2)$ $\quad$ $y \sim \mathrm{N}(\mu = x_t, \sigma^2 = 1.0^2)$ $\quad$ $y \sim \mathrm{N}(\mu = x_t, \sigma^2 = 10.0^2)$



Example: estimate of mean:

$$\hat{\mu} = \frac{1}{m}\sum_{j=1}^{m} x_j = 3.8626 \qquad \hat{\mu} = 4.8772 \qquad \hat{\mu} = 4.8159$$

Comparison: histograms vs. target distribution



red: $N(\mu = 5, \sigma^2 = 1.0^2)$

Simulated annealing

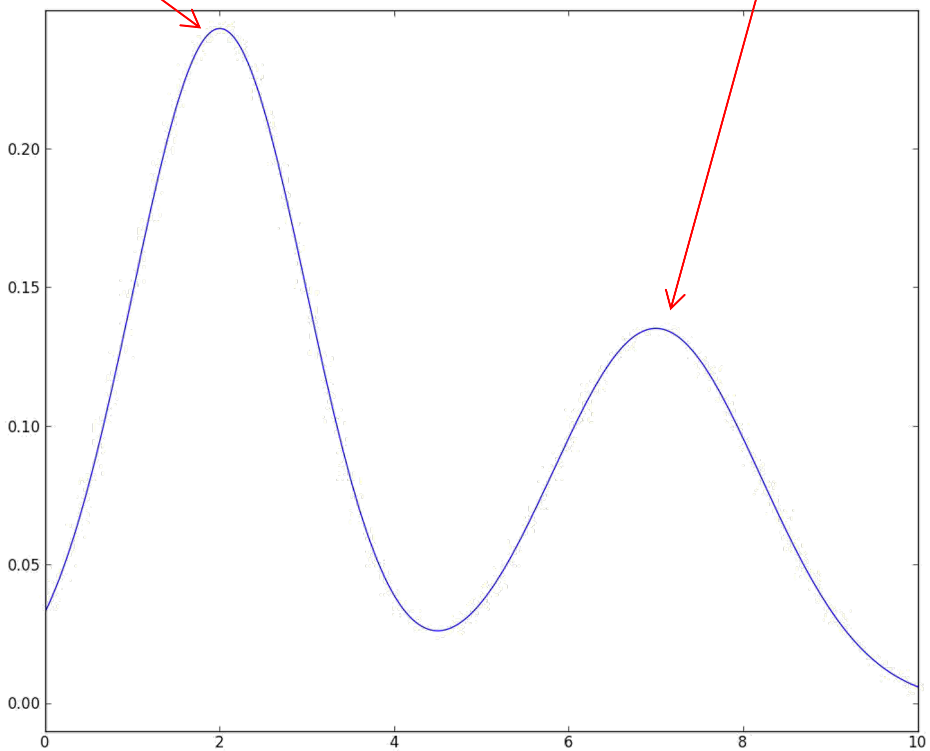
Simulated annealing

- We have used MCMC to find the **maximum** of a score function (e.g. maximum a posteriori (MAP) estimate)
- MCMC is advantageous compared to a uniform distribution (if the latter can be used at all), since more samples are drawn from regions with a high score value
- This effect can be increased when "simulated annealing" is used
- Using this, the global maximum can be found with probability 1 (if certain conditions are fulfilled)

Simulated annealing

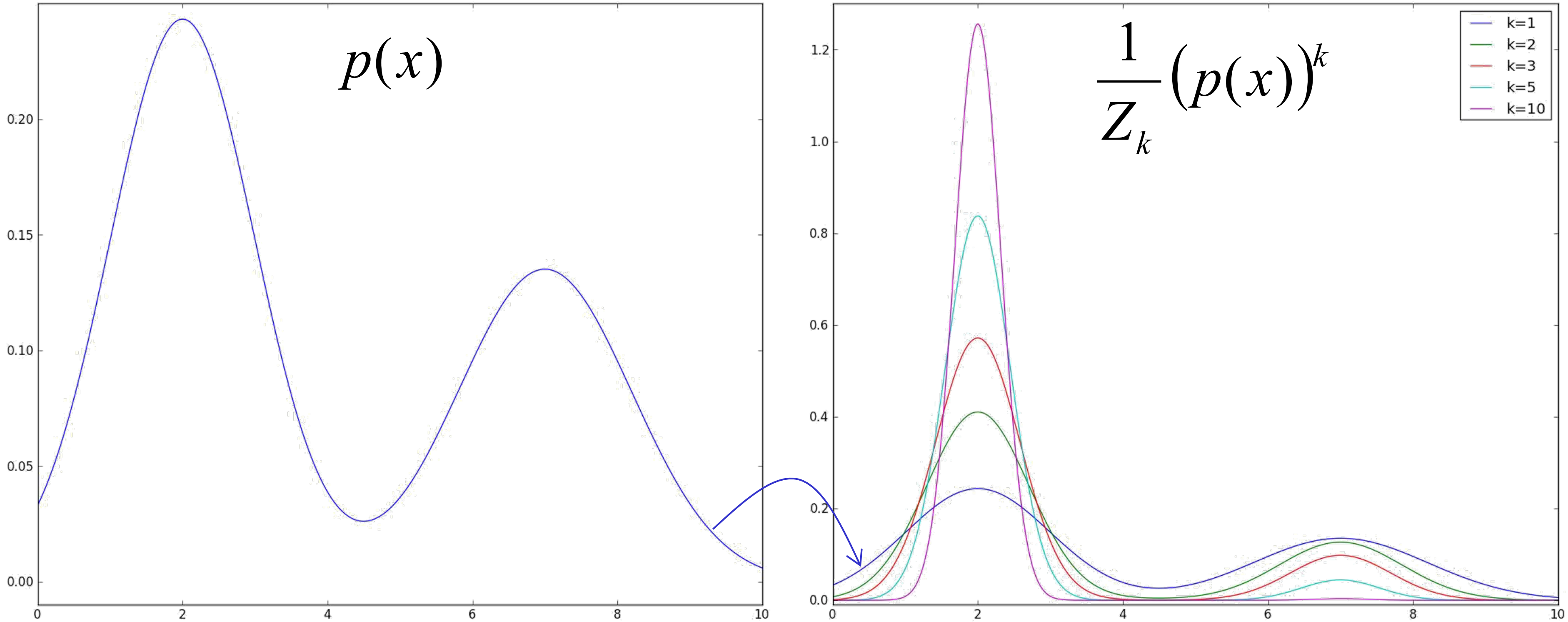
- Example of a (very simple) density with a global and another local maximum

$$p(x) = \frac{6}{10} \mathrm{N}(x \mid \mu = 2, \sigma^2 = 1.0^2) \quad + \quad \frac{4}{10} \mathrm{N}(x \mid \mu = 7, \sigma^2 = 1.2^2)$$



Simulated annealing

- What happens when we raise it to the power of k: $(p(x))^k$ (normalized to 1)
- Observation: when k increases, the global maximum stands out more clearly (c.f. maximum norm)



Integrating MCMC with simulated annealing

- Idea: use MCMC, but also increase *k* during the simulation
- If $p(x)$ has a global maximum at $x_{\max}$, then the sequence:

$$1/Z_k\left(p(x)\right)^k \rightarrow \delta(x - x_{\max})$$ (Dirac delta)

- I.e., with increasing *k* the random draws concentrate around the global maximum
- Typically, instead of *k*, a "temperature" *t* is used

$$k = 1/t$$

Integrating MCMC with simulated annealing

- Modification of the Metropolis-Hastings algorithm:
  - Instead of $p(x)$ one simulates: $1/Z_{1/t_i}\left(p(x)\right)^{1/t_i}$
  - The "temperature" t_i starts at 1.0 and decreases towards 0 with increasing *i*
  - The relation between t_i and *i* is defined by the "cooling scheme". The temperature decrease has to be "sufficiently" slow
- The only modification to the Metropolis-Hastings algorithm is the computation of the acceptance probability:

$$\alpha(X_i, Y) := \min\left(\frac{(\pi(Y))^{1/t_i}\, q(X_i \mid Y)}{(\pi(X_i))^{1/t_i}\, q(Y \mid X_i)}, 1\right)$$

- Note:
  - The normalization (partition function) Z_{1/t_i} needs not to be included since this cancels out
  - The Markov chain defined in this way is not homogeneous anymore!

Metropolis-Hastings with simulated annealing

```
x = Initial value
Initialize   π_max, x_max, t_1
for i=1...m:
    draw   y ~ q(·|x)
    compute   α = min( (π(y)/π(x))^(1/t_i) · q(x|y)/q(y|x), 1 )
    draw   u ~ uniform(0,1)
    if  u ≤ α :
        x = y
        if  π(x) > π_max:
            π_max = π(x),   x_max = x
    decrease the temperature t_i
return x_max
```

x = Initial value

Initialize $\pi_{\max}, x_{\max}, t_1$

for i=1...m:

draw $y \sim q(\cdot \mid x)$

compute $\alpha = \min\left(\left(\frac{\pi(y)}{\pi(x)}\right)^{1/t_i} \cdot \frac{q(x \mid y)}{q(y \mid x)}, 1\right)$

draw $u \sim \text{uniform}(0,1)$

if $u \le \alpha$:

$x = y$

if $\pi(x) > \pi_{\max}$:

$\pi_{\max} = \pi(x), \quad x_{\max} = x$

decrease the temperature t_i

return $x_{\max}$

Metropolis-Hastings with simulated annealing

- Convergence: the maximum will be found with probability 1, if
  - The Markov chain is ergodic for every choice of *t*
  - The cooling scheme is sufficiently slow
- Sufficiently slow: logarithmic cooling

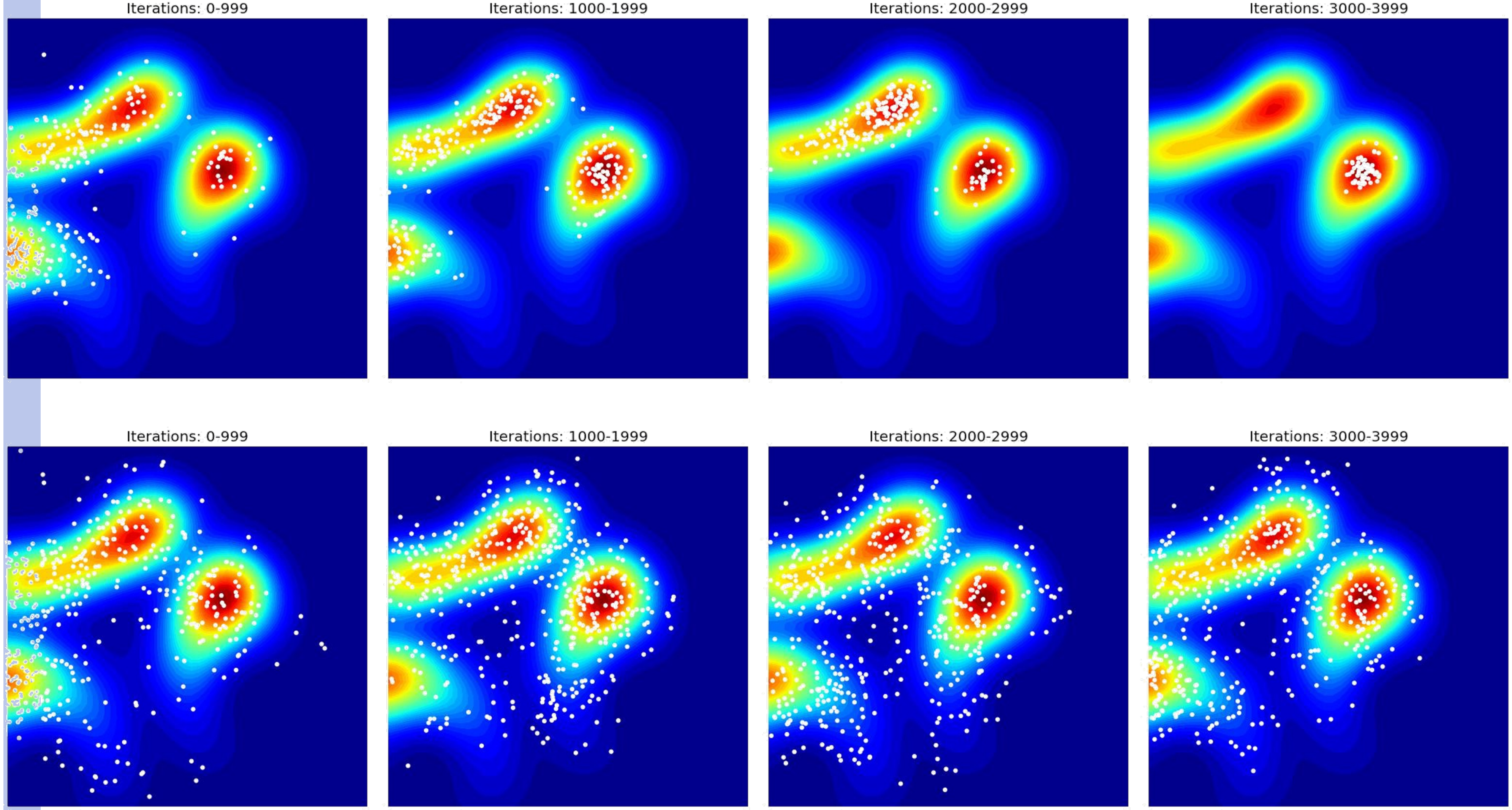
$$t_i = \frac{1}{C \cdot \log(i + t_0)}$$

(with C, t_0 chosen problem-specific)

- This is extremely slow (remember growth of log)!
- Therefore, in practice, one usually tries faster cooling schemes, e.g.:

$$t_i = 1 - (1 - t_{\min})\left(i / i_{\max}\right)^{1/2} \qquad t_i = 1 - (1 - t_{\min})\left(i / i_{\max}\right)^{1/3} \qquad \ldots$$

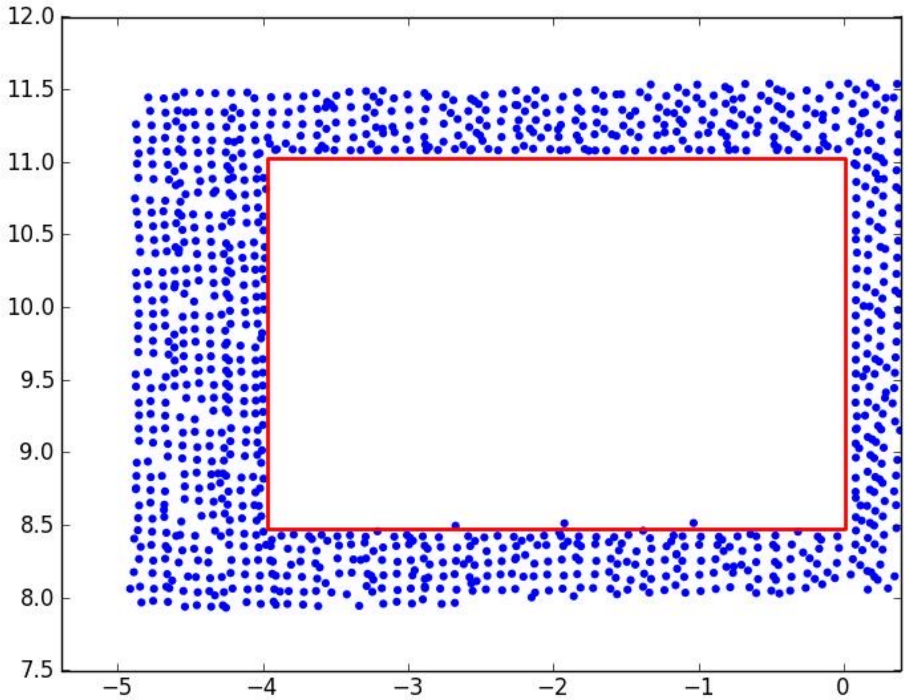
Example: robot localization (exercises)
Above: w/ simulated annealing, below: w/o



rjMCMC – reversible jump MCMC

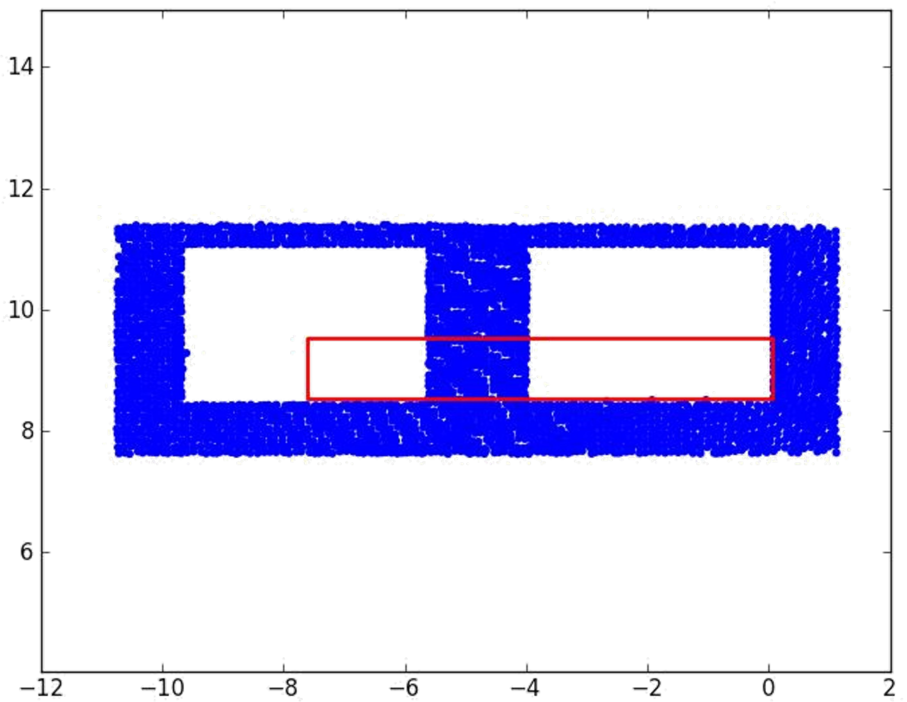
rjMCMC – basic problem

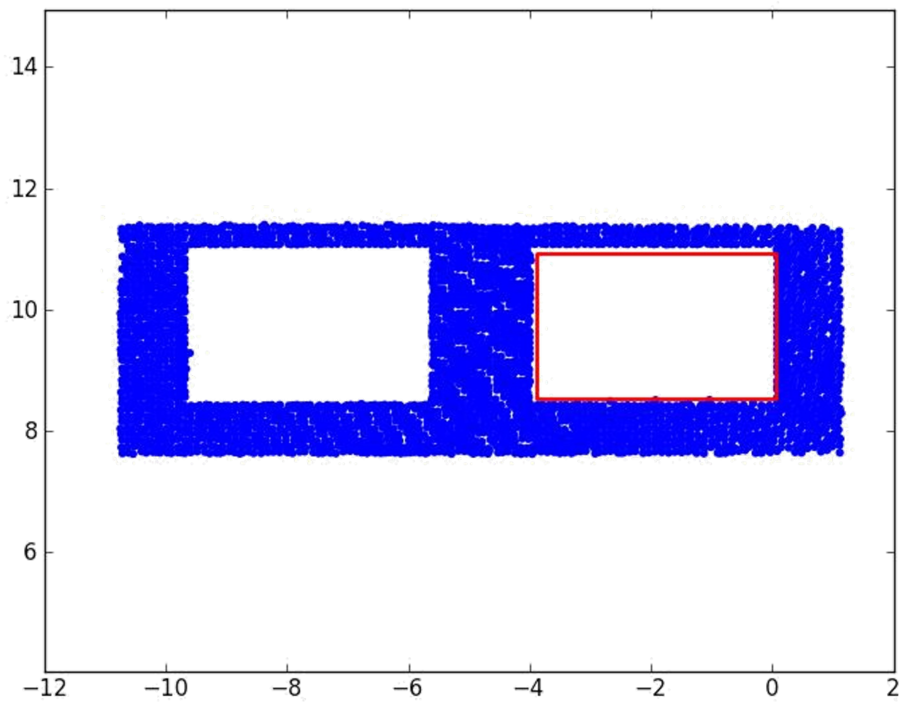
- So far:
  - MCMC was used to draw samples from a space of fixed dimensionality
  - E.g.:
    - Position of a robot: samples are from $(x_j, y_j) \in \mathrm{R}^2$
    - Finding a window: samples are from $(x_j, y_j, w_j, h_j) \in \mathrm{R}^4$



rjMCMC – basic problem

- Problem: what happens in case there are 2 windows?
  - In the best case, we find one of the two (depending on the definition of the score function)



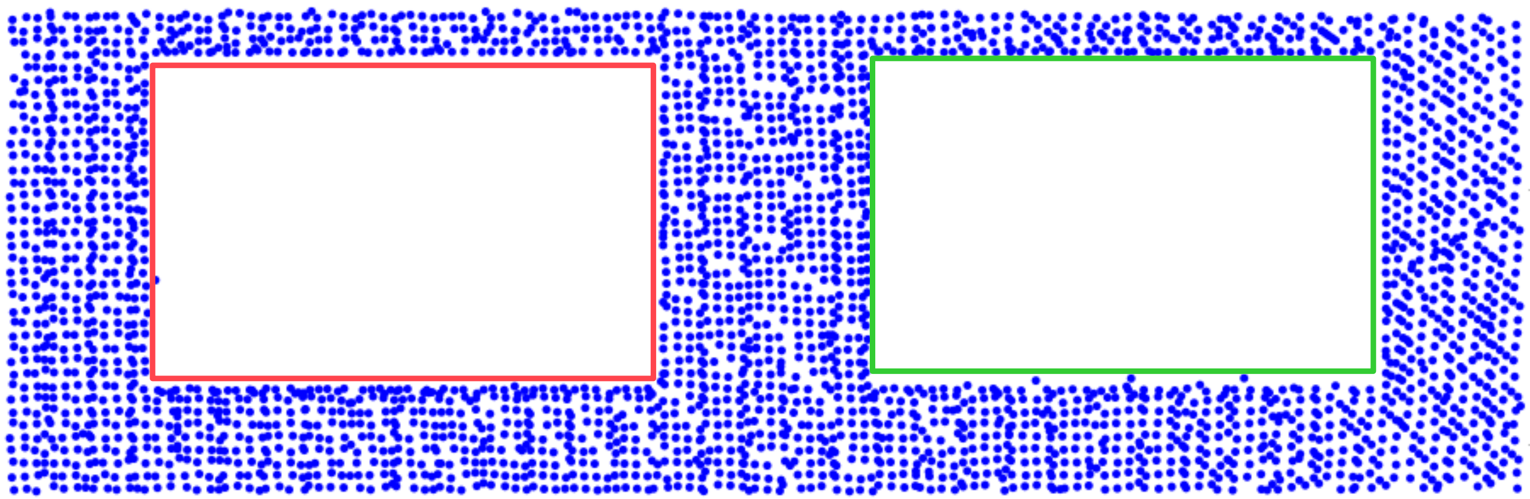


rjMCMC – basic problem

- If it is known in advance that there are two windows, one can optimize the parameters of both of them:

$$(x^{(1)}, y^{(1)}, w^{(1)}, h^{(1)}, x^{(2)}, y^{(2)}, w^{(2)}, h^{(2)}) \in \mathrm{R}^8$$

