Freie Universität Berlin

Bayesian Point Cloud Reconstruction

Hannes Geist
Raoul Seifert

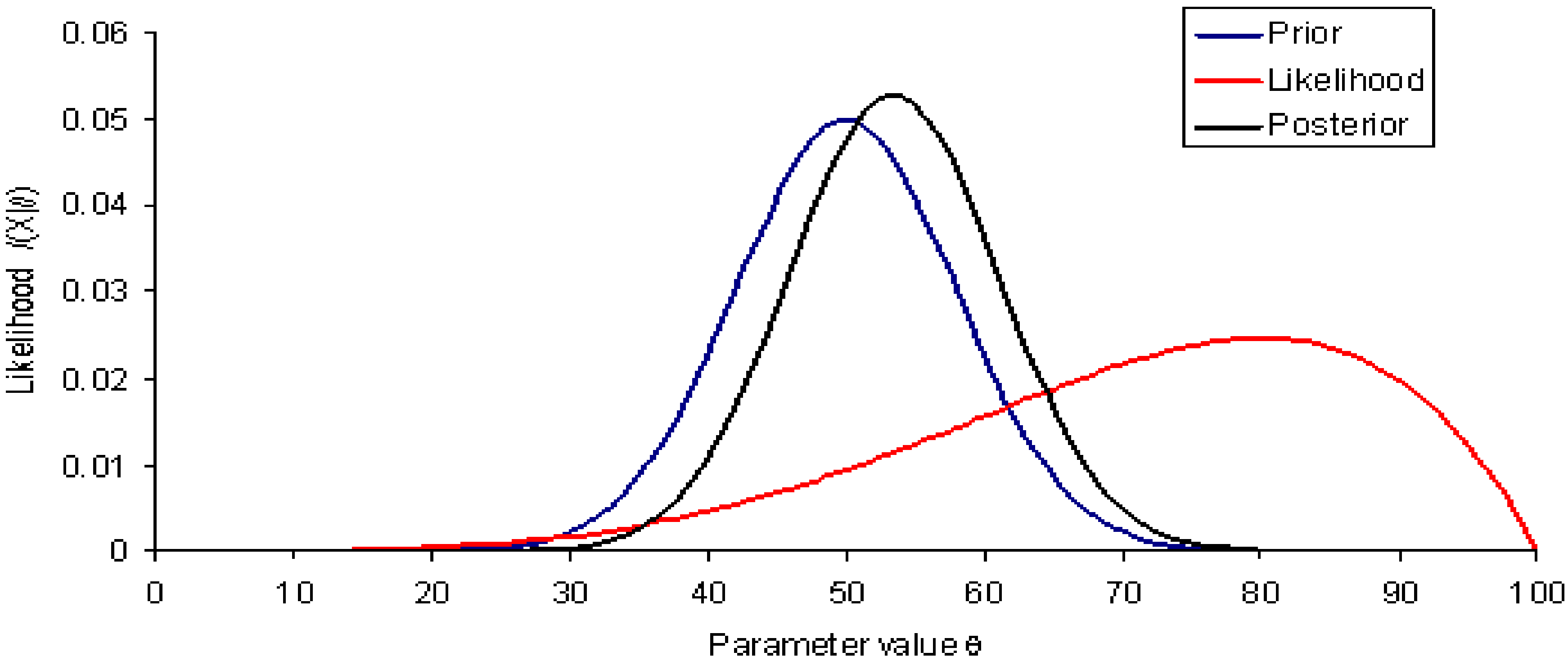
Structure

1. The Mathematical Model

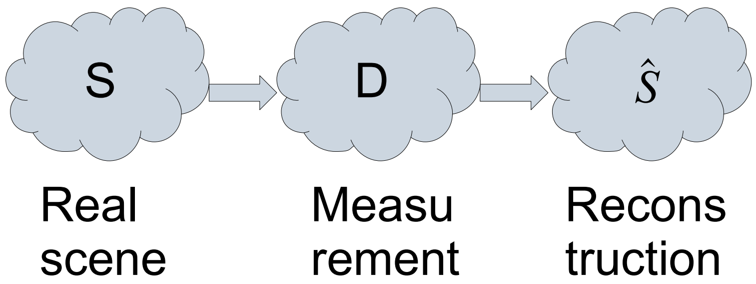
$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