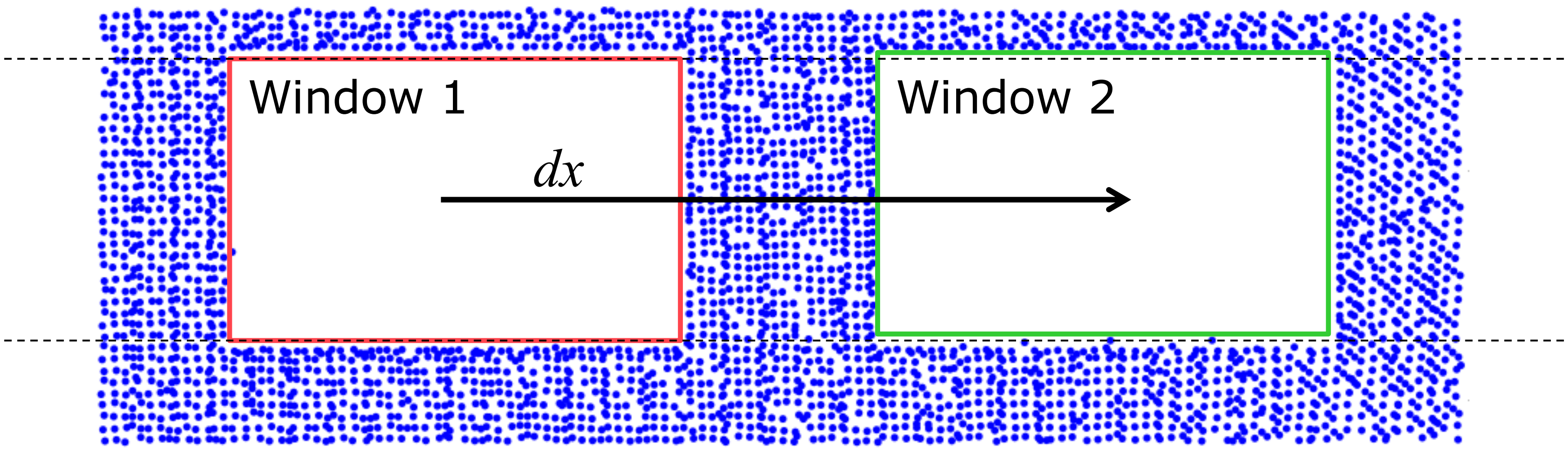
- (in addition, the score function would have to penalize overlapping rectangles)



rjMCMC – basic problem

- Normally, one would prefer a model which enforces identical window sizes and identical lower edges
  - could be modeled as: (x, y), width, height of window 1
  - offset *dx* of window 2

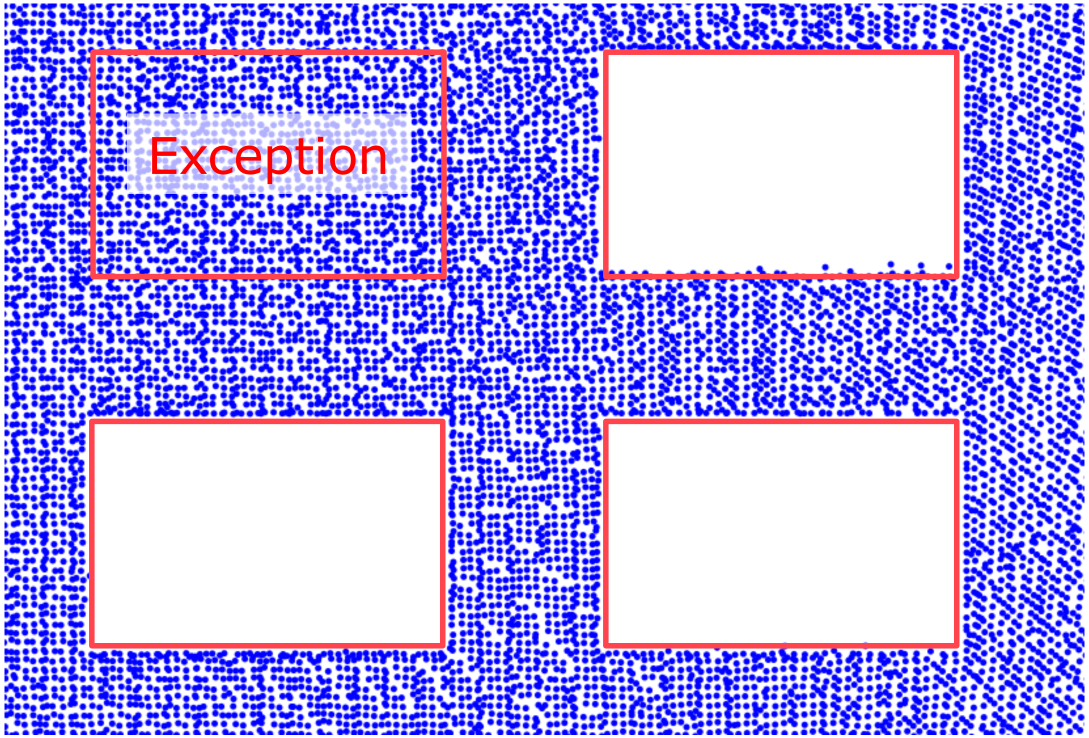
$$(x^{(1)}, y^{(1)}, w^{(1)}, h^{(1)}, dx) \in \mathrm{R}^5$$



rjMCMC – basic problem

- Regular raster
  - Window 1: (x, y), width, height
  - dx and dy
  - Number of rows and columns: r, c = 1, 2, …
  - How to model exceptions?

$$(x^{(1)}, y^{(1)}, w^{(1)}, h^{(1)}, dx, dy, r, c) \in \mathrm{R}^6 \times \mathrm{N}^2$$



rjMCMC – basic problem

- Realization: We have the choice from different models:
  - Single window
  - Two (or more) independent windows
  - Identical windows, which are arranged as a regular raster, in rows and columns
  - Regular row/column arrangements, including exceptions
  - …
- All those models **differ in the number of parameters**

- → The optimization has to be across spaces of differing dimensionality
- A simulation of solutions using MCMC has to “jump” between those spaces

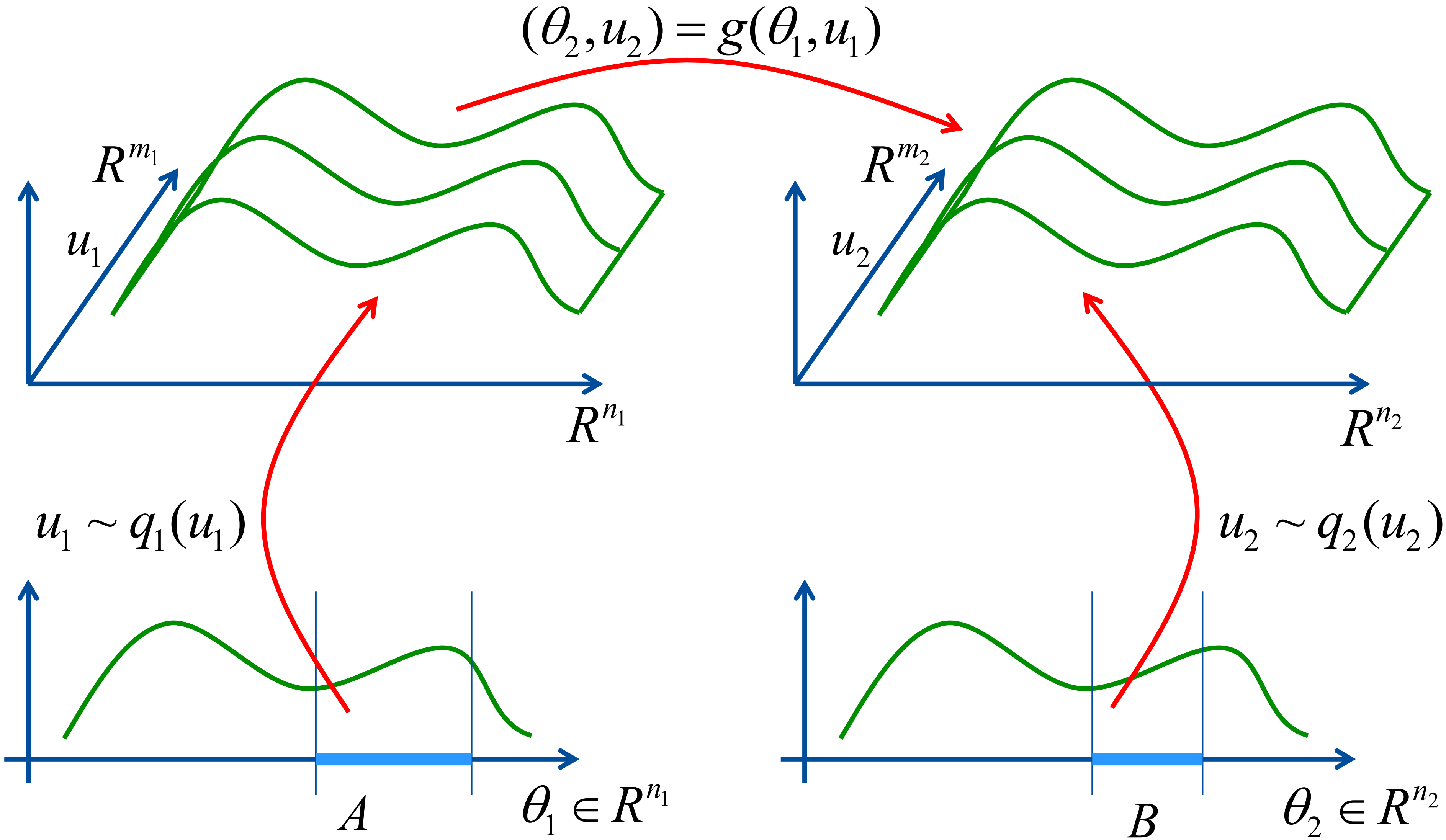
rjMCMC – solution by Green (1995) (sketch)

- The quotient of densities is meaningless when different dimensions are involved → we cannot compute the acceptance probability using the above formula

$$\alpha(x,y) = \min\left(\frac{\pi(y)q(x \mid y)}{\pi(x)q(y \mid x)}, 1\right)$$

- Now, both spaces are complemented, so that they have the same dimension
- A deterministic, differentiable, invertible function is defined as a transformation between the two spaces
- “Jumps” are defined between the spaces, which are reversible
- Now, the acceptance probability has to include the jump probabilities as well as the determinant of the Jacobi matrix of the deterministic transformation.

rjMCMC – solution by Green (1995) (sketch)



$$n_1 + m_1 = n_2 + m_2$$

rjMCMC – solution by Green (1995) (sketch)

- Both spaces are supplemented using a suitable *u*
  - For example, one could do: from 2D to 3D via 4D
  - However, usually there is $m_2 = 0$, for example:
    one window ←→ array of windows

$(x, y, w, h) \in \mathrm{R}^4$	$(x, y, w, h, dx, dy, r, c) \in \mathrm{R}^6 \times \mathrm{N}^2$
$n_1 = 4, m_1 = 4$	$n_2 = 8, m_2 = 0$

- Let $j(\cdot)$ be the probability to select a certain step
- Then, the acceptance probability is given by this ratio (due to Green (1995)):

$$\min\left(1, \frac{\pi(2,\theta_2)\, j(2,\theta_2)\, q_2(u_2)}{\pi(1,\theta_1)\, j(1,\theta_1)\, q_1(u_1)} \left|\frac{\partial(\theta_2, u_2)}{\partial(\theta_1, u_1)}\right|\right)$$

rjMCMC – solution by Green (1995) (sketch)

- Examples for “jumps”:
  - Adding additional components (“**birth**”) or deleting components (“**death**”)
    - for example: “a new window”
  - **Merging** or **splitting** of components
    - for example: removing or adding a component of a mixture distribution

- Normally, the simulation generates samples “inside one space”, then jumps, from time to time, to different spaces, e.g.:
  - Variation of parameters: position, width, height of windows (all continuous), variation of the (discrete) number of rows and columns
  - Jump (change in dimension) single window ←→ array of windows.

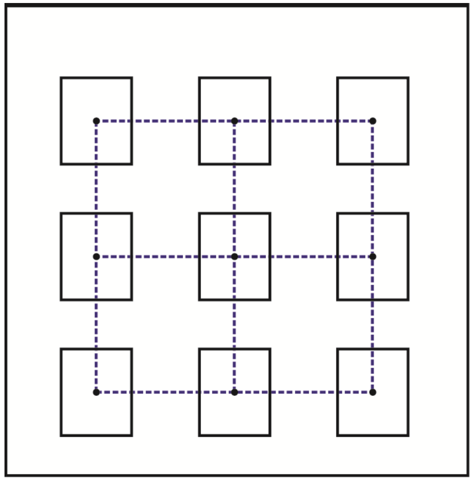
Final remarks on MCMC

- Reminder:
  - **Any method**, which simulates a distribution f by generating an ergodic Markov chain which has f as its stationary distribution, is termed a “Markov chain Monte Carlo” method.
- There are more methods which were not covered in this lecture, e.g.:
  - Gibbs Sampling
  - Slice Sampling.

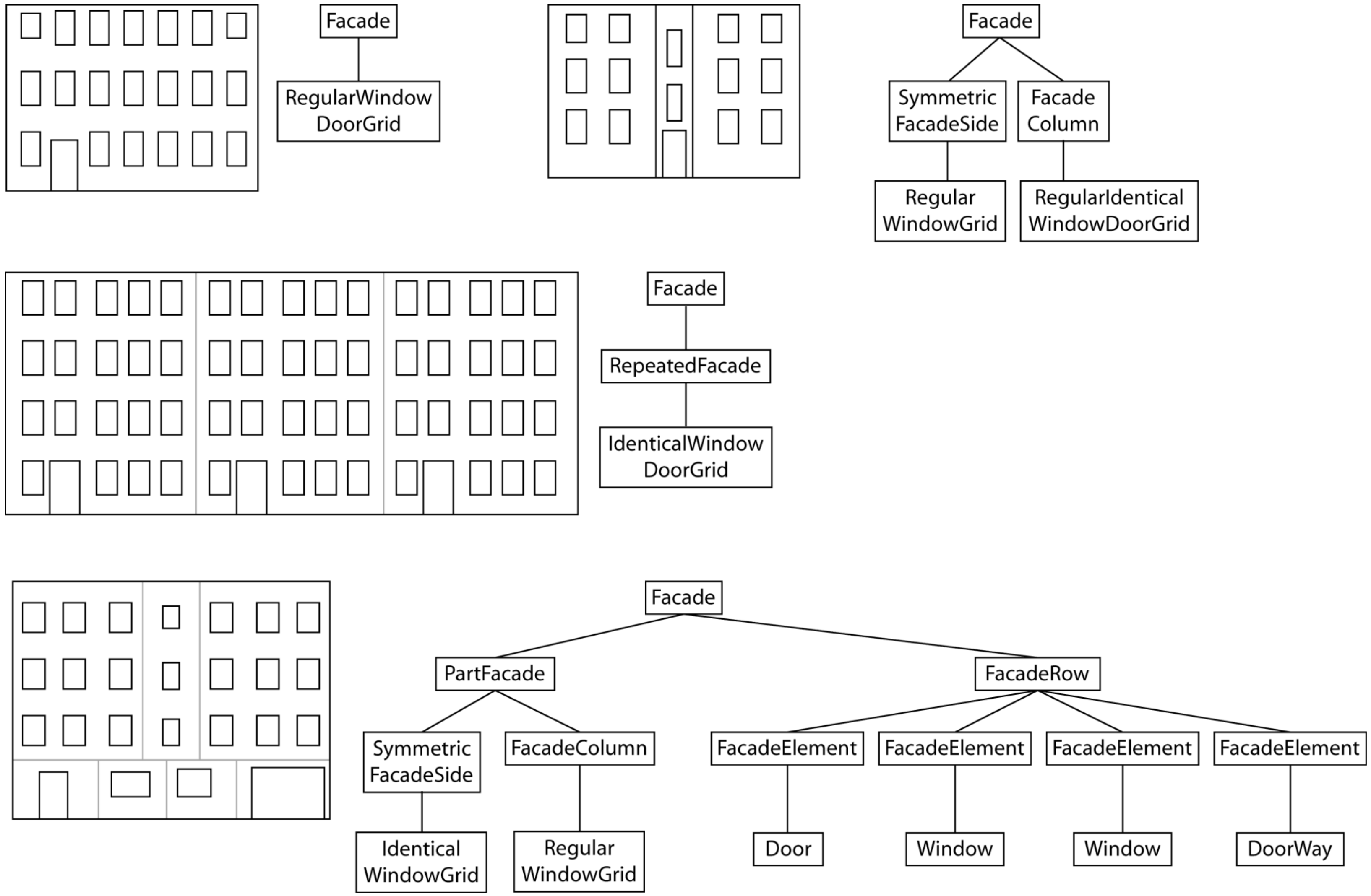
Example 1: Façade reconstruction
Nora Ripperda (2009)

Modelling using formal grammars

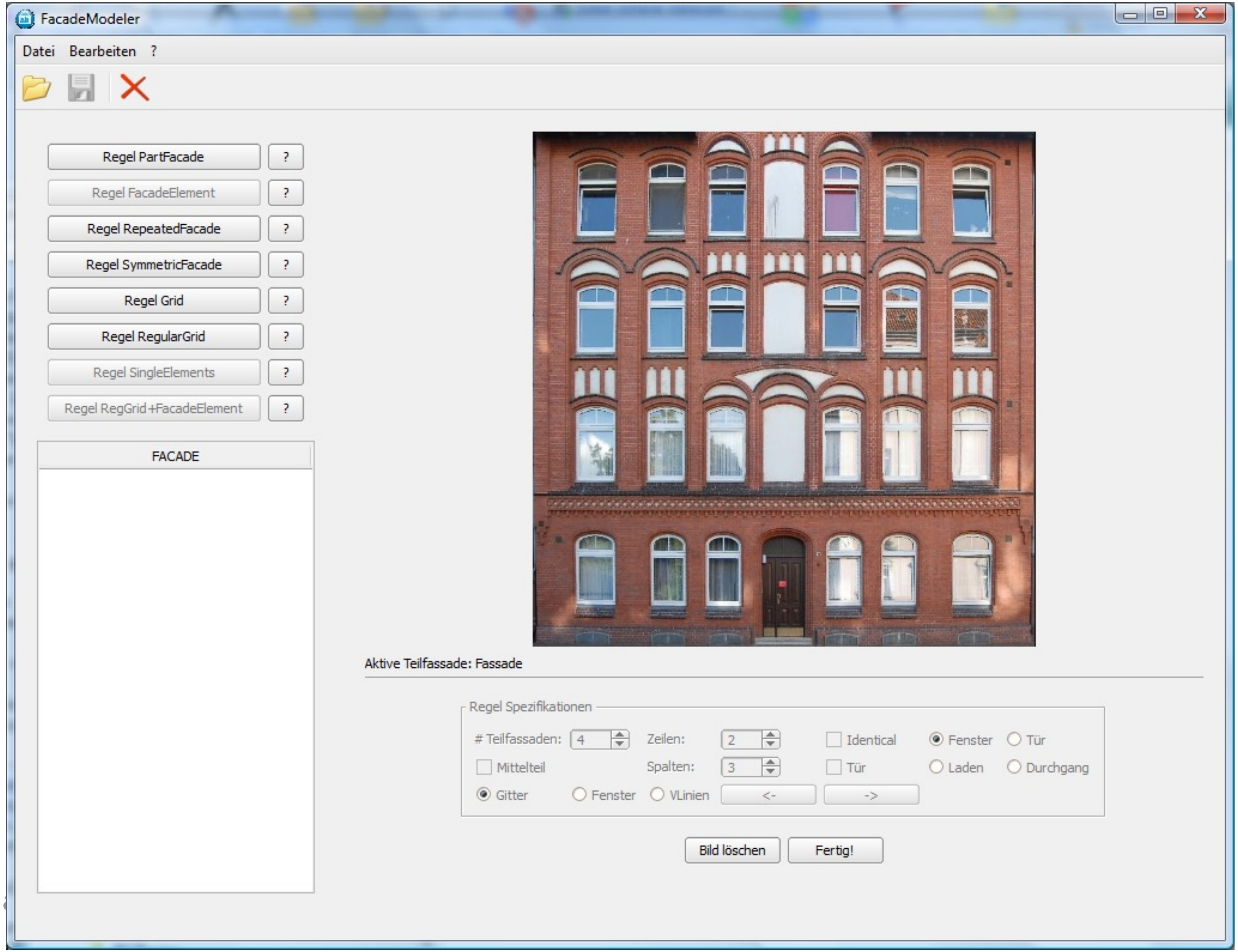
- Building facades are designed according to certain rules
  - Symmetries, grid structures (arrays)
- Often, only a geometric reconstruction is made
  - No information about the structure
  - No semantics
- Modeling using a grammar
  - Introduces semantics: objects have a meaning
  - Constraints can be formulated
  - Compact storage



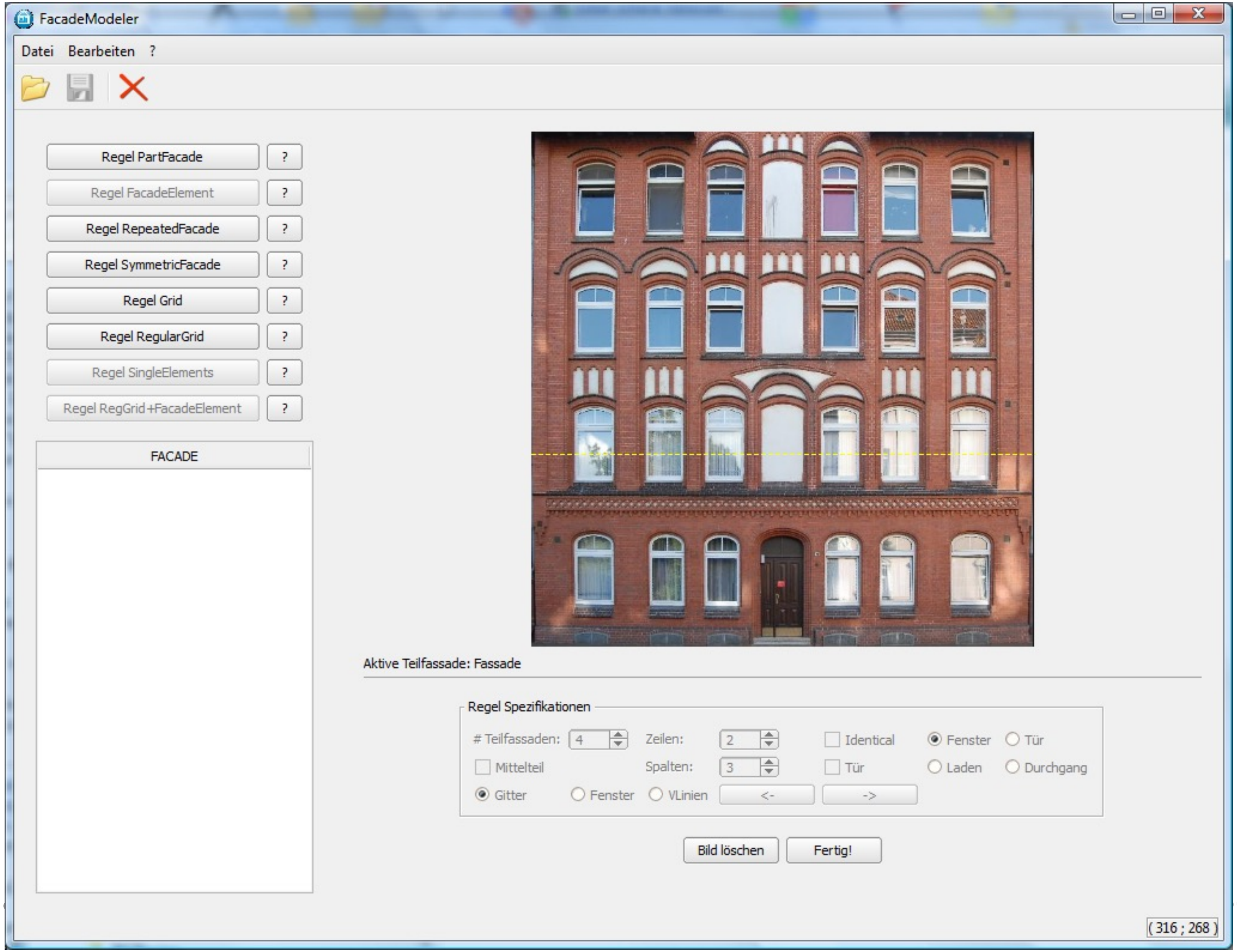
Example models



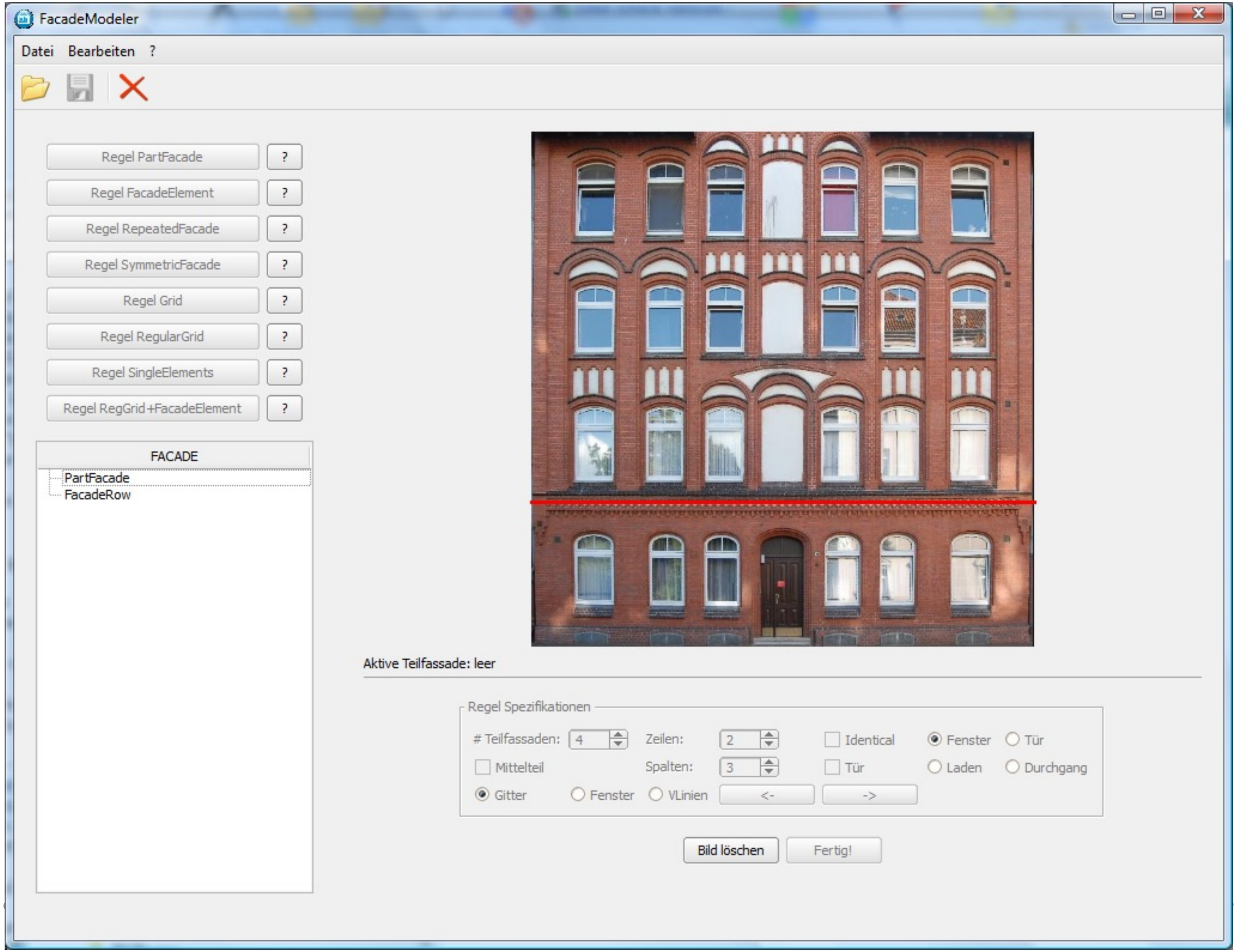
Start: manual modeling…



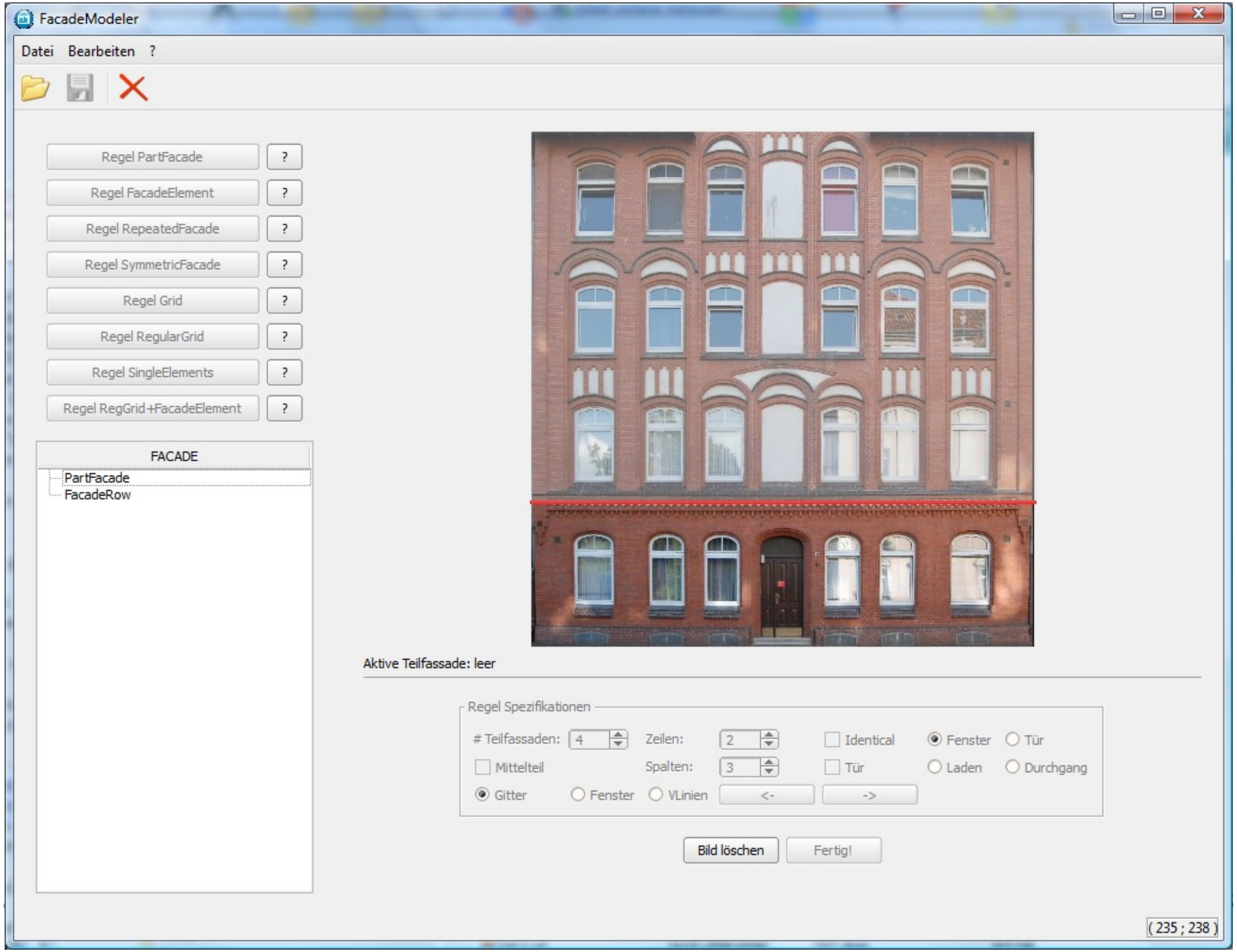
Start: manual modeling…



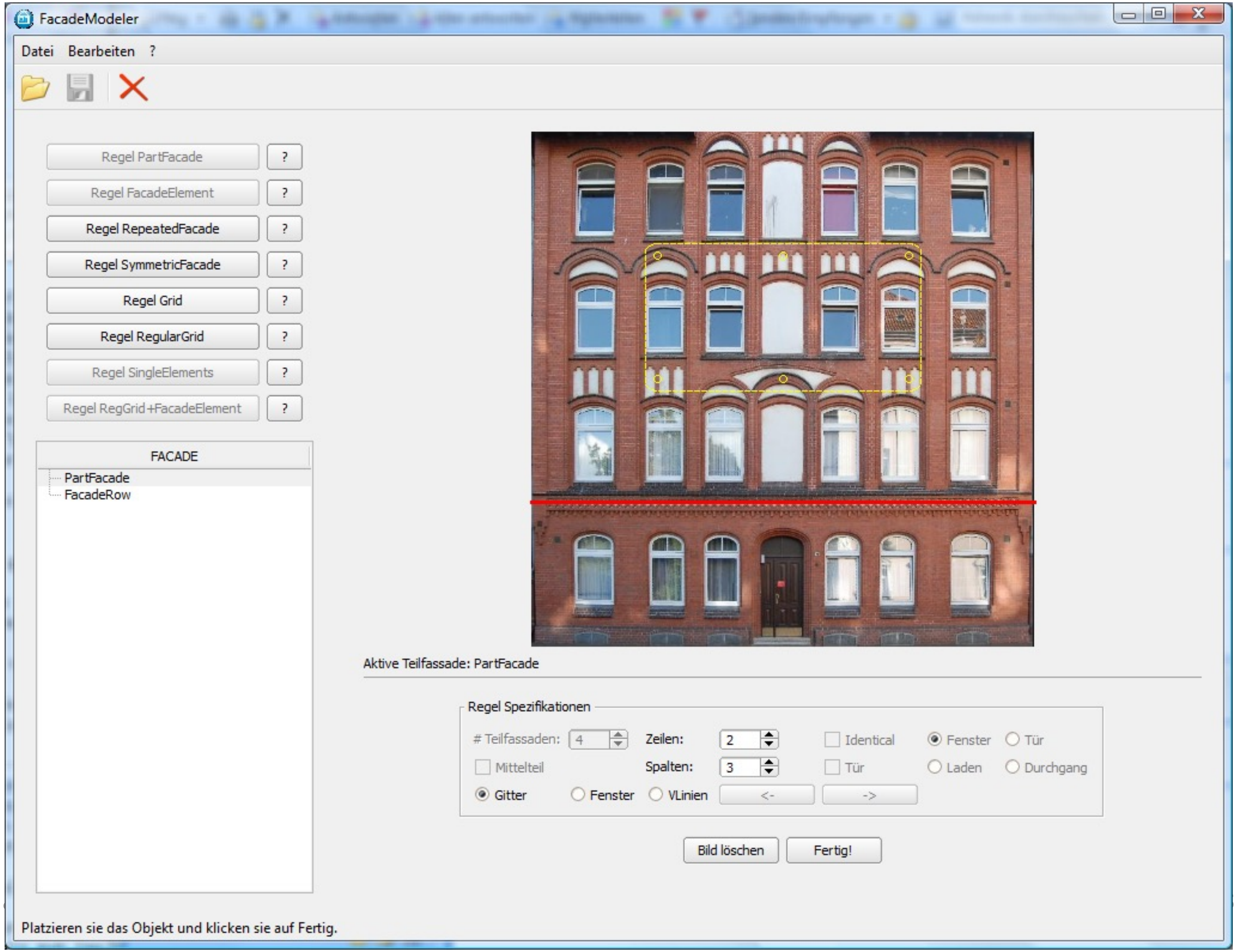
Start: manual modeling…



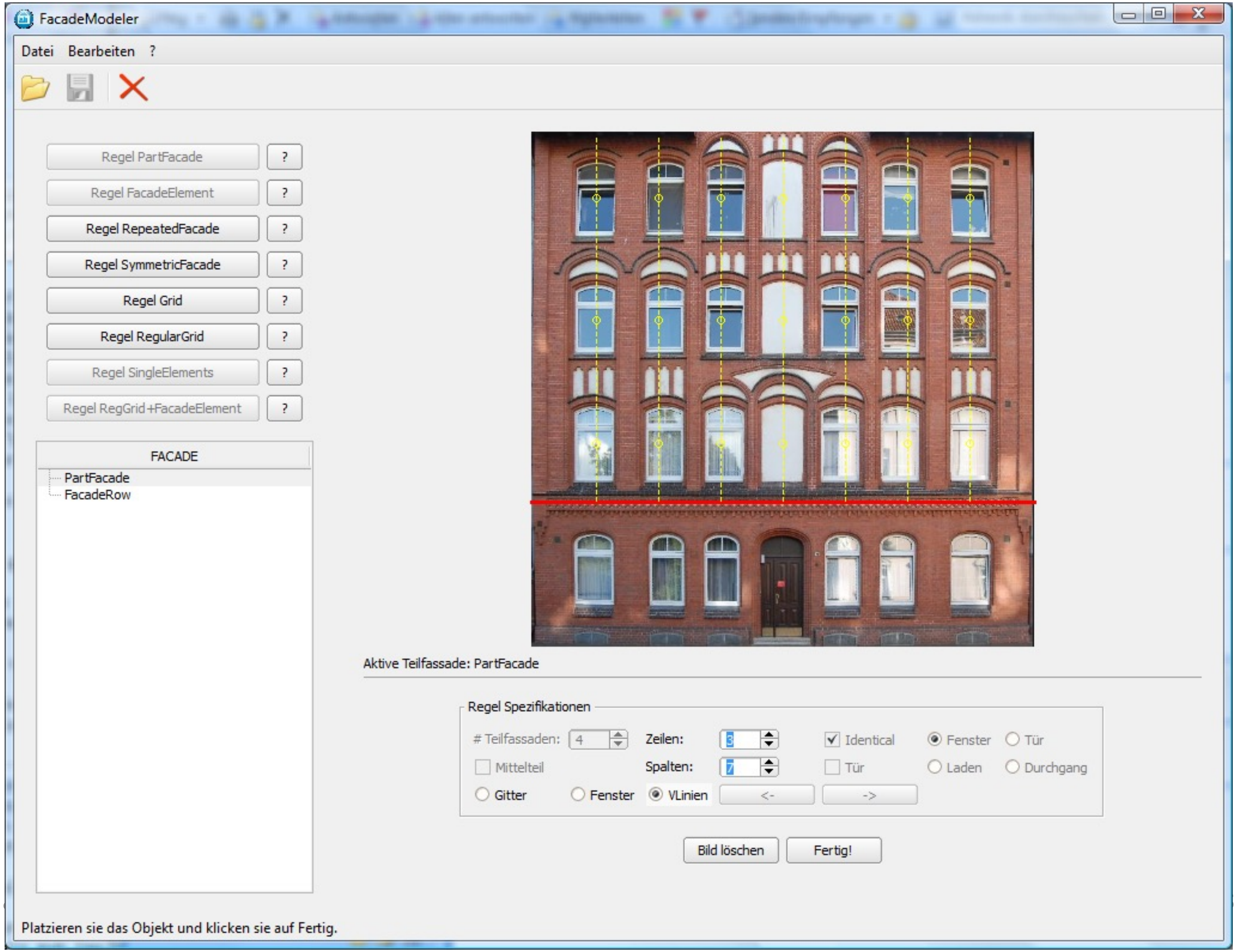
Start: manual modeling…



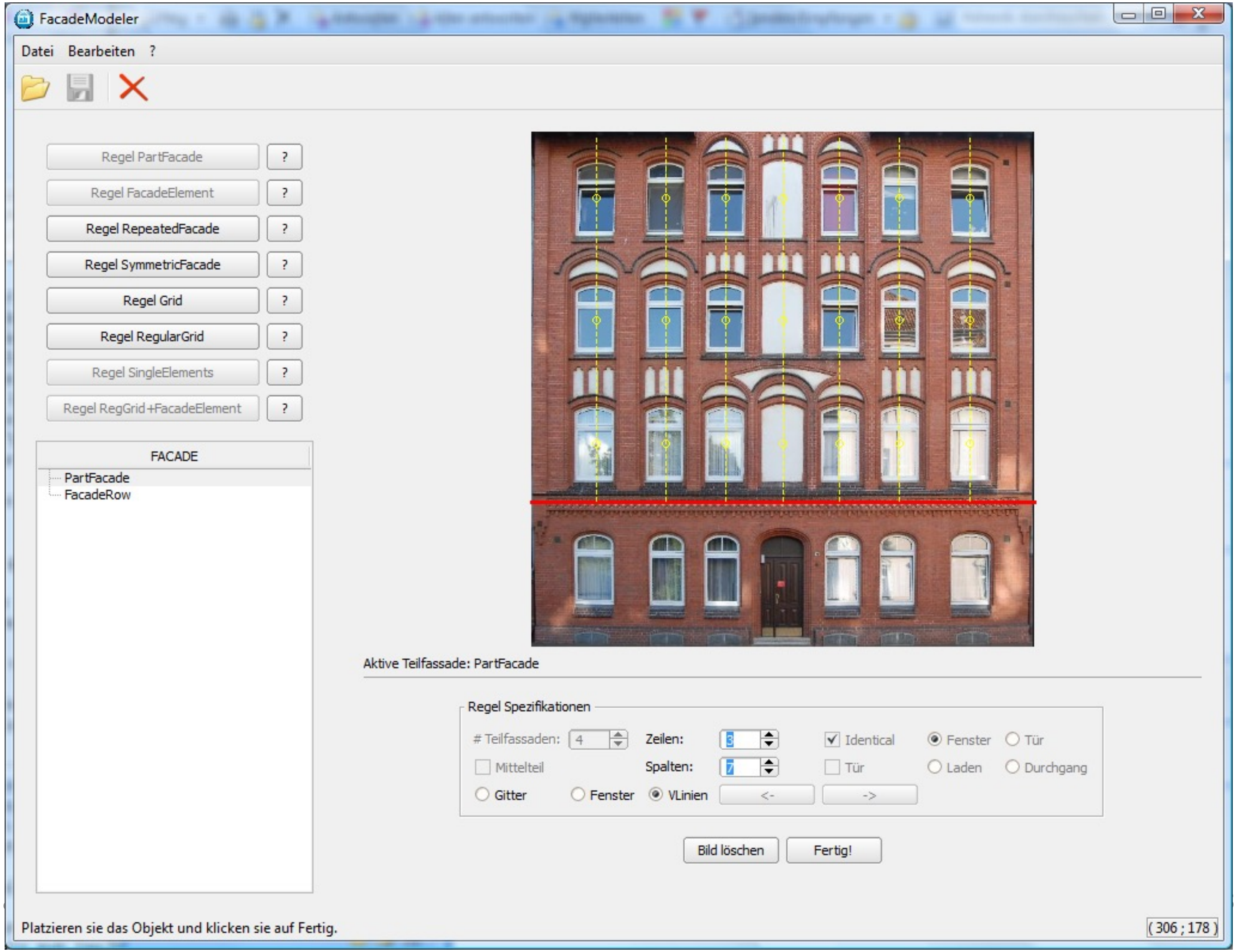
Start: manual modeling…



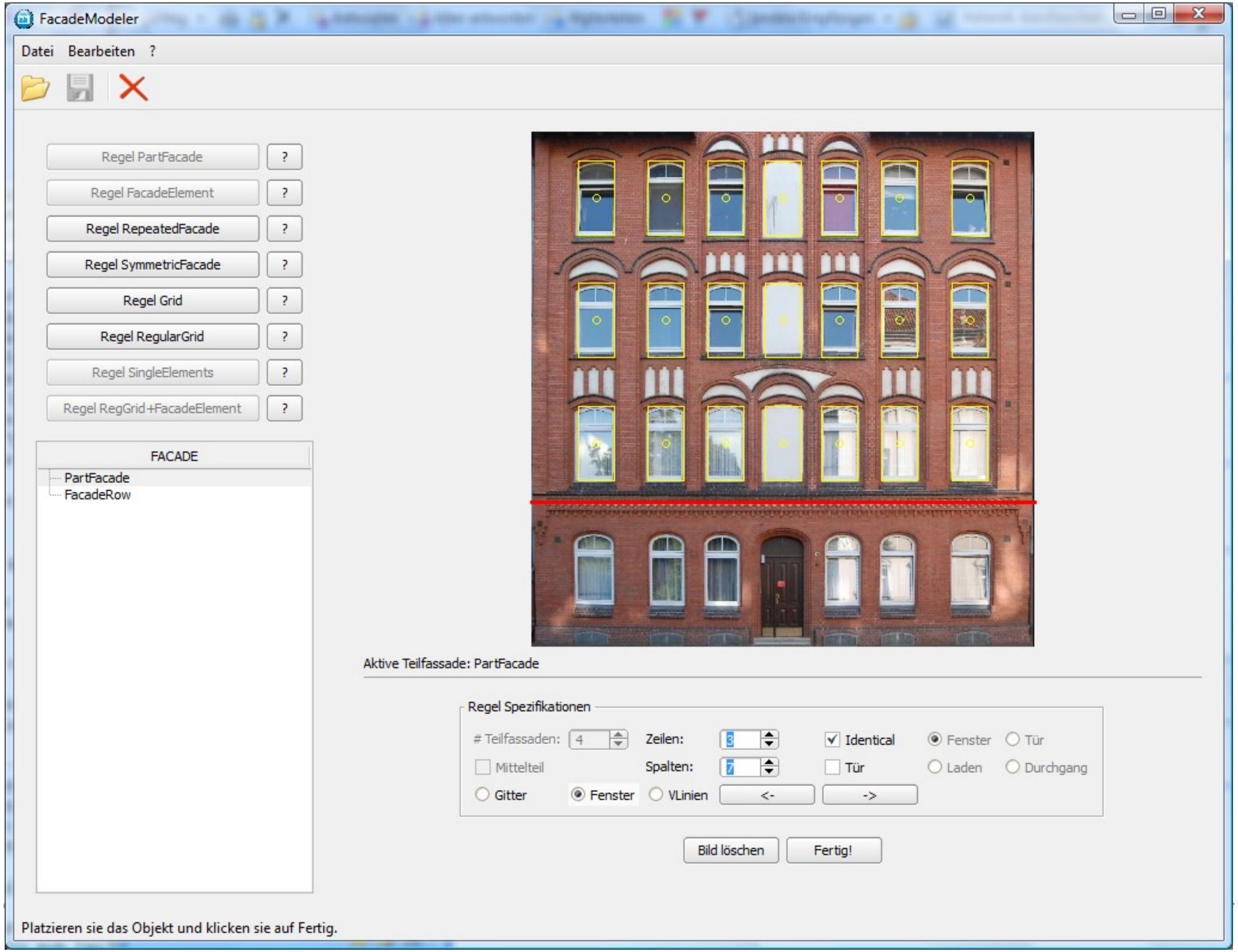
Start: manual modeling…



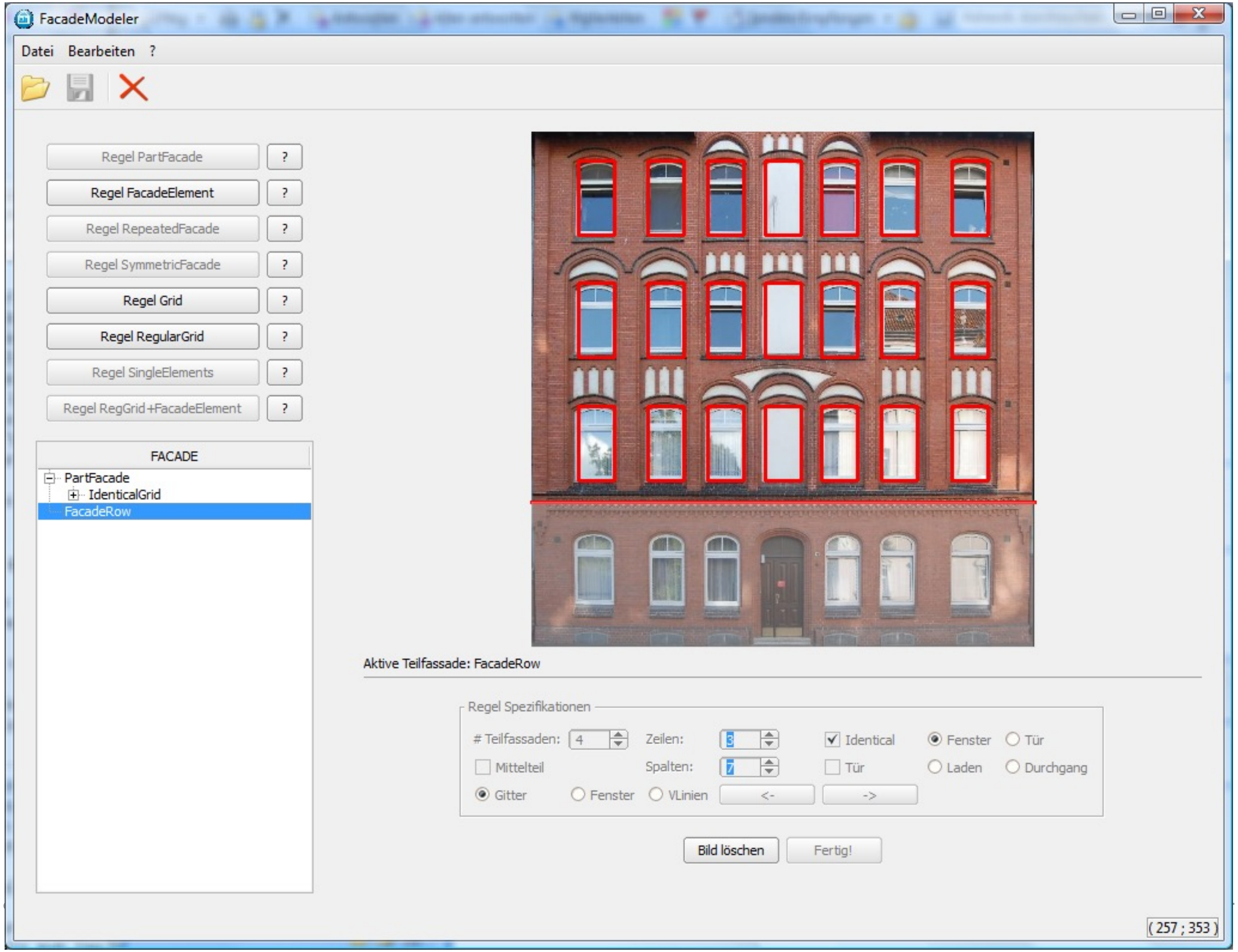
Start: manual modeling…



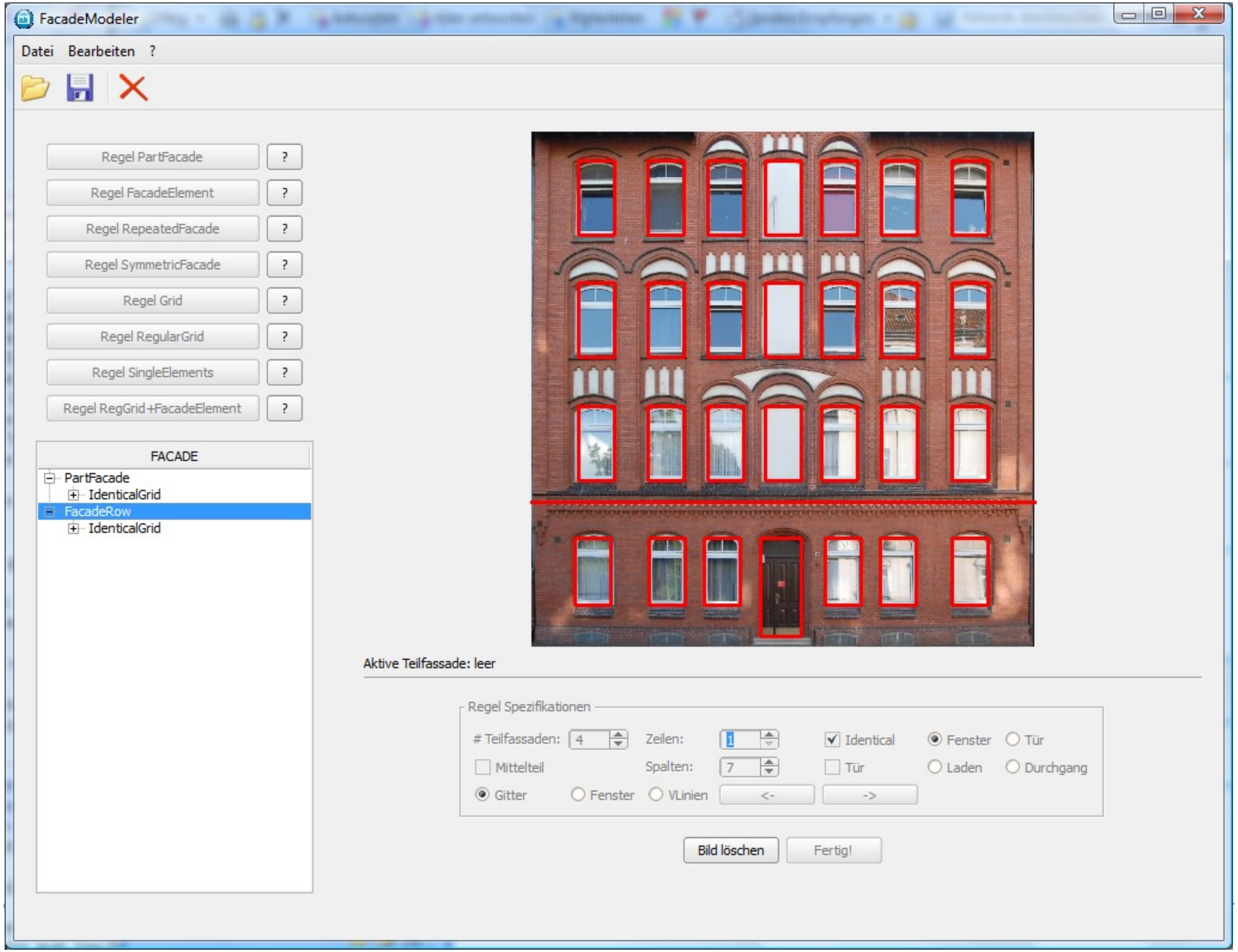
Start: manual modeling…



Start: manual modeling…



Start: manual modeling…

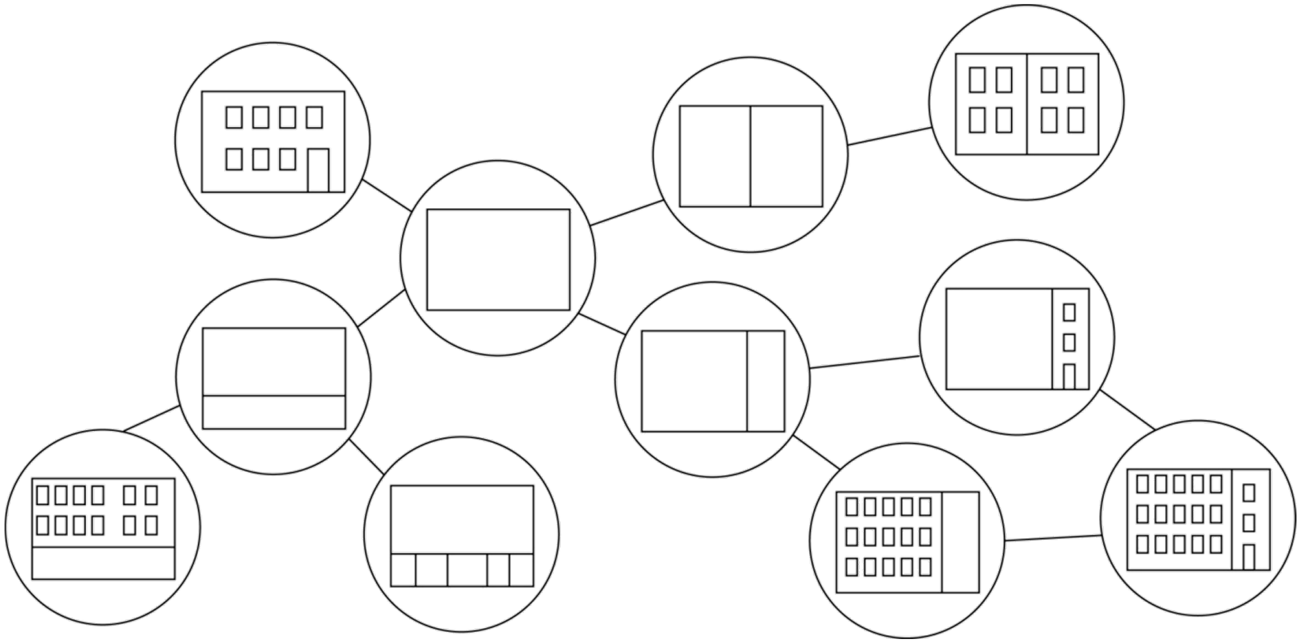


Results of a user evaluation

- Facades may be modeled in different ways!

Introduction of automatic modeling

- Using a grammar, models can be generated independently of a real scene
- Automation of the derivation process
- Reversible jump Markov chain Monte Carlo simulates drawing from the distribution of all facades using a Markov chain

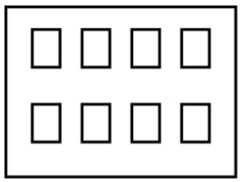


Example: constant dimension

Markov chain Monte Carlo (variable dimension)

- Automatic derivation of a façade model according to the grammar
- States of the Markov chain are the derivation trees
- Models may have varying dimensions

Width, height



Width, height, number of rows and columns, grid position, grid spacing, window width and height

Acceptance probability

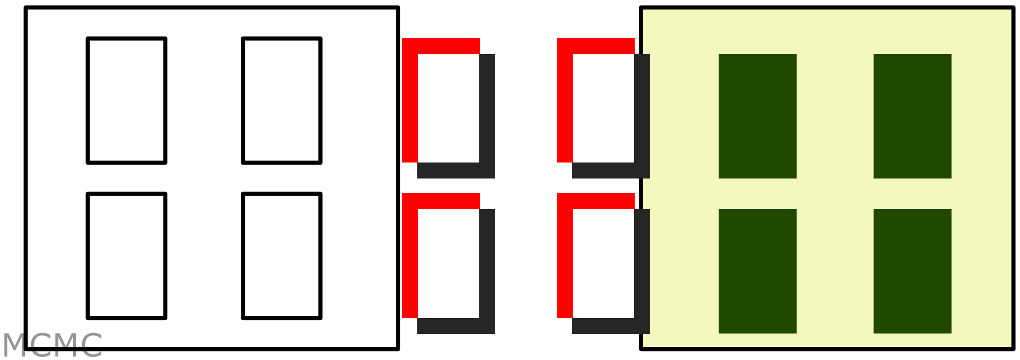
- The rating of the solution (score function) consists of two components:
  - Likelihood: how probable is the data, given the model
  - Prior: how probable is the model

$$p(\theta_t \mid D) \propto p(D \mid \theta_t) \cdot p(\theta_t)$$

Counting the number of wrong pixels

- Computation of a "cluster image" from scan- and image data