1.1. Bayesian Statistics

$$Posterior = \frac{Likelihood \cdot Prior}{Normalization}$$



1.2. The Idea



Real World Scene S

- Assumption: S is a pointcloud

Measurement Points D

- D consists of measured points
- D is a subset of S with noise added

Reconstruction $\hat{S}$

$$P(\hat{S}|D) = \frac{P(D|\hat{S}) \cdot P(\hat{S})}{P(D)}$$

1.2. The Idea

Most Likely Reconstruction:

$$\underset{\hat{S}}{argmax}\, P(\hat{S}|D) = \underset{\hat{S}}{argmax}\, P(D|\hat{S}) \cdot P(\hat{S})$$

Maximum a posteriori estimation (MAP):

$$\hat{S}_{MAP} = \underset{\hat{S}}{argmin}(-\log P(D|\hat{S}) - \log P(\hat{S}))$$

Freie Universität Berlin

1.3. The Measurement Model

The Measurement Model

- Specifies the probability of a reconstruction agreeing with measured Data
- Fix for each measurement process

Assumptions:

- All measurement errors are independent
- The error is gaussian noise
- The measurement process is unbiased

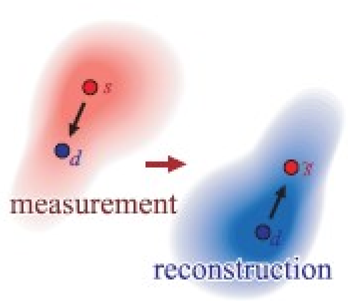
1.3. The Measurement Model

$d_i \in D, s_i \in S$

$$Measurement\ error: p_i(s_i + \Delta x)$$

$$Location\ of\ \tilde{s}_i: p_i(d_i - \Delta x)$$

$$-\log P(D|\hat{S}) = \frac{1}{2} \sum_{i=1}^{n} (\tilde{s}_i - d_i)^T \Sigma_i^{-1} (\tilde{s}_i - d_i)$$



1.4. Priors

The Prior

- Defines what artifacts are considered noise
- Depends on assumptions of the object

Assumptions:

- The objects consist of piecewise smooth patches seperated by sharp boundaries (good for man made objects)

Freie Universität Berlin

1.4. Piors

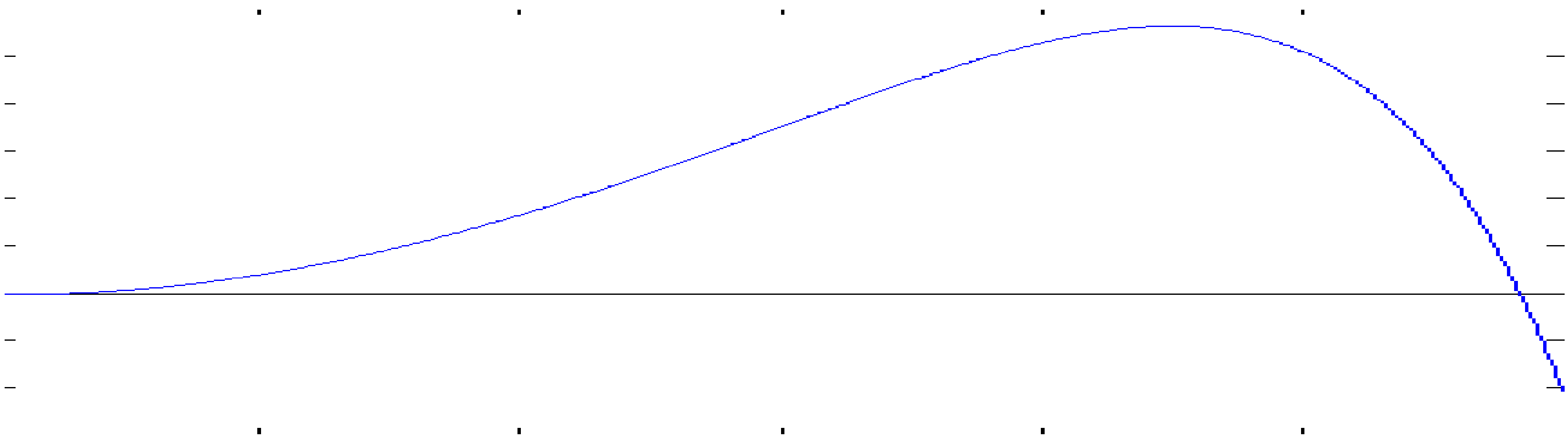
$$p(S)=\frac{1}{Z}\,p_{density}(S)\,p_{smooth}(S)\,p_{discrete}(S)\cdot w(S)$$

$Used\ for\ normalization\,(can\ be\ omitted\,!)$

1.4.1. Density Prior

Used to obtain a well-sampled reconstruction

1. Estimate surface area of the oject
2. Estimate expected distance δ between two points
3. Define stochastic potential between two points p_{dist}
   - Local maximum on the expected point distance



1.4.1. Density Prior

$$P_{density}(S) = \sum_{i=1}^{n} \sum_{i_j \in N_{2\delta}(s_i)} p_{dist}(s_i, s_{i_j})$$

$$N_\delta(x) := set\ of\ all\ points \in S\ within\ radius\ \delta\ of\ the\ point\ x$$

1.4.2. Smoothness Prior

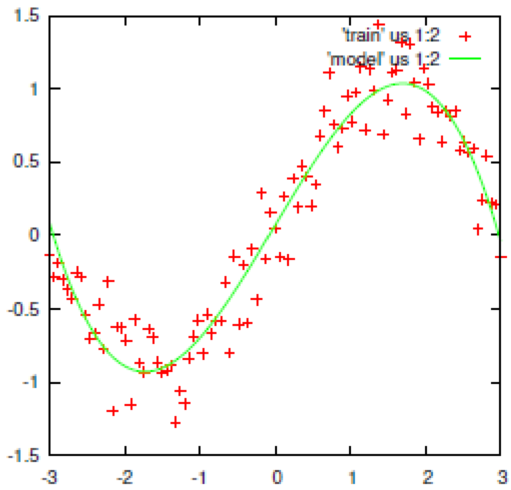
1. Fitting a Plane to a ε-Environment around point s_i using Principle Component Analysis
   - The Eigentvector with the smallest Eigenvalue is the normal direction
   - The other two Vectors span tangetial coordinates u, v
   - Consider the (u, v, n)-coordinates a highfield

1.4.2. Smoothness Prior

2. Linear Regression with quadratic features

- Fix a set of basic functions $\{b_j(u,v)\}_{j=1\ldots k}$
- Compute least squares fit to the corresponding points

$$\begin{pmatrix} \langle b_1, b_1\rangle & \cdots & \langle b_1, b_k\rangle \\ \vdots & \ddots & \vdots \\ \langle b_k, b_1\rangle & \cdots & \langle b_k, b_k\rangle \end{pmatrix} \begin{pmatrix} c_1 \\ \vdots \\ c_k \end{pmatrix} = \begin{pmatrix} \langle b_1, \Phi\rangle \\ \vdots \\ \langle b_k, \Phi\rangle \end{pmatrix} \qquad \Phi(u,v)=n$$



With

$$\langle f, g\rangle = \sum_{j=1}^{|N_\varepsilon(s_i)|} f(u_j, v_j)\cdot g(u_j, v_j)\cdot \omega_{fit}(u_j, v_j)$$

$$\omega_{fit} := weighting\ function\ that\ makes\ solution\ continuous$$

1.4.2. Smoothness Prior

The prior in general

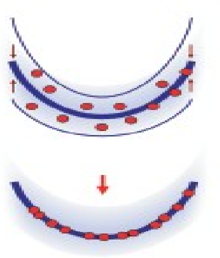
$$P_{lt} = \prod_{i=1}^{n} e^{-f(c(s_i), N(s_i))}$$

- Evaluation function f assigns negative loglikelihood to each set of basisfunctions