  - Windows, doors and walls differ in depth and color

- Cluster image and model are compared
  - The number of differing pixels is assessed



Likelihood

- The probability $p(D \mid \theta)$ is needed
  - To determine this, one would need lots of training data (which would be too expensive)

- Alternative method (Fua and Hanson 1989)

$$L(D \mid \text{without model}) = A \cdot \log_2 N_C$$

$$L(D \mid \theta) = \underbrace{-(n_0 \cdot \log_2 \frac{n_0}{A} + n_1 \cdot \log_2 \frac{n_1}{A})}_{\text{Identification of outliers (entropy * A)}} + \underbrace{n_0 \cdot \log_2 N_C}_{\text{Modeling of the outliers}}$$

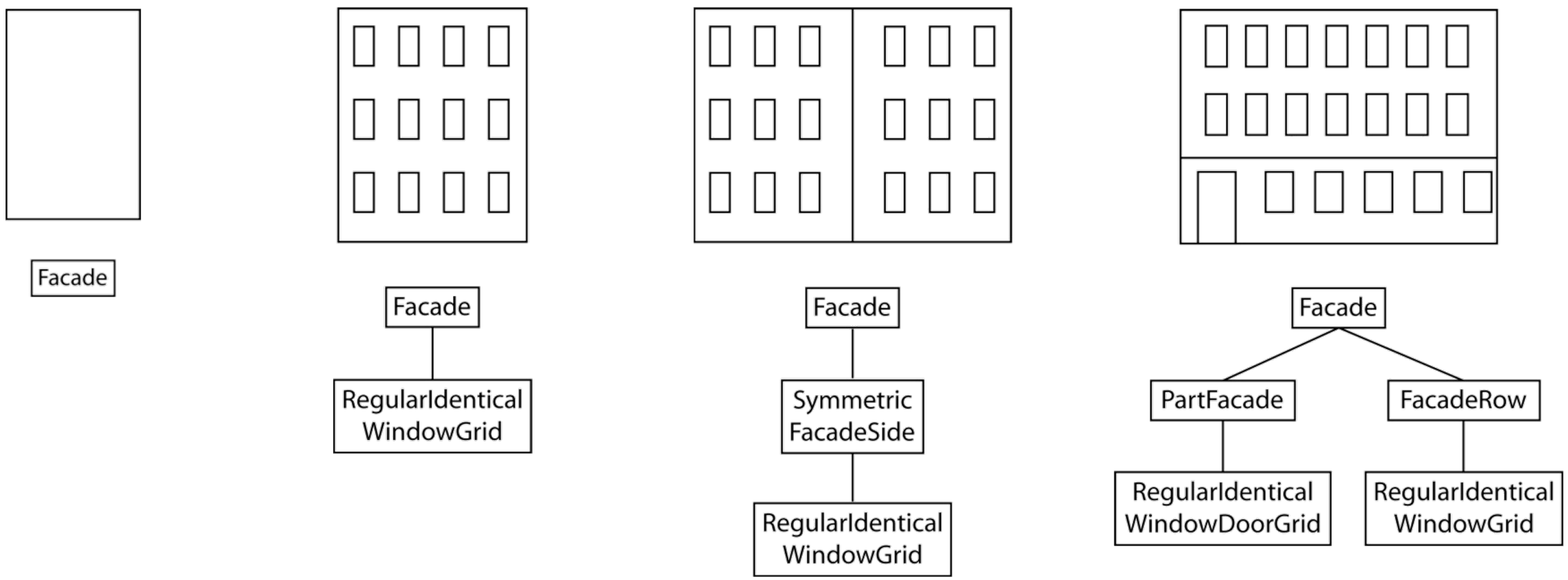
N_C #cluster labels
A #pixels
n_0 #outliers
n_1 #model pixels

Identification of outliers (entropy * A)

Modeling of the outliers

Prior – Model complexity

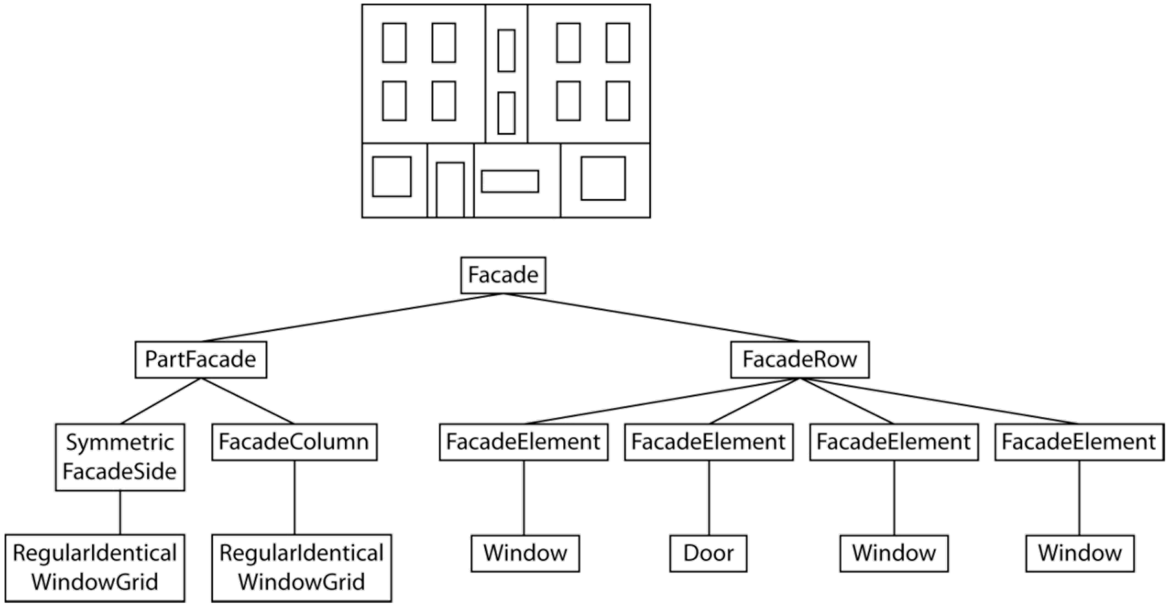
- Finding $p(\theta)$ is too costly (needs training data)
- → Assessment is done based on symbols and parameters
  - This can be computed directly from the derivation tree
- Yields an intuitive assessment of the model



G=4,32 G=120,64 G=124,97 G=325,61

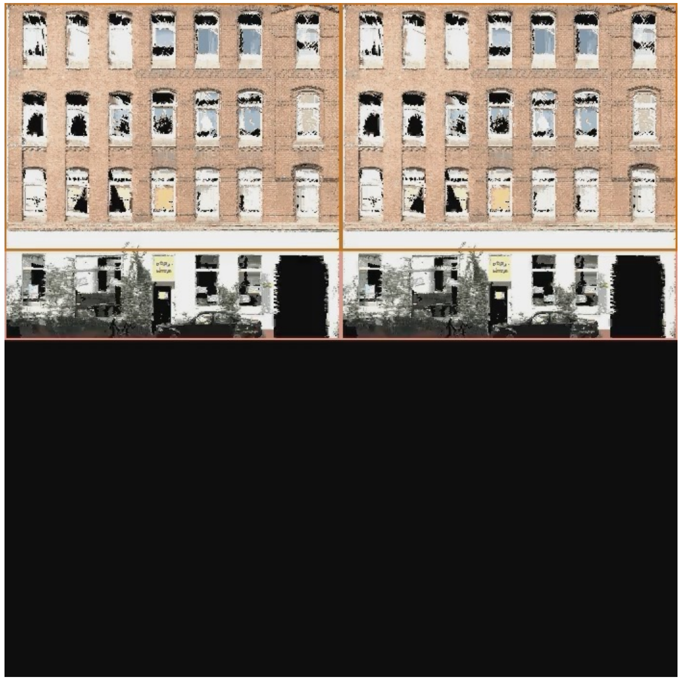
Prior – Model complexity

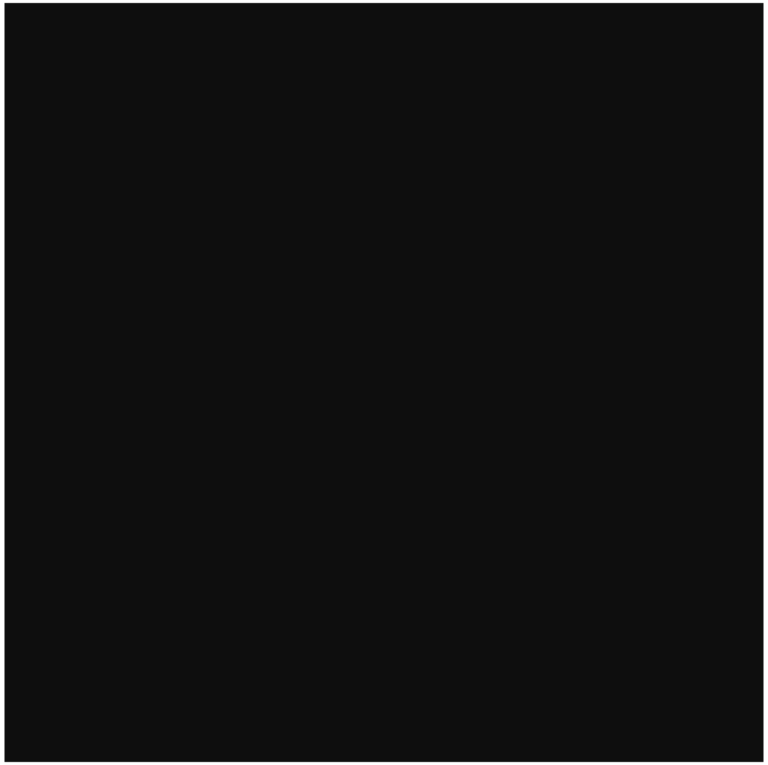
- Finding $p(\theta)$ is too costly (needs training data)
- → Assessment is done based on symbols and parameters
  - This can be computed directly from the derivation tree
- Yields an intuitive assessment of the model



G=640,83

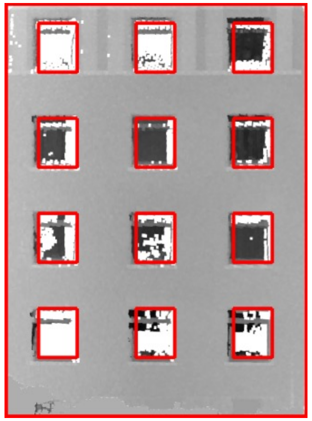
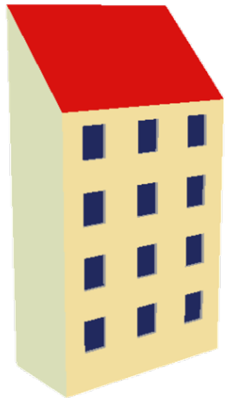
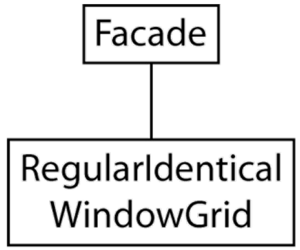
MCMC at work…



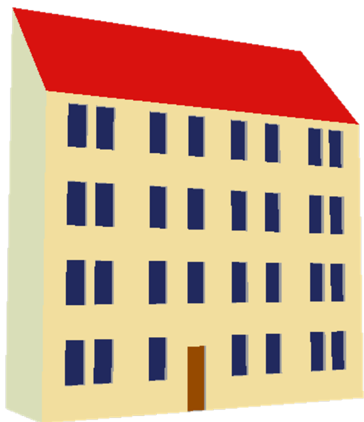
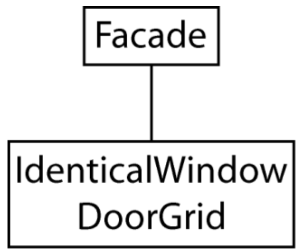


Results of the façade reconstruction I

- Reconstruction from scan data only

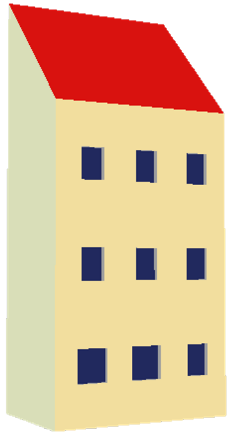




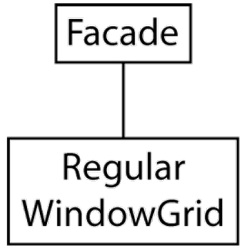
- Reconstruction from scan and image data