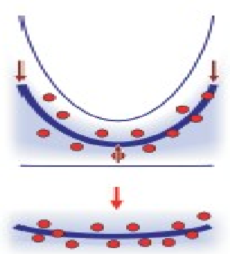
1.4.2. Smoothness Prior

Concrete:

- Basis functions as monomials of second order (1,u,v,uv,u²,v²)
- Weigthed sum (user chosen weights) of two functions: f_{onSurf}, f_{curv}



$$f_{onSurf} = \sum_{i=1}^{N_\varepsilon(s_i)} \left(\left(\sum_{q=1}^{k} c_q b_q(u_j, v_j) \right) - n_j \right)^2$$



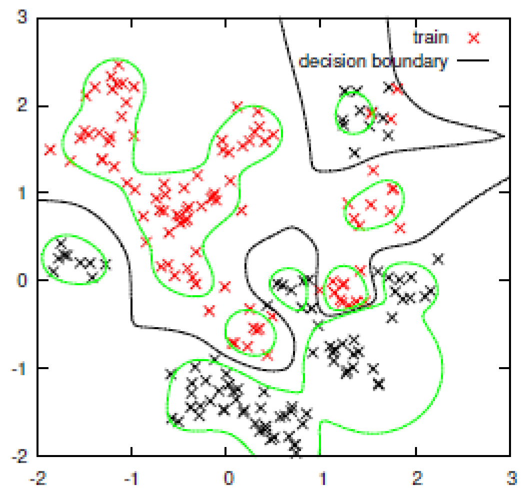
$$f_{curv} = c_{1,1}^2 + 2c_{2,0}^2 + 2c_{0,2}^2$$

1.4.3. Discrete Properties and Sharp Features

Used to distinguish between sharp and smooth features

Assign discrete attributes to each point in S

- Type attribure {region, edge, corner}
- Id number to identify corresponding entity



1.4.3. Discrete Properties and Sharp Features

Problem:

to estimate the discrete attributes knowledge of the continuous attributes is beneficial and vice versa!

General Solution: Expectation Maximization (EM-algorithm)

1.4.3. Discrete Properties and Sharp Features

Assumptions:

- Corner points need at least two points with different edge-ID in neighbourhood
- The number of corner points is exactly one
- Probability distribution for its position peaks at point closest to all edges

1.4.3. Discrete Properties and Sharp Features

Simplified Process:

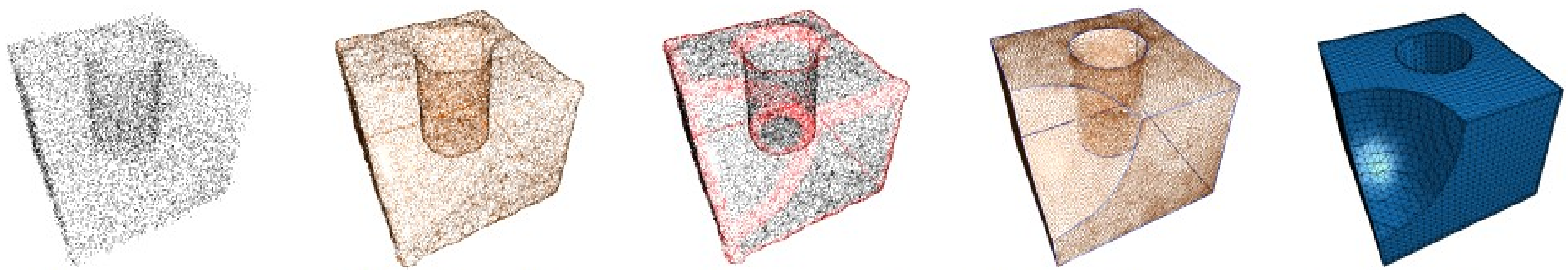
- Probabilty for being an edge grows with the curvature of the local neighbourhood
- Probabilty for being a corner depends on the number of edge points from different edges
- Impose Priors on the shape of edges
  - Uniform sampling (Density Priors)
  - Smoothness

Additional type attribute for each region, defining either „locally polynomial“ or „planar“

Just for simplification

2. Reconstruction

- Apply optimization techniques to find approximate MAP-reconstruction



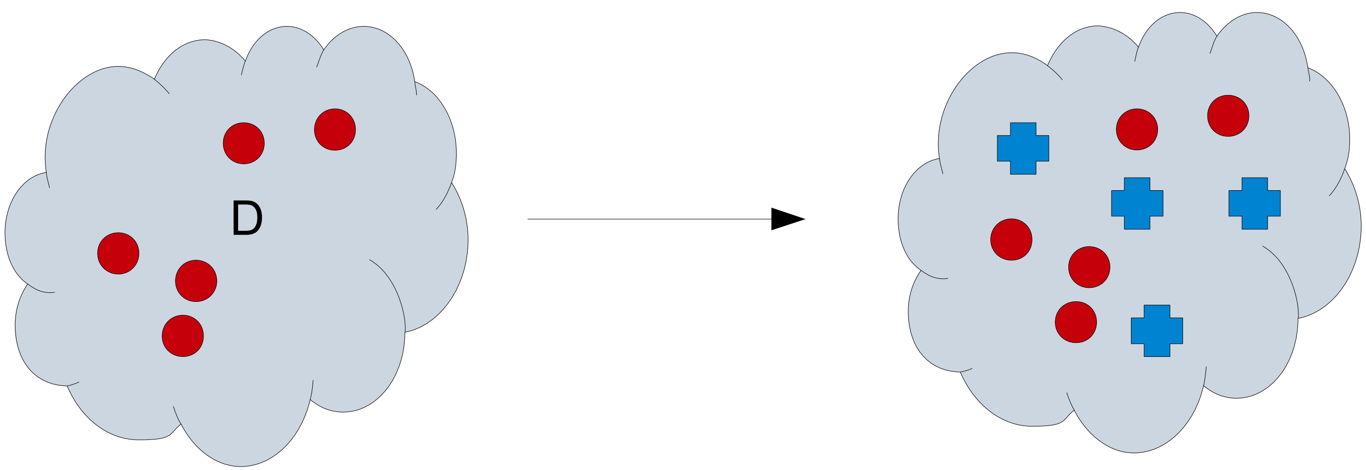
(a) Noisy input point cloud. (b) Initial smoothing. (c) Estimating edge probabilities. (d) Smoothing with discrete attributes. (e) Triangulation of the reconstruction.

2.1. Initialization

1. Initialize with original measurement points D

2. Additional n-m points are distributed randomly near points from D

3. Tool for semi-automatic hole-filling inverse to sampling density



2.2. Numerical Optimization


(b) Initial smoothing.

- In this step: neglect discrete components

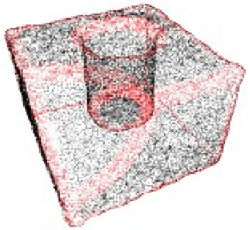
1. Compute gradient of posterior propability $P(S|D)$

2. Gradient descent to maximize posterior propability
   - → implementation of 3 different techniques

3. Measurment likelihood $P(D|S)$ of simple quadric form $\frac{1}{2}\Sigma^{-1}x$
   - → less overhead
   - → accurate analytical solution

In contrast to numerical estimation

2.3. Discrete Optimization



(c) Estimating edge probabilities.

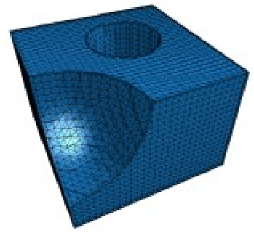
1. Run pass with curvature penality = 0
   - → smooting effect contradicts estimation of edge propabilities

2. Assign propability of point beeing edge
   - → high curvature yields edge

3. Region growing algorithm
   - All points != edges belong to certain region with ID x

4. Second pass of continous optimization with smoothing

3. Triangulation



(e) Triangulation of the reconstruction.