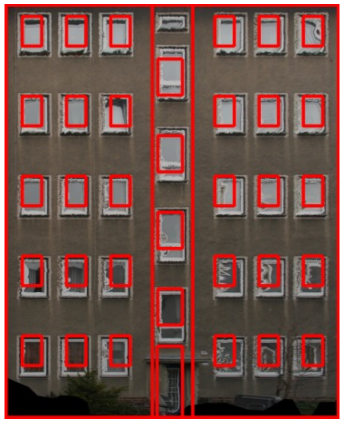
Results of the façade reconstruction II

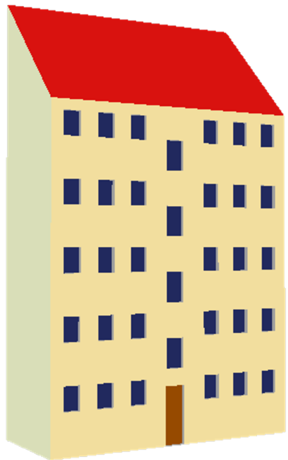
- Different windows

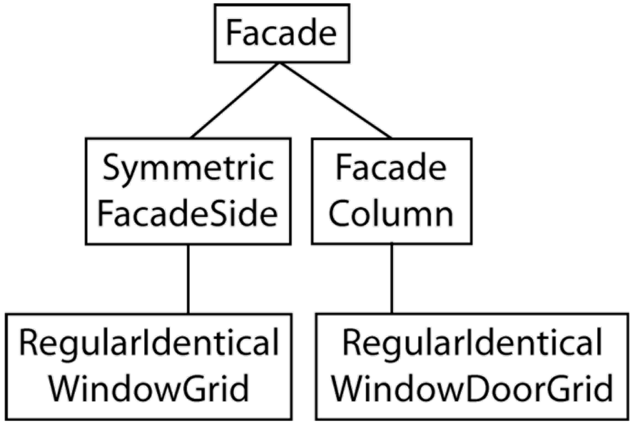




- Symmetric structure

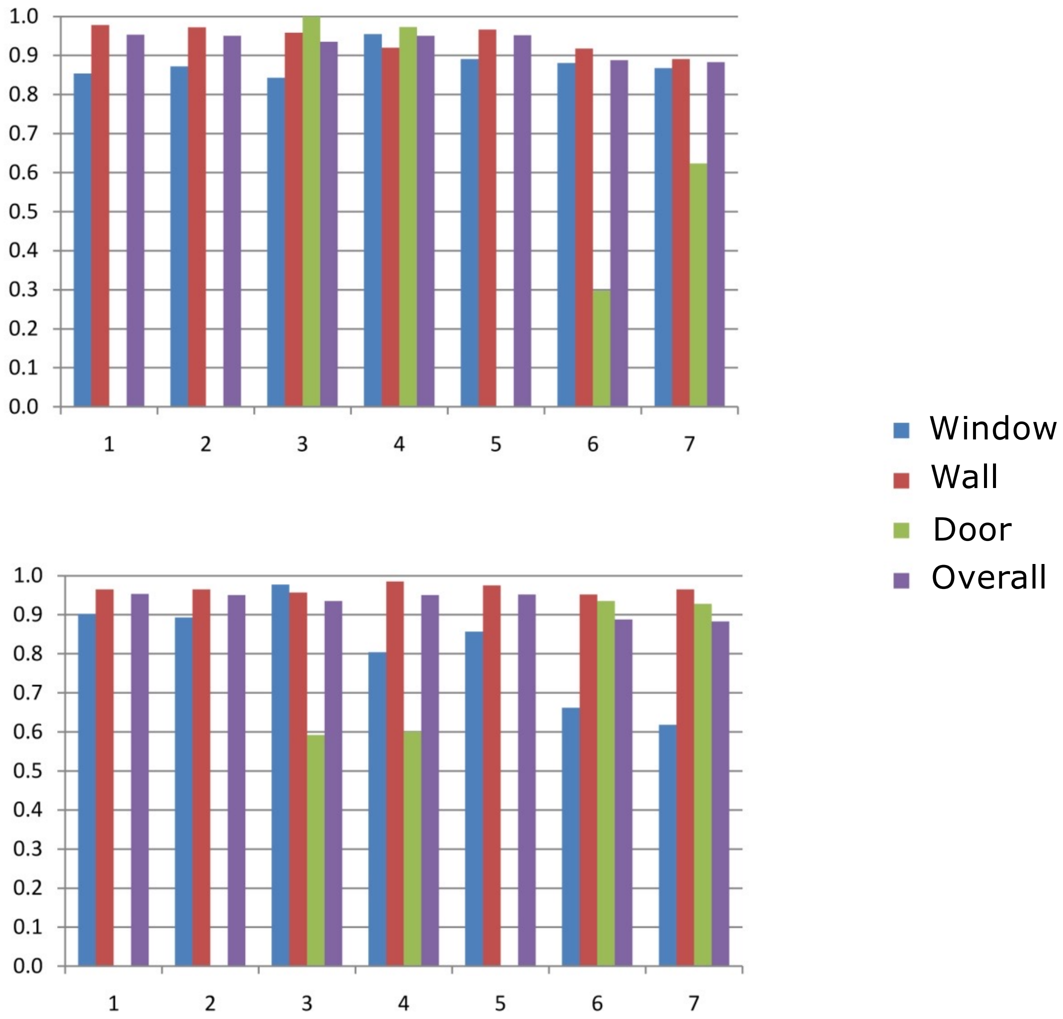






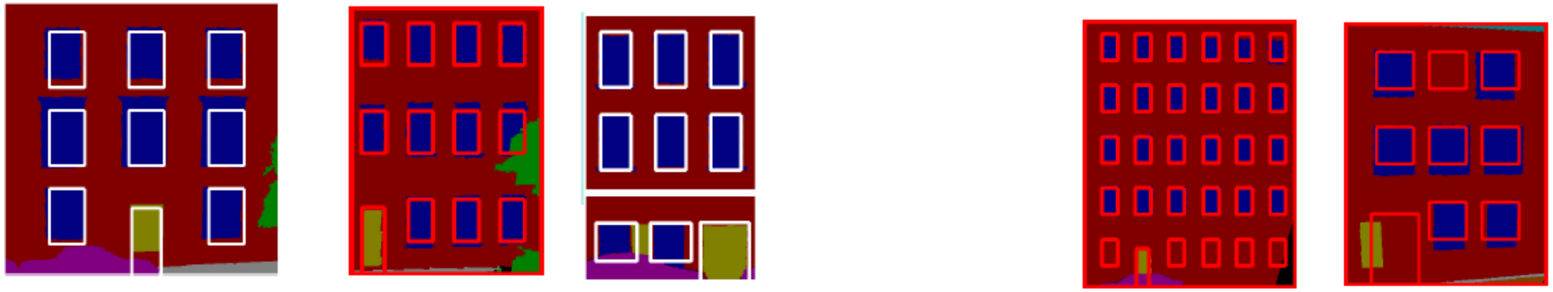
Results of the façade reconstruction III

- Analysis of the correctness and completeness of the reconstruction
  - Correctness
  - Completeness

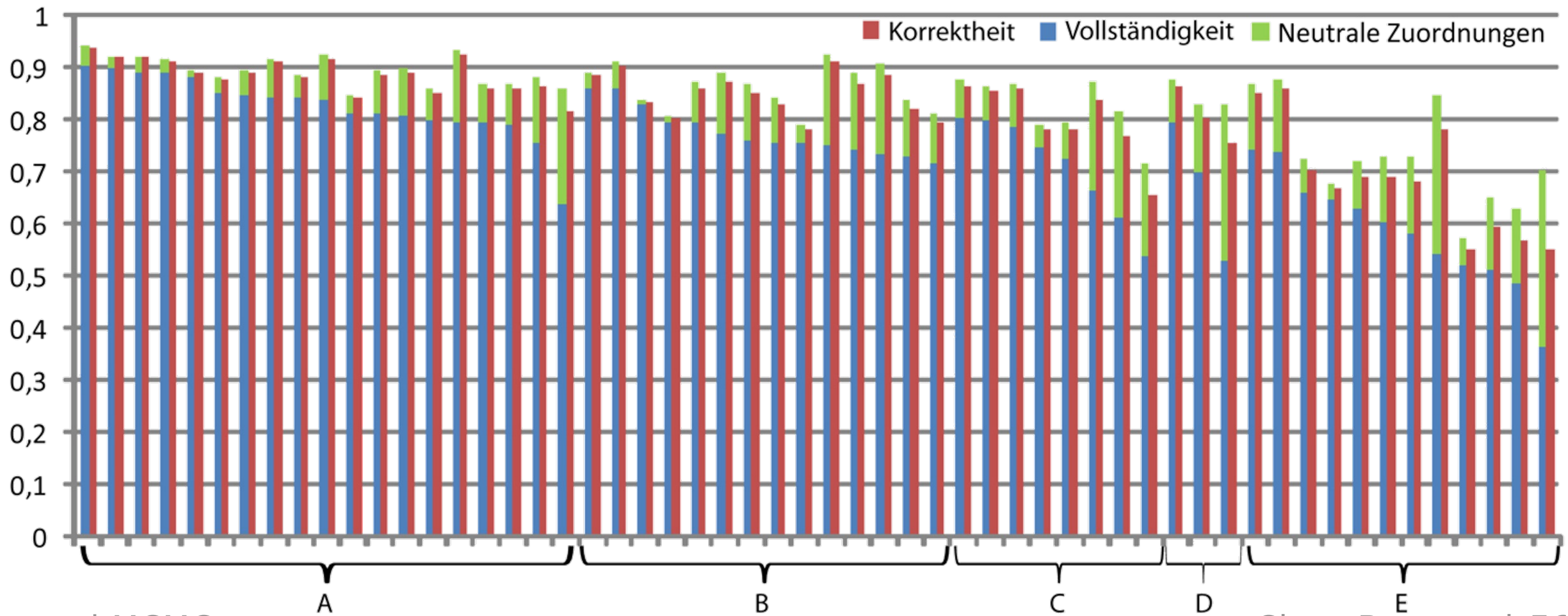


Evaluation using data of the eTrims database

- Manually segmented façade images using 8 classes

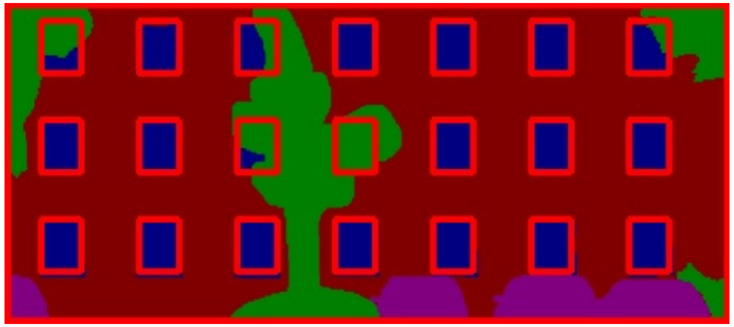


- Correctness and completeness of all reconstructions

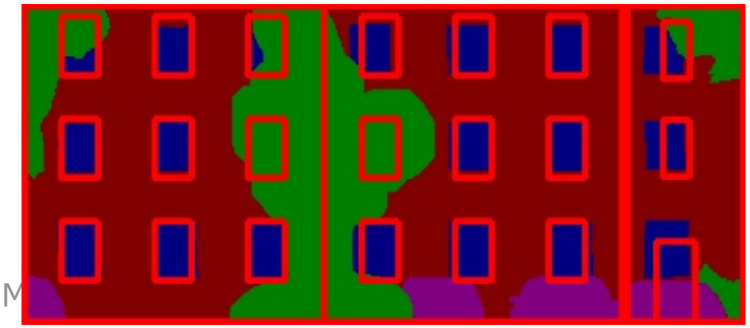


Results when occlusions are present

- Grammar allows to reconstruct occluded areas
- Test with images which were synthetically modified
- Correct reconstruction even if the tree is expanded up to 70 cm in each direction



- Results change for larger values, e.g. when expanded by 170 cm



Example 2: Building reconstruction
Hai Huang et al. (2013)

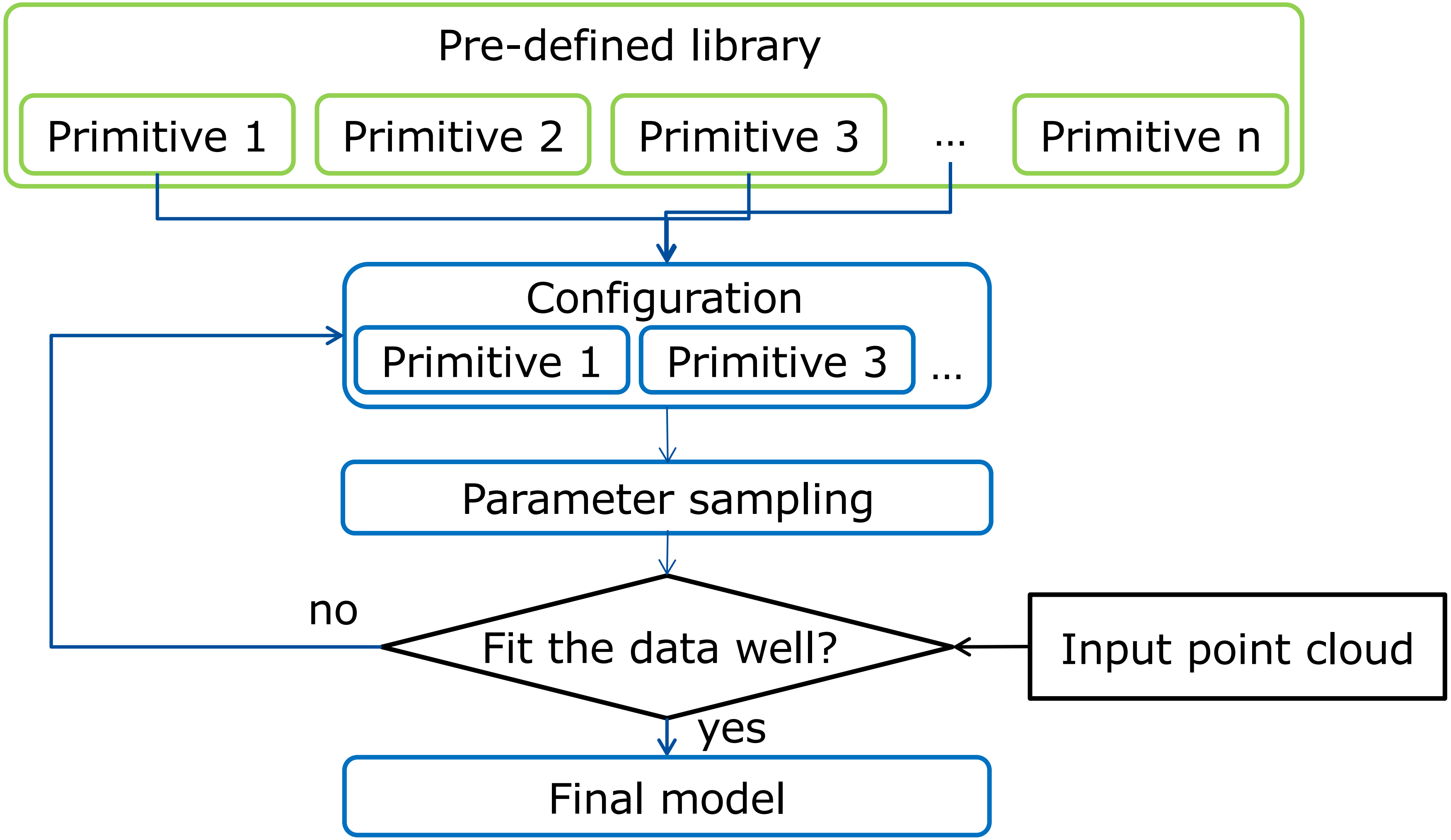
Introduction

- Airborne laser scanning data of urban scenes have:
  - Clutter by trees
  - Reflection from windows and waterlogged areas on the roof
  - HVAC equipment
  - Possibly low point density

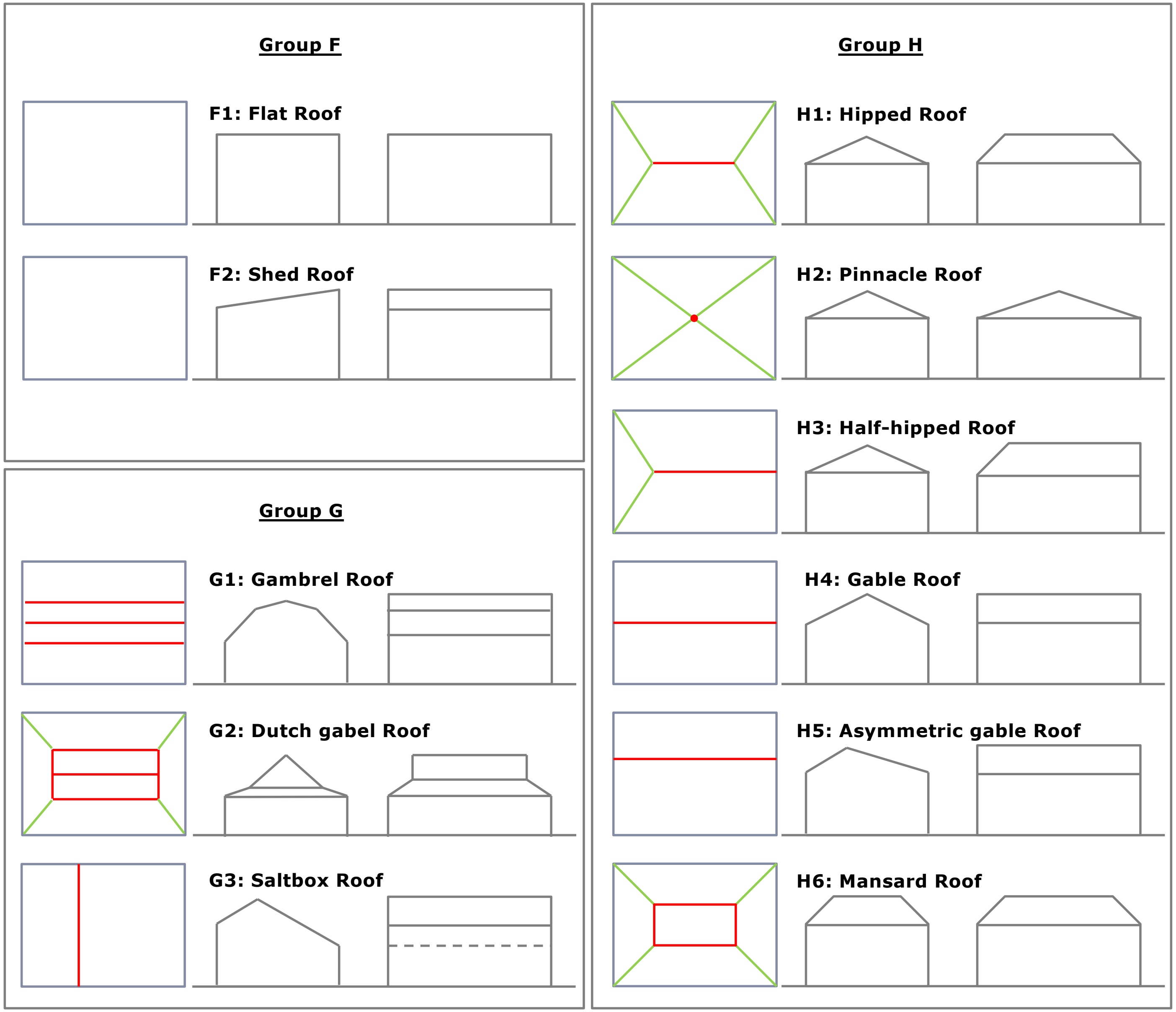



Introduction

- Top-down reconstruction via generative models

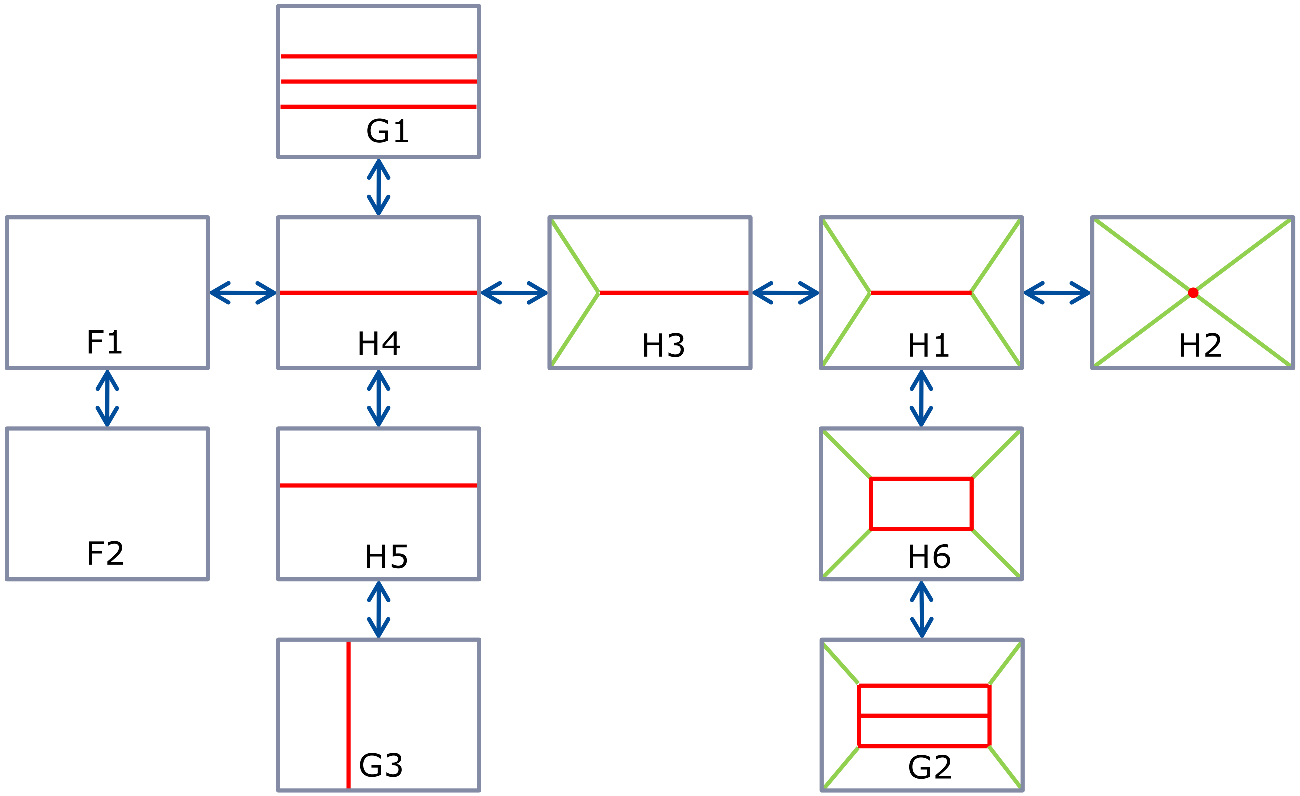


Library of Primitives

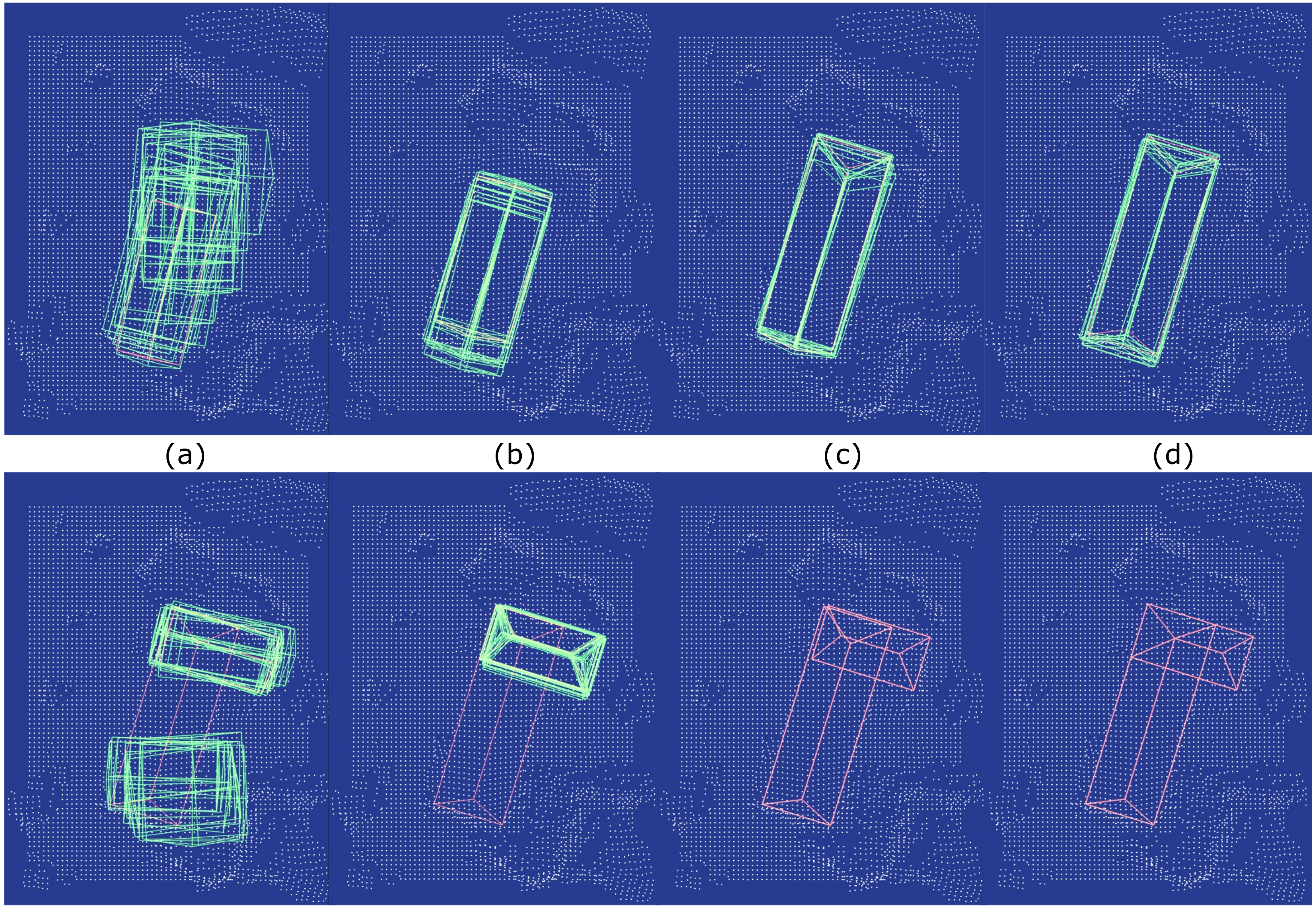


Reversible Jumps

- Reversible Jump Markov Chain Monte Carlo (RJMCMC)
  - For sparsely distributed sampling space
  - For searching in variable dimensions



Example of Searching

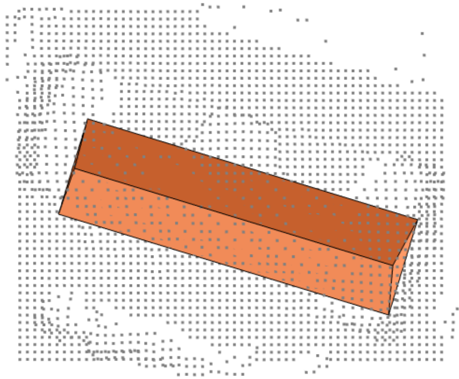


(a) (b) (c) (d) (e) (f) (g) (h)

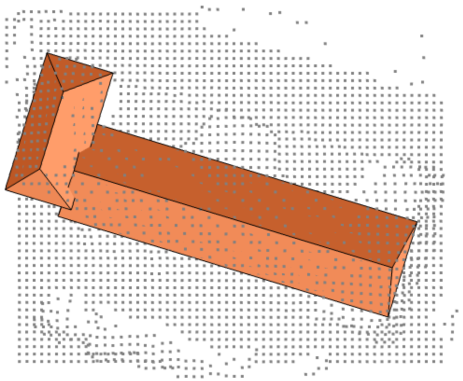
Primitive Combination

Reference image

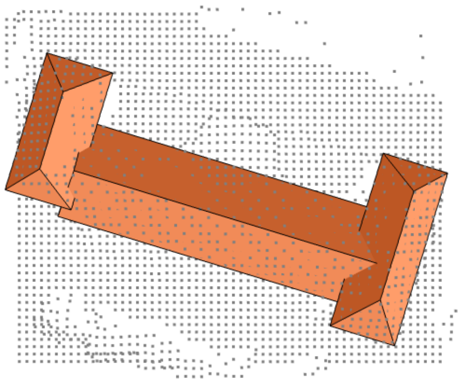
Point cloud



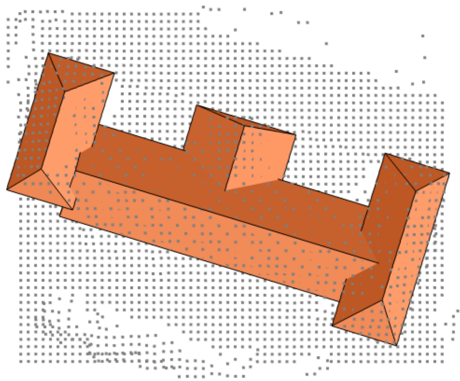
First primitive



+Second primitive

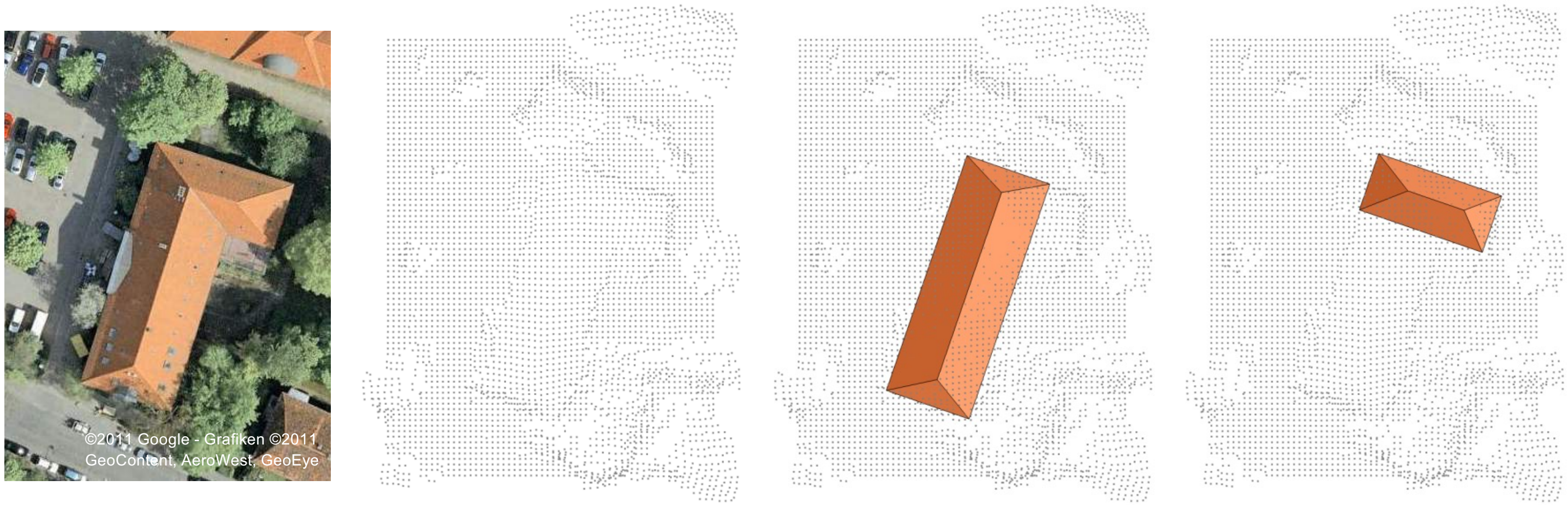


+Third primitive

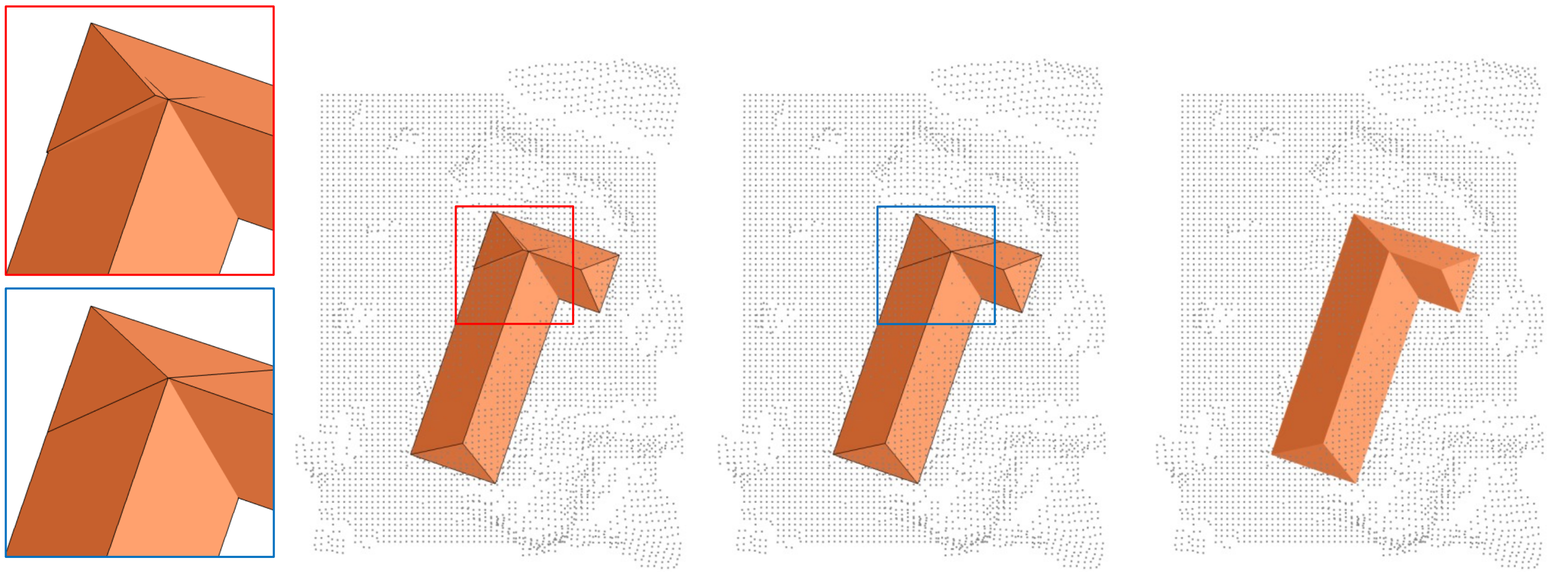


Reconstruction

Primitive Merging

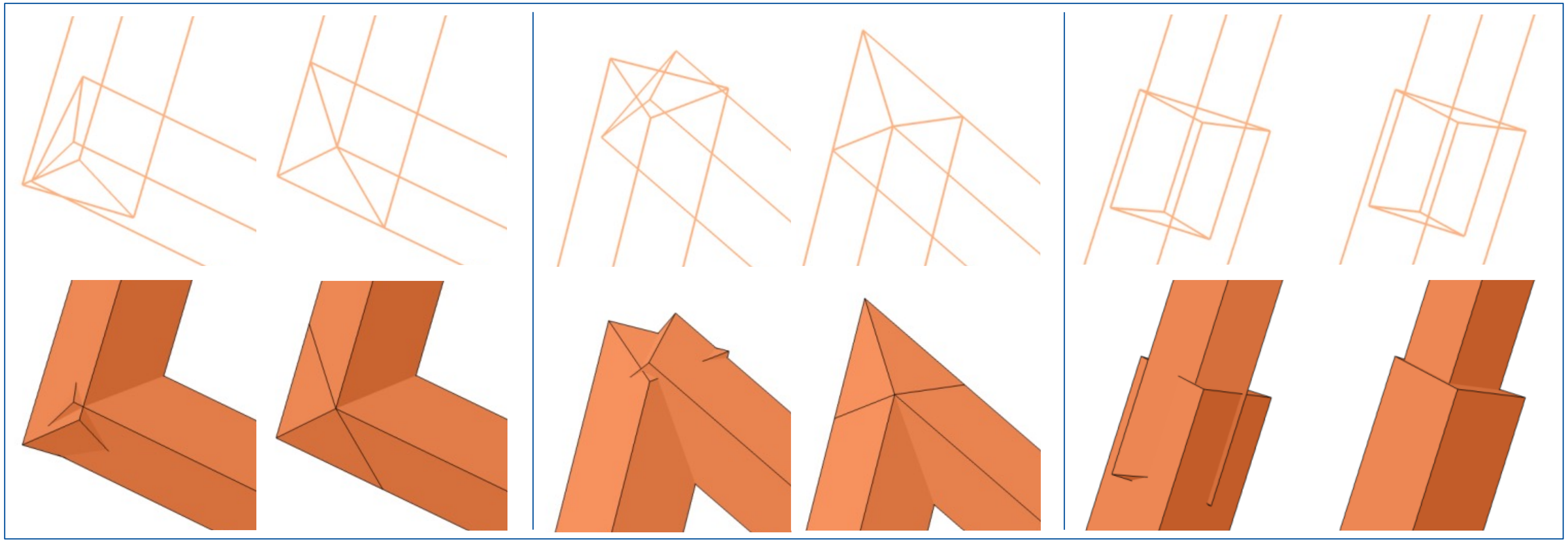


Reference image Point cloud First primitive Second primitive



Primitive merging Roof reconstruction

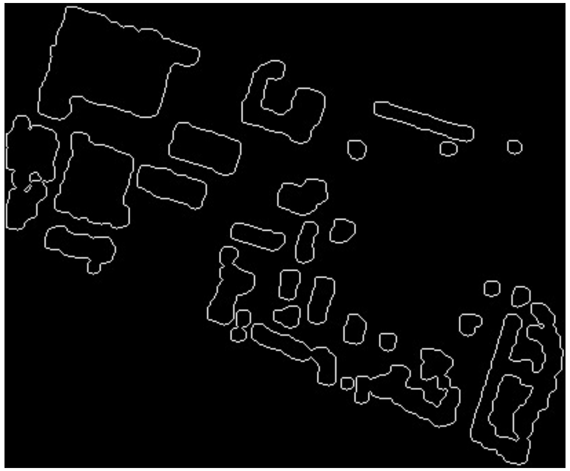
Primitive Merging

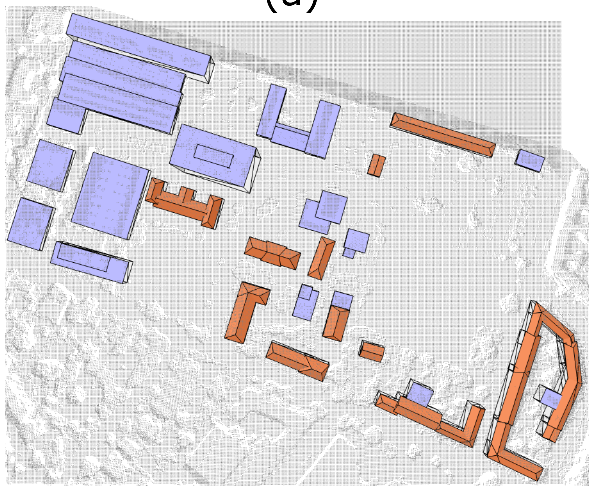


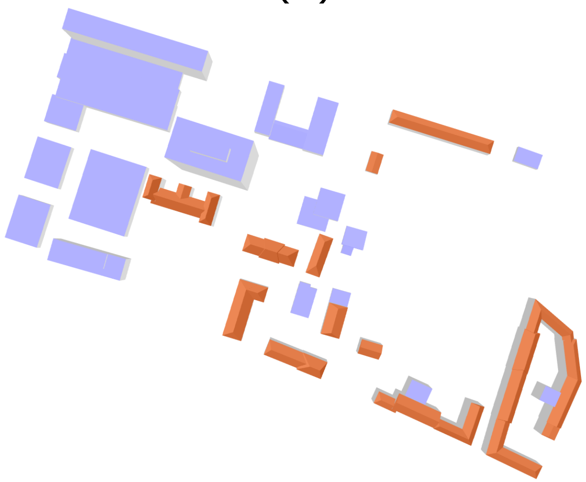
Experiment

- 1m raster, 89000 m², 21 buildings and 1 city block


(a)


(b)

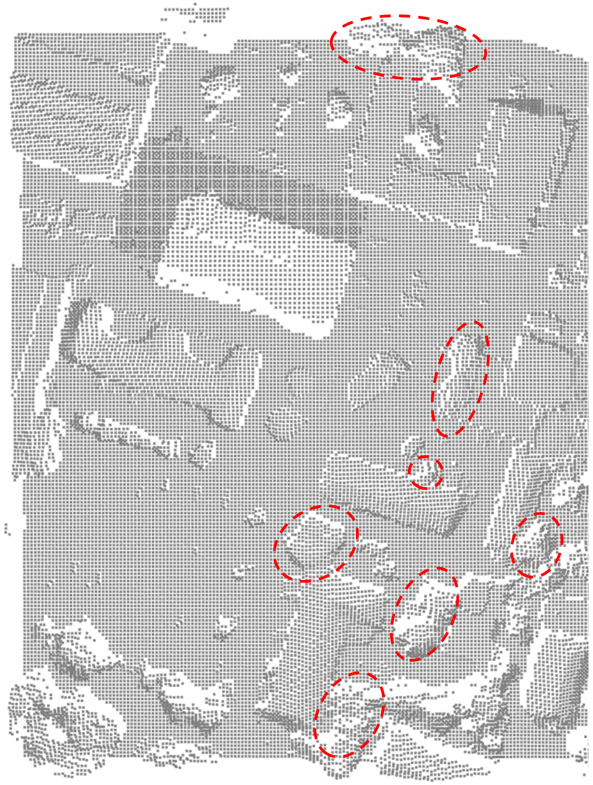

(c)


(d)

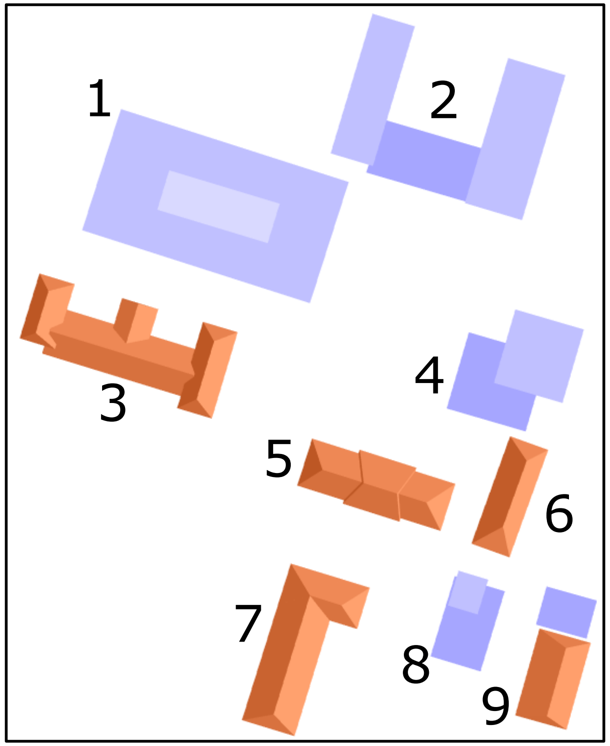
Experiment

- Comparison with the reference image

Reference image



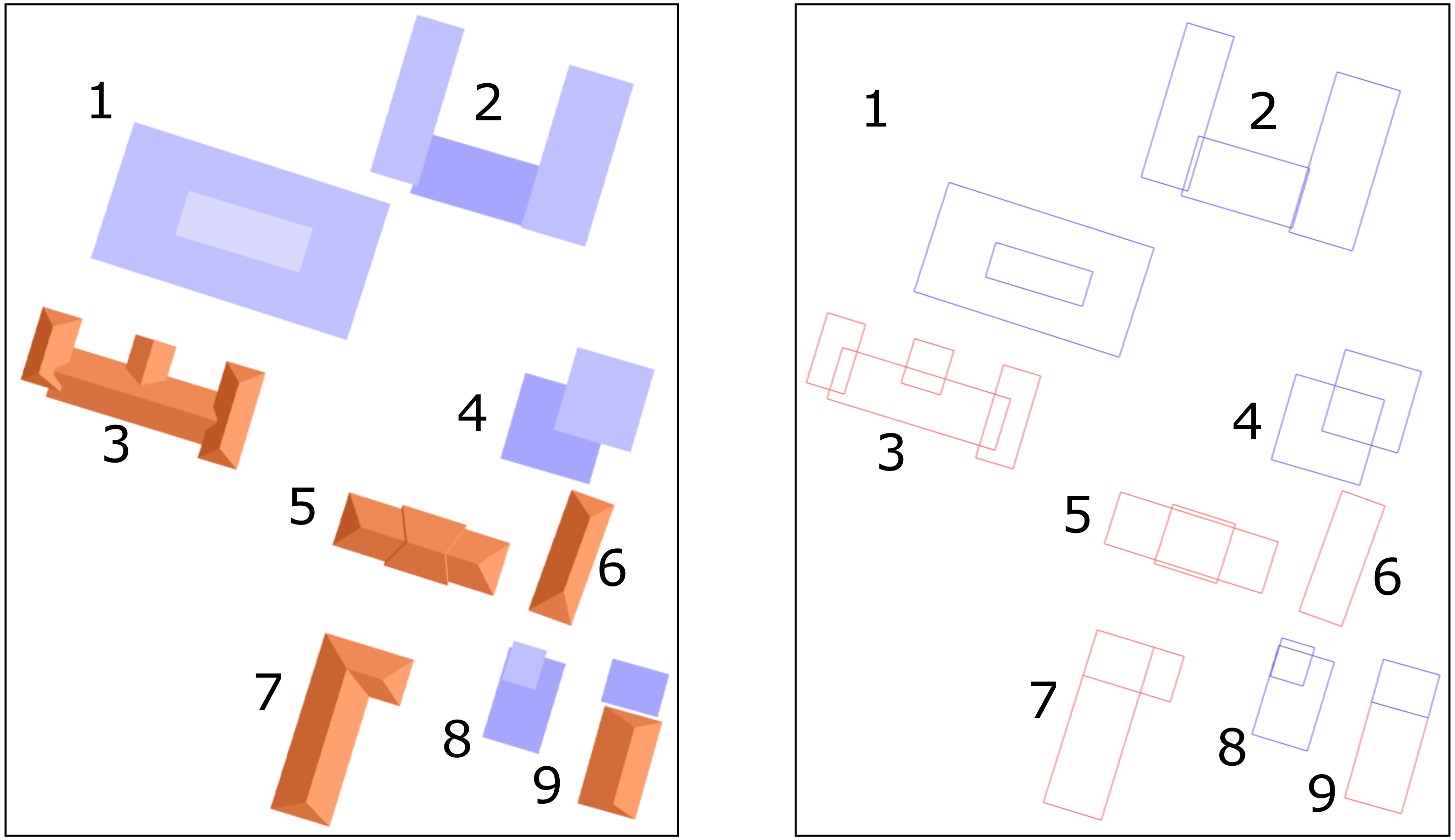
Point cloud



3D roof models

Experiment

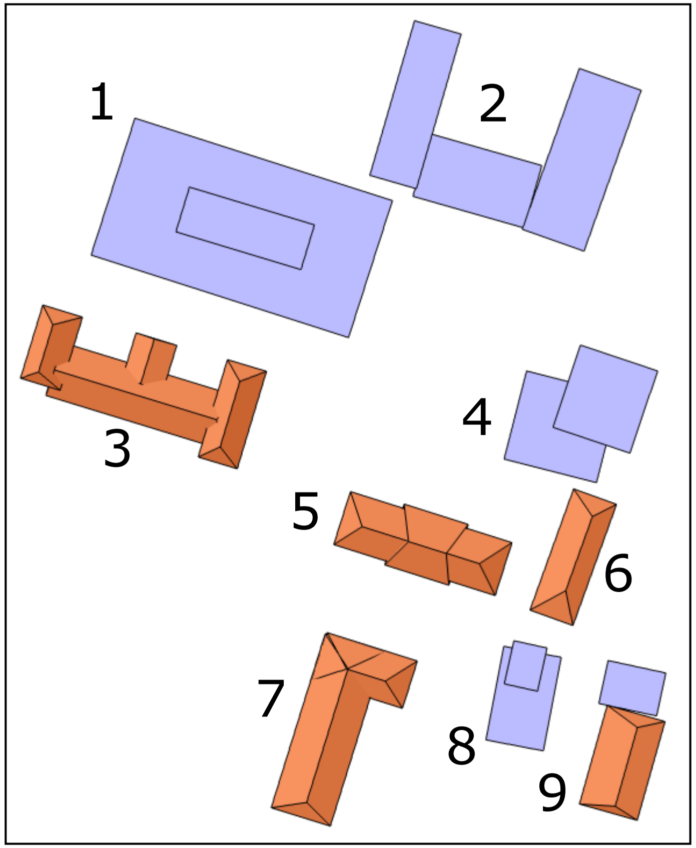
- Combination of primitives

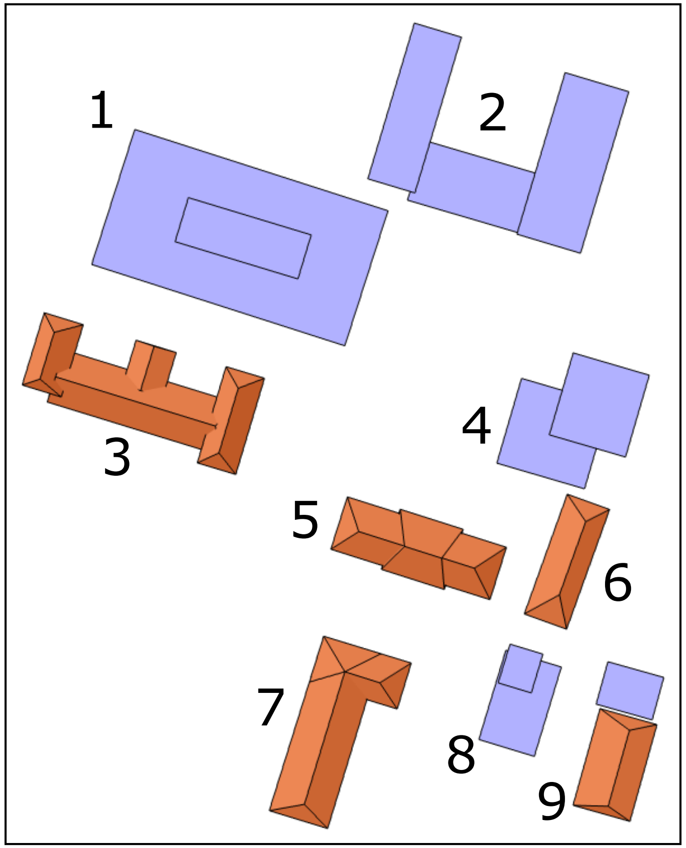


(Snapshots of 3D models)