1. Triangulation of points by modified marching cubes algorithm
   - → Include sharp features

2. Overlapping edges are cut to approximate plane of region

3. Step 2 creates gaps between regions
   - → snap vertices together according to region info.

4. Summary

	# Data points	# Rec. points	Rec. time [sec]
Box	2000	4506	28
Holes	4,790	31,717	271
Mechpart	9,521	104,578	1,759
Carved Object	9,973	41,911	269
Face	19,995	300,000	3,772
Fandisk	46,494	216,338	3,881
Floor	199,970	811,352	1,540

Table 1: *Computation time and model complexity.*

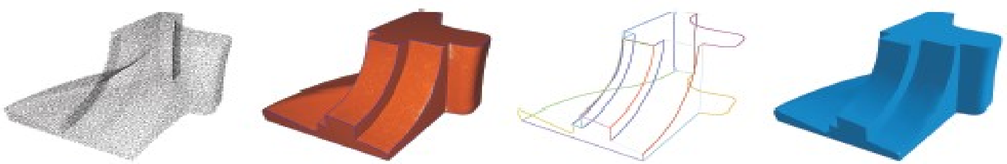
1. Relatively long run time

2. Satisfying results for test objects

3. Good method for sharp man-made objects

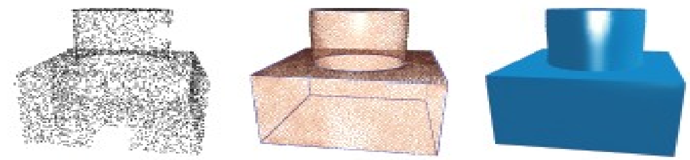
4. Future work
   - Improve speed
   - Better Hole-Filling
   - Better statistical model
   - Handle scenes that change over time

Test Objects



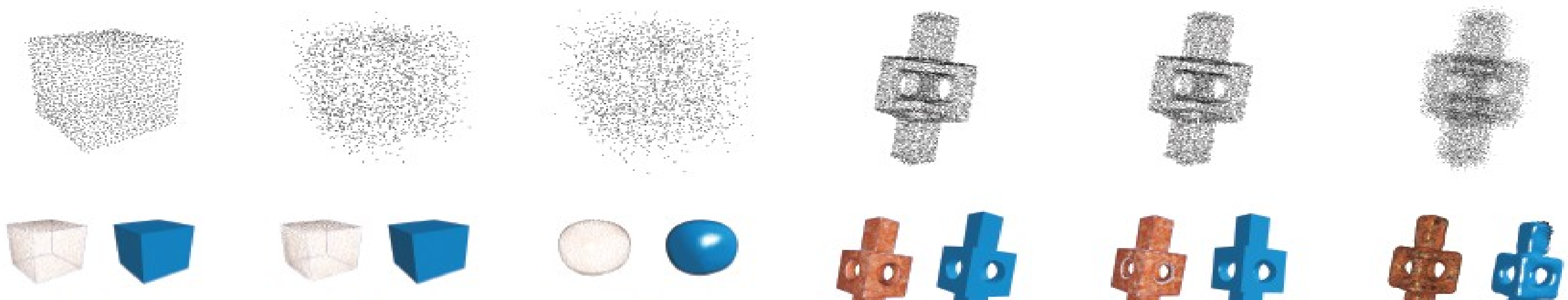
(a) Original data (b) Reconstructed point cloud (c) Reconstructed edges (d) Final mesh

Figure 6: *Reconstructed* fandisk *data set*



(a) Original data (b) Reconstructed point cloud (c) Final mesh

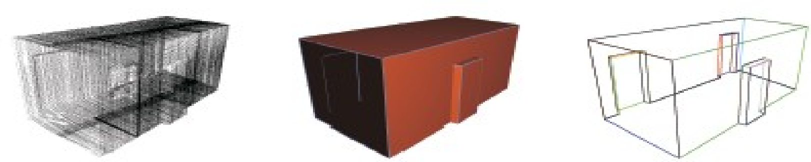
Figure 7: *Data set with small holes*



(a) 1% noise (b) 8% noise (c) 12% noise (d) 1% noise (e) 2% noise (f) 8% noise

Figure 8: *Reconstruction for different noise levels (Gaussian noise, standard deviation relative to bounding box size)*

(a) (b) (c) (d)



(a) Original data (b) Reconstruction (c) Topology