Experiment

- Primitive merging and adjustments

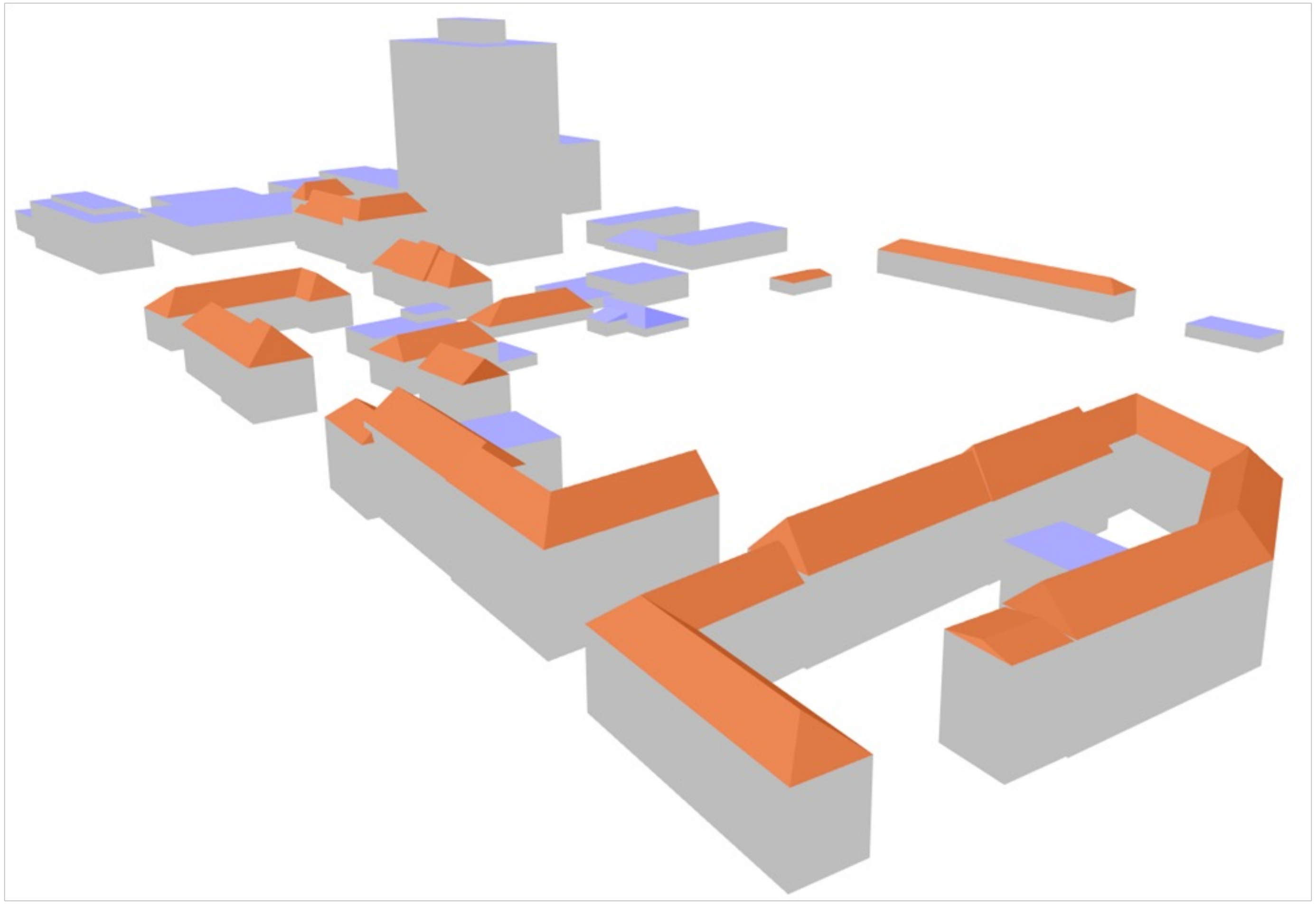




(Snapshots of 3D models)

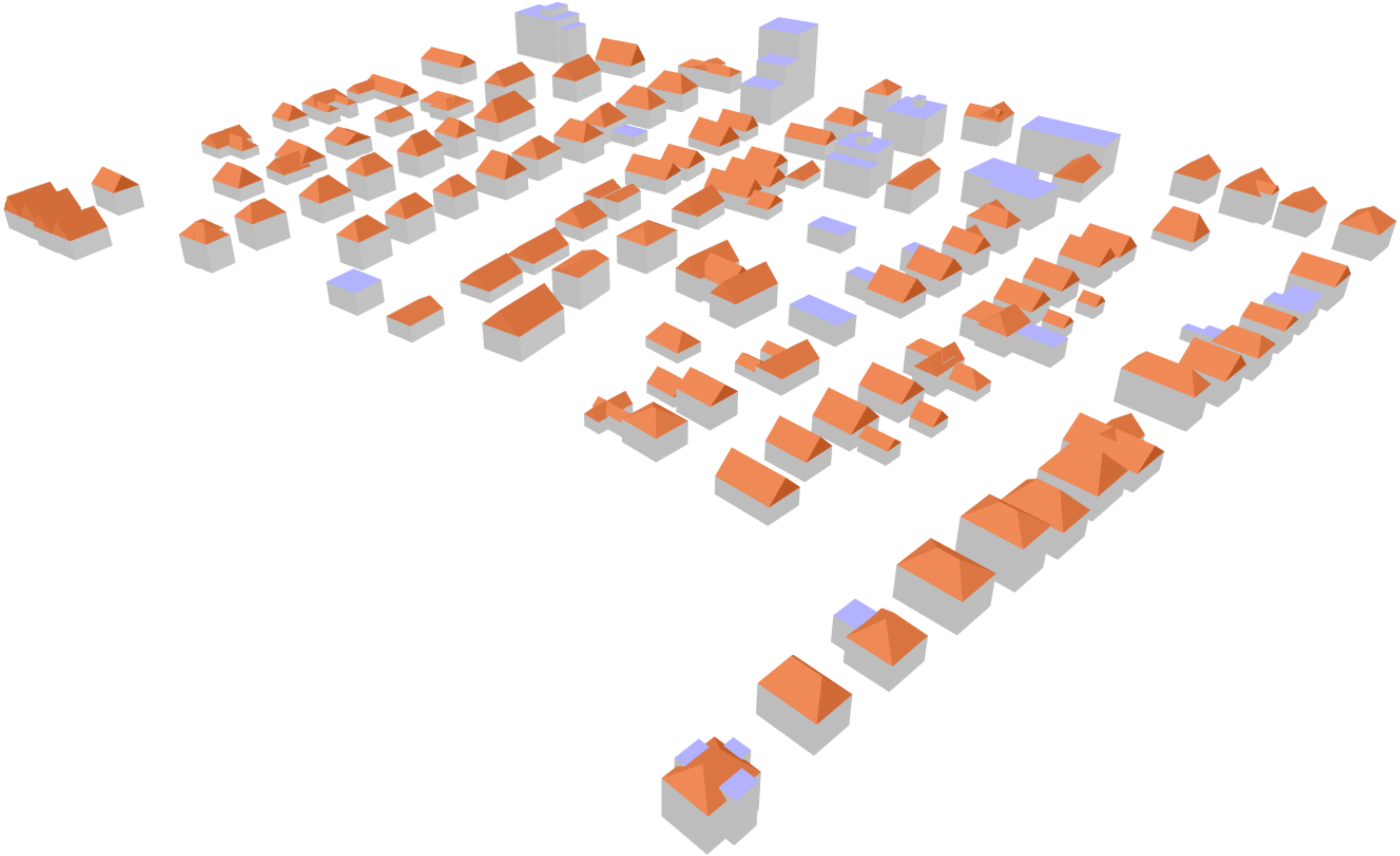
Results

- Building models



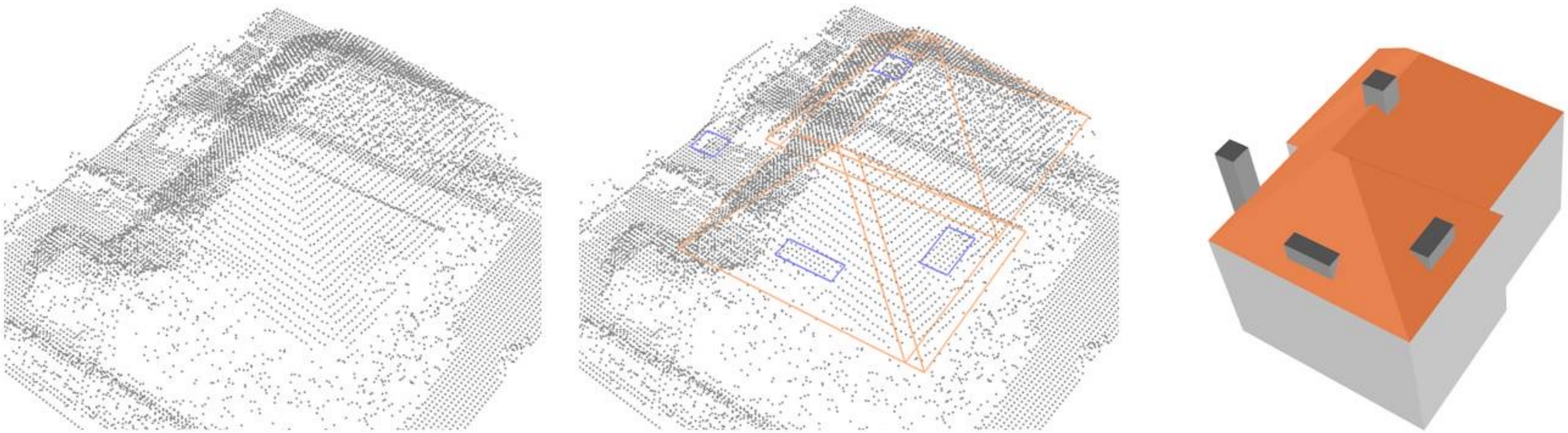
Results

- Oldenburg (0.5 m raster, 0.4 km^2, 96 buildings)



Results

- High resolution data (~0.18 m) and superstructures

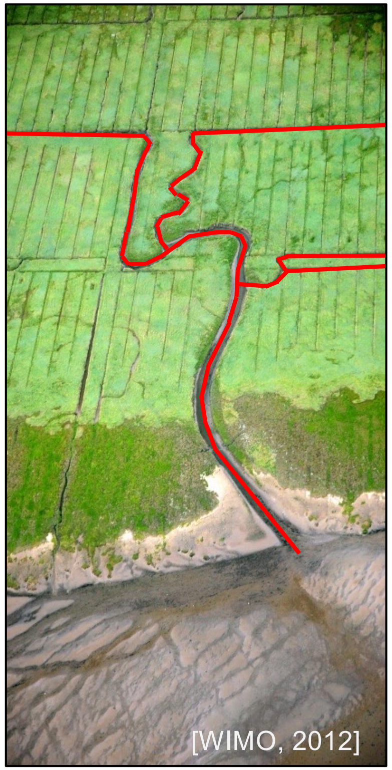


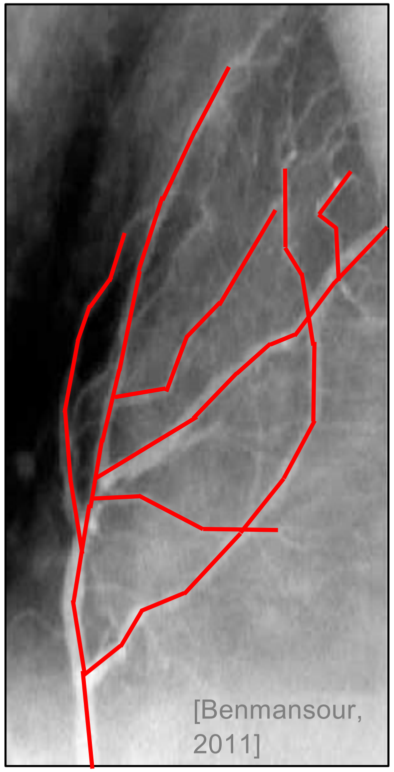
Data from ITC, Enschede

Example 3: Forest point processes for the automatic extraction of networks in raster data (Alena Schmidt)

Courtesy of Alena Schmidt (slides from her Ph.D defense),
Institute of Photogrammetry and GeoInformation

Applications



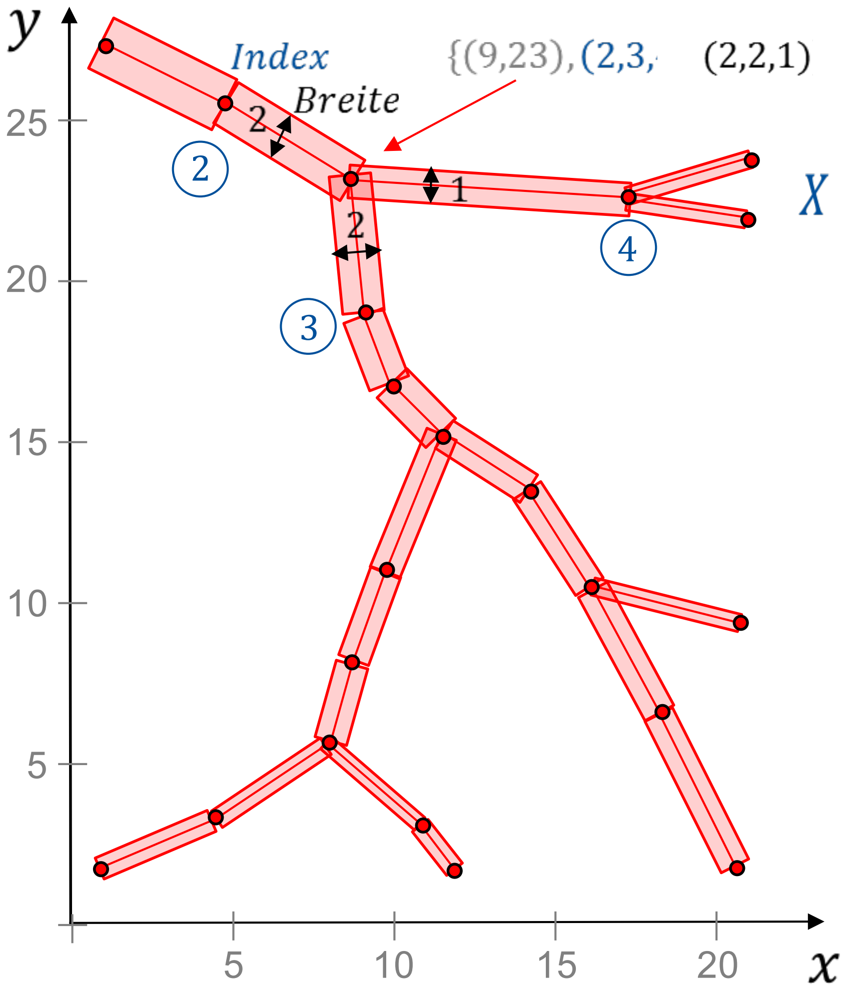


Road networks

Water networks

Vessel networks

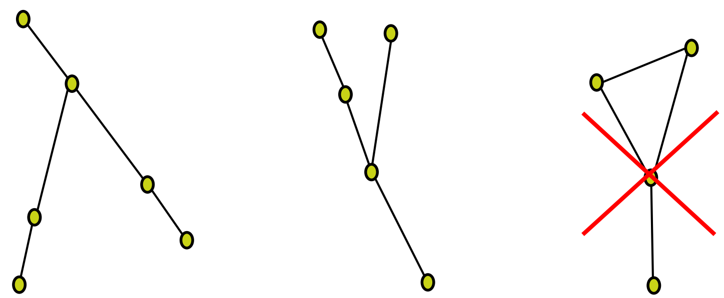
Marked point processes



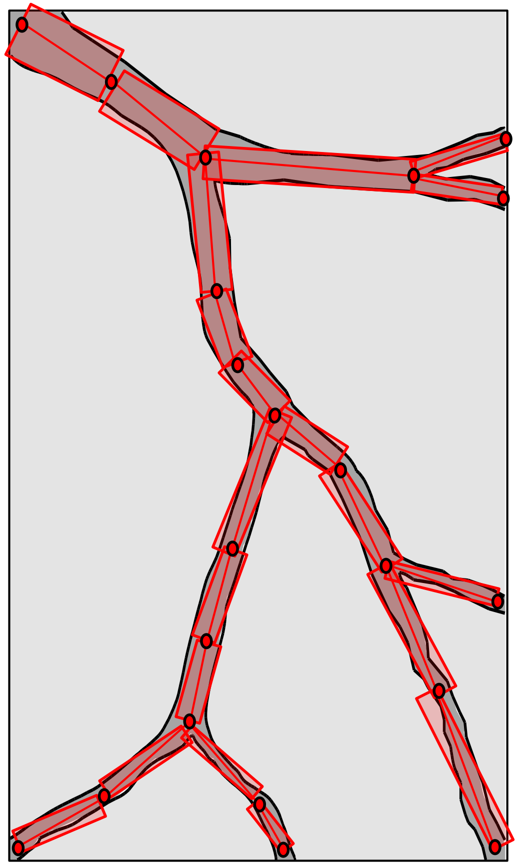
Forest Point Process

$$X = \{(p_1, m_1), (p_2, m_2), \ldots, (p_n, m_n)\}$$

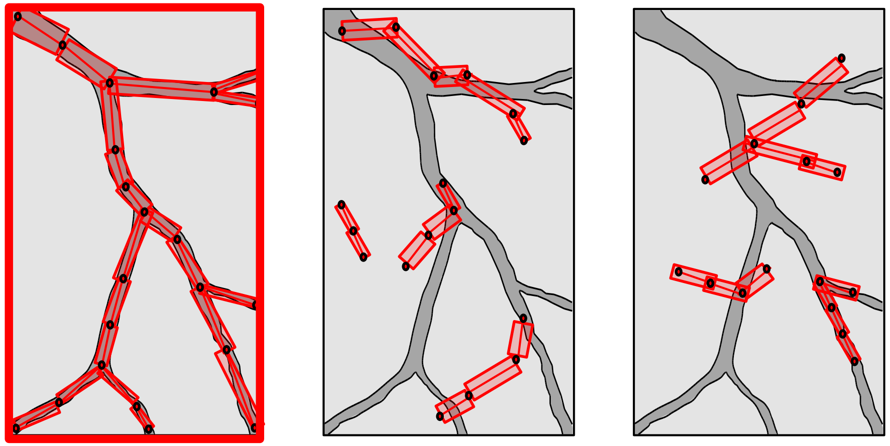
→ Graph: forest (set of trees)



Marked point processes

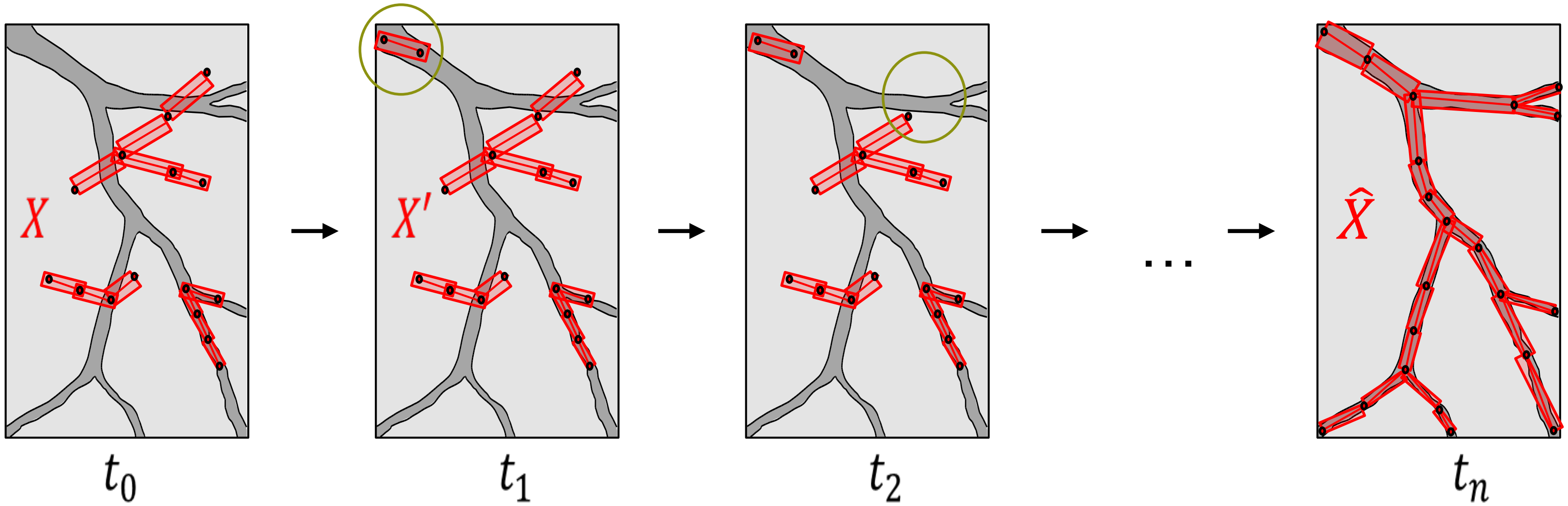


- Object extraction: generation of different configurations X



RJMCMC-Sampling

- Change of the configuration: $X \rightarrow X'$



- Evaluation of the change
- Acceptance or rejection of the new configuration

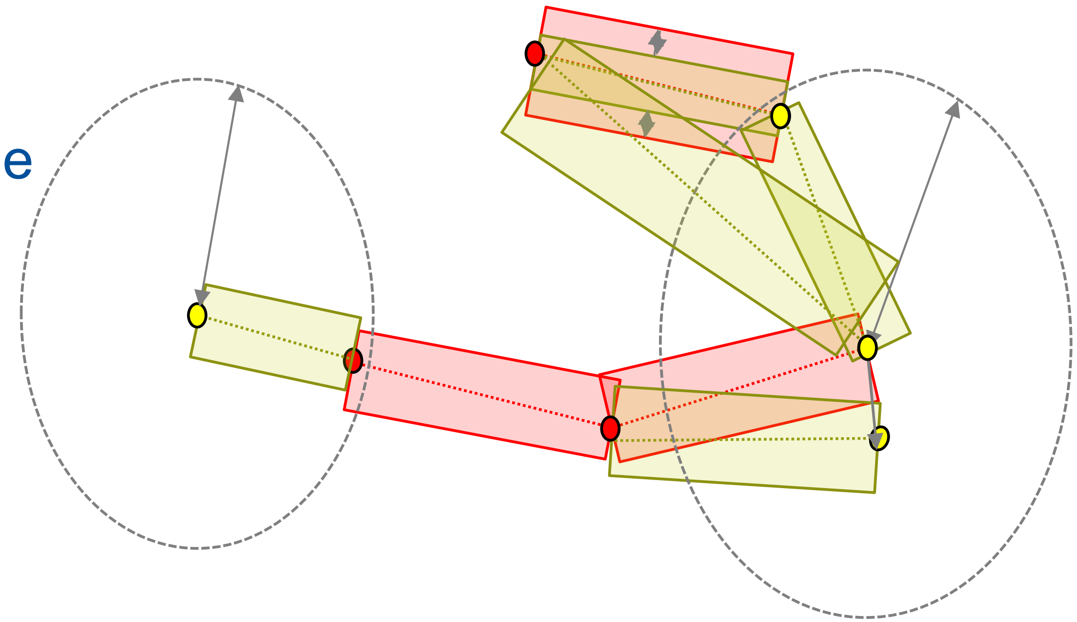
RJMCMC-Sampling

(1) Birth-and-death

(2) Modification

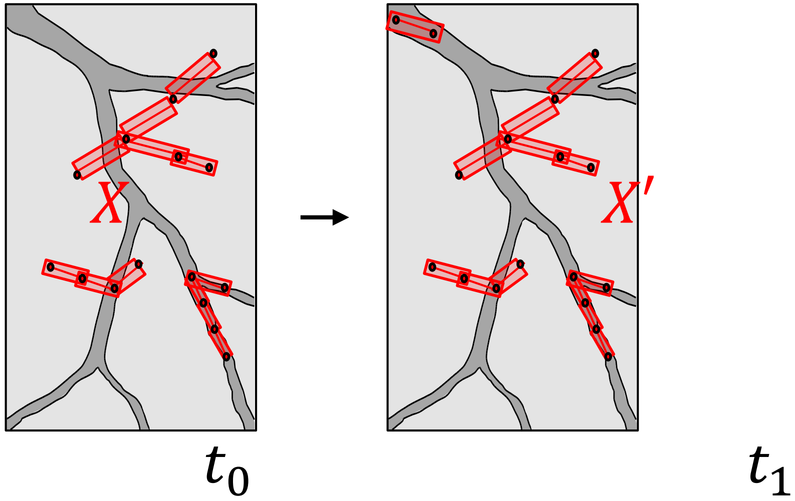
- Coordinates of a node
- Degree of a node
- Width of an edge

(3) Split-and-merge



RJMCMC-Sampling

- Change of the configuration: *X* → *X'*

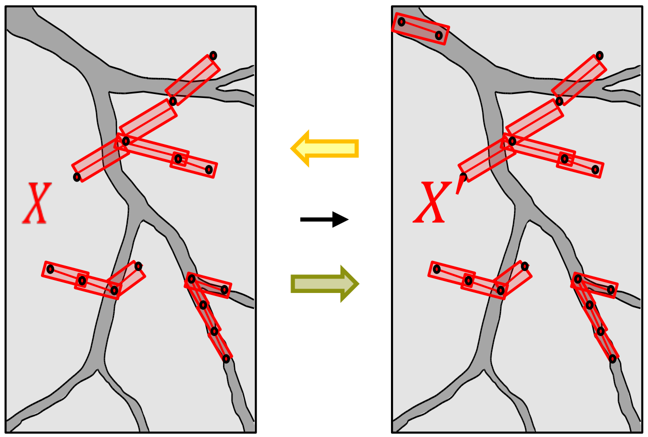


- Evaluation of the change
- Acceptance or rejection of the new configuration

RJMCMC-Sampling

- Computation of the *Green-Ratio* R [Green, 1995]

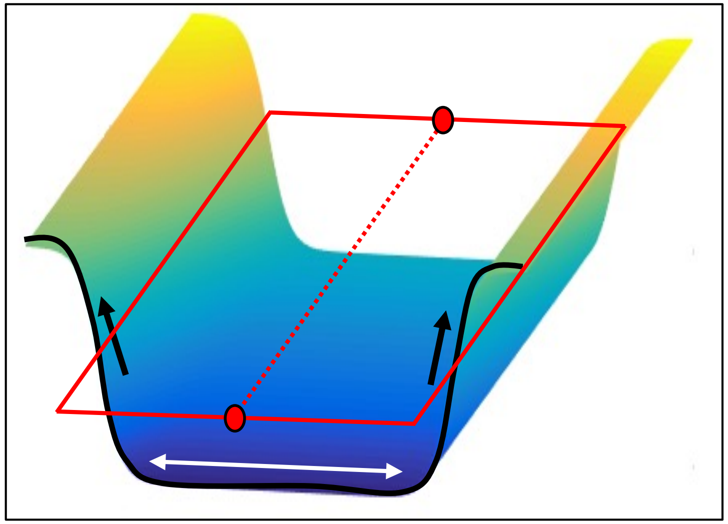
$$R = \underbrace{\exp\left(-\frac{\overset{\text{Energy}}{U(X')} - U(X)}{T_t}\right)}_{\substack{\text{Comparison with data} \\ \text{and prior knowledge}}} \cdot \underbrace{\frac{\boxed{p_Q^r \quad Q(X' \rightarrow X)}}{\boxed{p_Q \quad Q(X \rightarrow X')}}}_{\text{Transition probability}} \cdot \underbrace{\left|\det\left(\frac{\partial f_{X \rightarrow X'}(X, h)}{\partial (X, h)}\right)\right|}_{\substack{\text{Accounts for} \\ \text{dimension change}}}$$



RJMCMC-Sampling

- Energiefunktion: $U = \beta \ U_{Daten} + (1 - \beta) \ U_{Prior}$

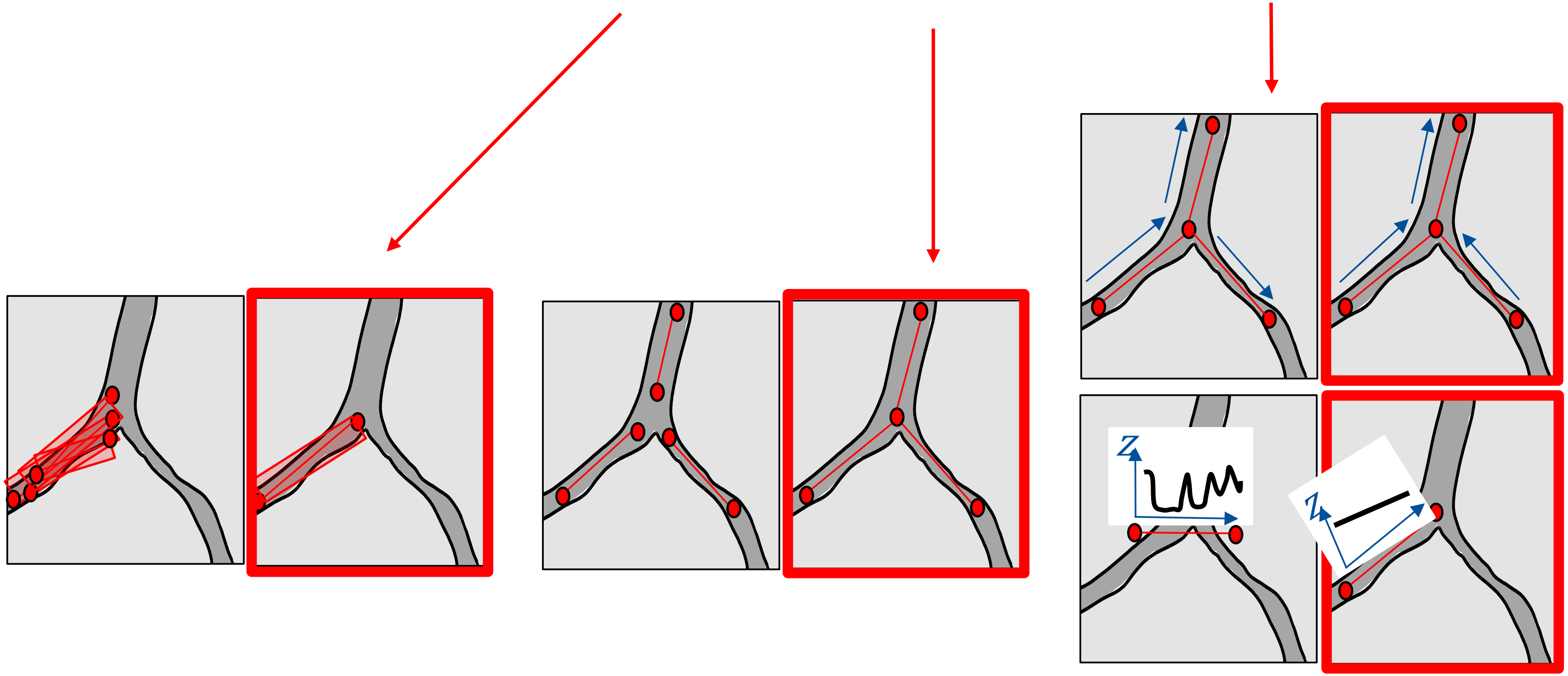
$$U_{Daten} = U_{Gradient\ \ddot{a}} + U_{Homogenit\ t}$$



RJMCMC-Sampling

- Energiefunktion: $U = \beta\, U_{\mathrm{Daten}} + (1-\beta)\, U_{Prior}$

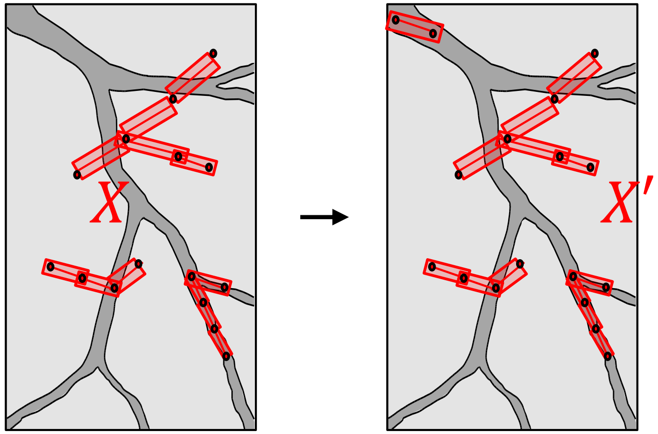
$$U_{Prior} = U_{Überlappung} + U_{Verbundenheit} + U_{Fließrichtung}$$



Overlap Connectivity Flow direction

RJMCMC-Sampling

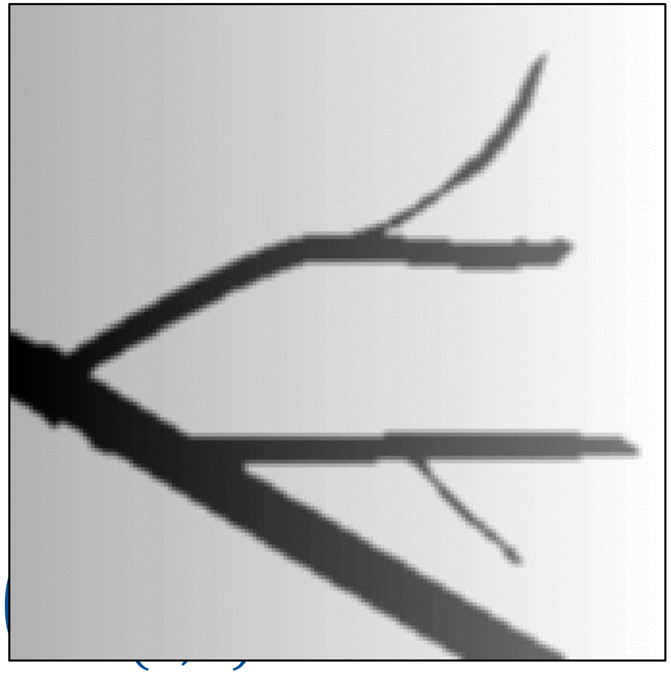
- Change of the configuration: *X* → *X'*



t_0 t_1

- Evaluation of the change
- Acceptance or rejection of the new configuration

RJMCMC-Sampling

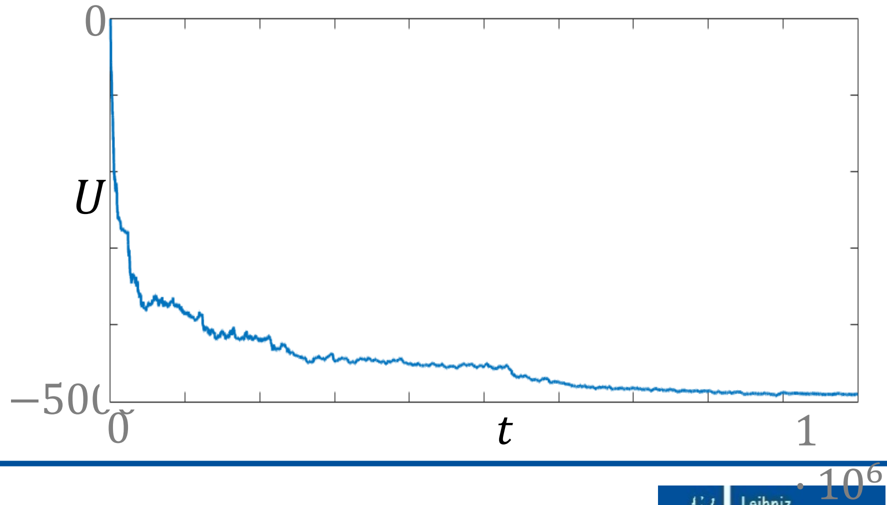


1. Change of configuration $X \rightarrow X'$
   - Birth-and-Death
   - Modification
   - Split-and-Merge

2. Evaluation of the change

$$R = \exp\left(-\frac{U(X') - U(X)}{T_t}\right) \frac{p_Q^r}{p_Q} \frac{Q(X' \rightarrow X)}{Q(X \rightarrow X')} \left|\det \right.$$

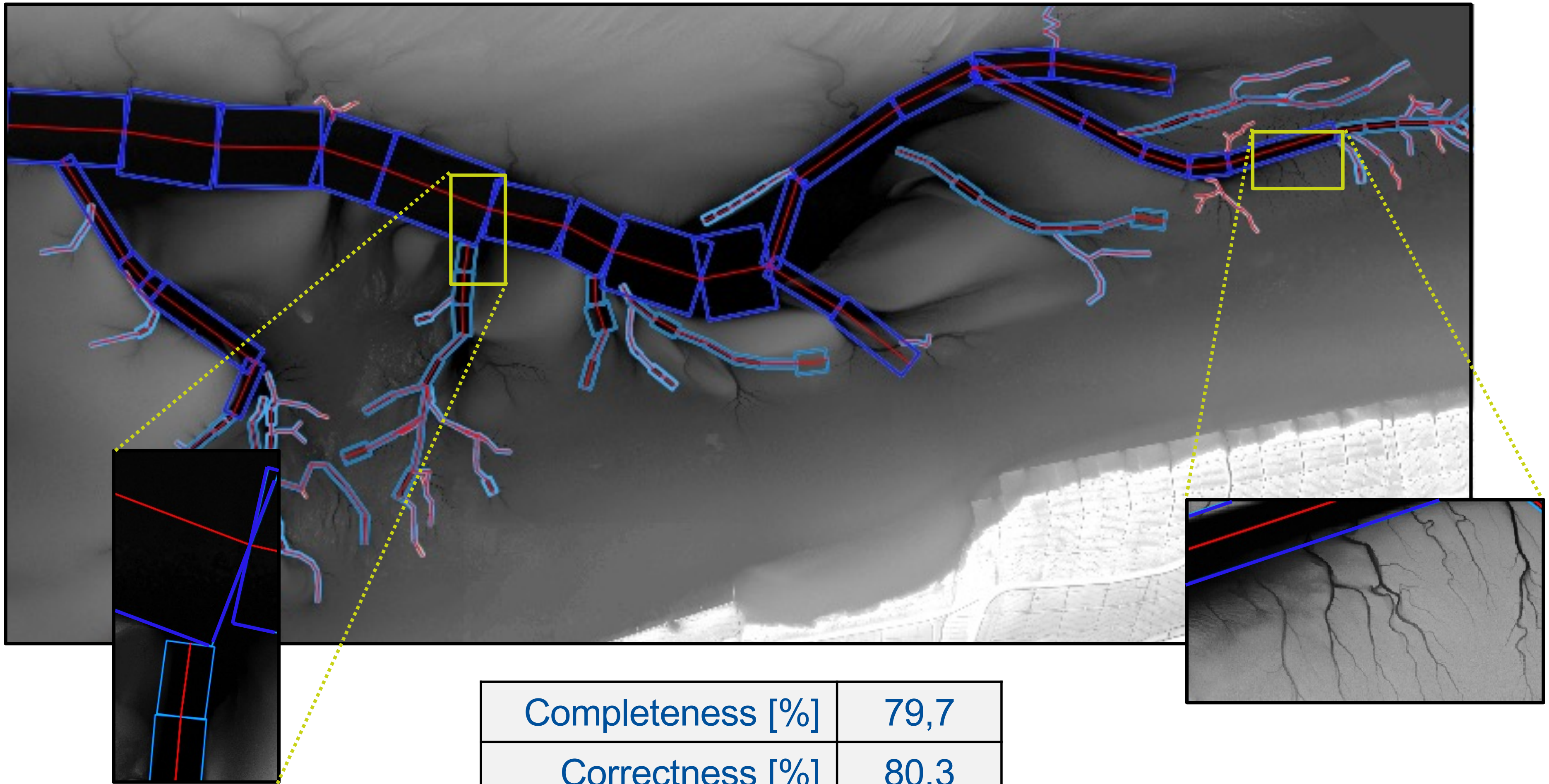
3. Acceptance or rejection



$$\alpha = \min(1, R)$$

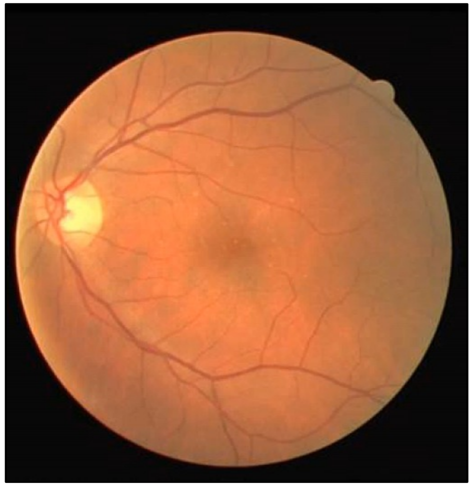
Extraction of river networks

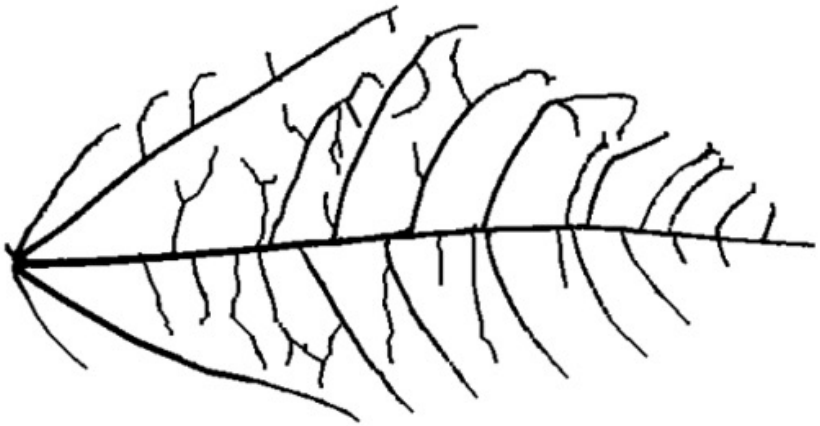
Wadden sea, Lower Saxony



Completeness [%]	79,7
Correctness [%]	80,3

Application to images

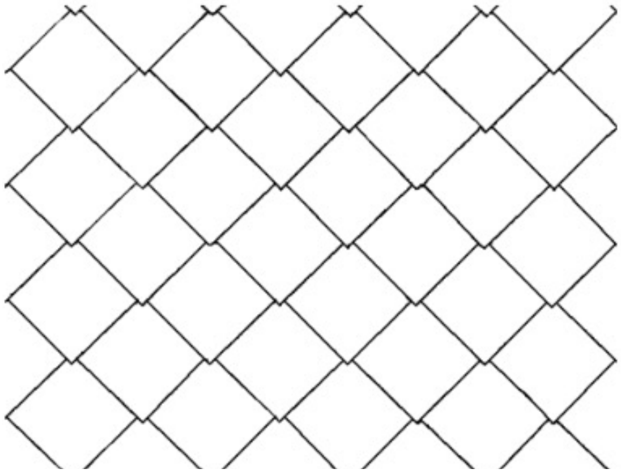




Reference

Image source:
[Verdié et al., 2014]

Reference



Reference

References

- C. M. Bishop: Pattern Recognition and Machine Learning, Springer 2006 (Kap. 11)
- C. P. Robert, G. Casella: Monte Carlo Statistical Methods, Springer 2004
- W. R. Gilks, S. Richardson, D. J. Spiegelhalter: Markov Chain Monte Carlo in Practice, Chapman & Hall/CRC 1996
- Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller: Equations of State Calculations by Fast Computing Machines, In: The Journal of Chemical Physics, vol. 21, pp. 1087–1092, 1953
- W. Keith Hastings: Monte Carlo Sampling Methods Using Markov Chains and Their Applications, In: Biometrika, col. 57, pp. 97-109, 1970
- Radford M. Neal: Slice sampling, In: Annals of Statistics, vol. 31, no. 3, pp. 705-767, 2003

References

- C. Andrieu, N. de Freitas, A. Doucet, M. I. Jordan: An Introduction to MCMC for Machine Learning. Machine Learning vol. 50, pp. 5-43, 2003
- P. J. Green: Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. Biometrika vol. 82, pp. 711-732, 1995

References

- N. Ripperda: Rekonstruktion von Fassadenstrukturen mittels formaler Grammatiken und Reversible Jump Markov Chain Monte Carlo Sampling. Dissertation, Wissenschaftliche Arbeiten der Fachrichtung Geodäsie und Geoinformatik der Leibniz Universität Hannover, Nr. 284. Zugleich erschienen in DGK Reihe C. 2009
- H. Huang, C. Brenner, M. Sester: A generative statistical approach to automatic 3D building roof reconstruction from laser scanning data. ISPRS Journal, vol. 79, pp 29-43, 2013
- Alena Schmidt, Florent Lafarge, Claus Brenner, Franz Rottensteiner, Christian Heipke: Forest point processes for the automatic extraction of networks in raster data, ISPRS Journal, 2017