EUROGRAPHICS 2006 / E. Gröller and L. Szirmay-Kalos
(Guest Editors)

Volume 25 (2006), Number 3

Bayesian Point Cloud Reconstruction

P. Jenke[1] and M. Wand[2] and M. Bokeloh[1] and A. Schilling[1] and W. Straßer[1]

[1]WSI/GRIS, University of Tuebingen, Germany
[2]Computer Graphics Lab, Stanford University

Abstract

In this paper, we propose a novel surface reconstruction technique based on Bayesian statistics: The measurement process as well as prior assumptions on the measured objects are modeled as probability distributions and Bayes' rule is used to infer a reconstruction of maximum probability. The key idea of this paper is to define both measurements and reconstructions as point clouds and describe all statistical assumptions in terms of this finite dimensional representation. This yields a discretization of the problem that can be solved using numerical optimization techniques. The resulting algorithm reconstructs both topology and geometry in form of a well-sampled point cloud with noise removed. In a final step, this representation is then converted into a triangle mesh. The proposed approach is conceptually simple and easy to extend. We apply the approach to reconstruct piecewise-smooth surfaces with sharp features and examine the performance of the algorithm on different synthetic and real-world data sets.

Categories and Subject Descriptors (according to ACM CCS): I.5.1 [Models]: Statistical; I.3.5 [Computer Graphics]: Curve, surface, solid and object representations

1. Introduction

In this paper, we consider the problem of surface reconstruction from unorganized, noisy point clouds. This problem arises in various application areas of computer graphics. The most important is probably the reconstruction of geometry from range scanning devices. Most current 3D surface scanning devices (such as structured light, time of flight and many stereo vision based scanners) create unstructured clouds of measurement points in three space. Due to physical limitations, these points are distorted by various kinds of measurement noise. This poses at least two challenges to a surface reconstruction algorithm: First, the surface topology has to be retrieved, and second, the geometry has to be reconstructed from the unorganized point cloud, i.e. noise artifacts have to be removed. These two tasks are often intertwined, as noise affects both the rough and fine scale reconstruction.

In this paper, we propose a surface reconstruction approach based on Bayesian statistics [DHS01]. Bayesian reasoning is probably the most canonical (and most often applied) technique to solving general reconstruction problems. The basic idea is easy to describe: Assume that we are given a real-world scene S and a measurement D (*data*). We then consider the probability space of $\Omega = \Omega_S \times \Omega_D$, the set of all possible real-world scenes and of all possible measurements of them. We know that the measurement D is created from S by a process involving statistical errors. We assume that we understand (at least roughly) the deficits of the measurement process so that we can find an analytic model of $P(D|S)$, which is the probability distribution of the likelihood of measurements D being made of scenes S. This knowledge by itself does not allow for a reconstruction. Assuming an unbiased measurement process, the most probable original scene is still the measurement itself (including all the noise). The Bayesian statistics approach to this problem is defining a probability distribution $P(S)$ over the set of all possible original scenes. Then, we can apply Bayes' rule to invert the measurement process in a statistical sense. We compute the probability of a reconstruction S being the original scene given measurement D as:

$$P(S|D) = \frac{P(D|S)P(S)}{P(D)} = \frac{P(D|S)P(S)}{\int_{\Omega_S} P(D|S)P(S)dS} \qquad (1)$$

The probability distribution $P(S)$ (so called *prior distribution*) will usually not be an exact probabilistic model of all potentially measured scenes (which is infeasible and probably not even well defined) but only a description of partial prior knowledge or belief of *reasonable* models (such as assuming smooth surfaces).

In order to find the most likely reconstruction, we have to determine the scene S that maximizes $P(S|D)$, the so called maximum a posteriori solution SMAP. As the denominator in Equation 1 is only a normalization constant, not depending on S, it is sufficient to compute:

$$S_{MAP} = \arg\max_S P(S|D) = \arg\max_S P(D|S)P(S) \qquad (2)$$

The Bayesian framework provides an elegant and intuitive way of defining reconstruction problems in terms of a probabilistic model of the measurement process and prior beliefs about the object to be reconstructed. However, only few authors have applied the technique to the reconstruction of three dimensional geometry from point clouds [ST89, LP99, DTB06]. A major technical problem is the definition of the probability space for S: A canonical choice would be to consider S being a subset of $\mathbb{R}^3$. However, defining a probability space over all of these sets is not possible and defining a suitably reduced class is mathematically involved. Even then, such a representation is not algorithmically tractable but must be discretized prior to a numerical solution.

At this point we employ an engineering solution: We directly define the set of original scenes to be point clouds themselves that have been subsampled and distorted during the measurement process. This yields a probability space of $\Omega = \Omega_S \times \Omega_D = \mathbb{R}^{3n} \times \mathbb{R}^{3m}$, where n is the number of original points and m the number of measured points. This space can be treated with mathematical standard tools. The prior scene model will be partially based on differential properties of S. In our discretized model, we will approximate these quantities with local fitting models. The main advantage of the point-cloud-based scene model is its generality: Given a large enough number of points n, almost arbitrary real-world scenes can be represented effectively. Especially, it is no problem to adapt the point-based model to arbitrary topologies.

The remainder of this paper is structured as follows: After discussing related work in the next section, we develop a measurement and prior probability model in Section 3. The prior model will include both assumptions on continuous properties (such as surface smoothness) as well as discrete properties for representing sharp features. In Section 4 we describe how to find an approximate maximum a posteriori reconstruction using numerical optimization for continuous and a heuristic search for discrete properties. Finally, as described in Section 5, the reconstructed point cloud is converted into a triangle mesh using a standard triangulation algorithm [HDD*92] that has been augmented to recognize the reconstructed information about sharp features. Finally, we apply our reconstruction pipeline to a set of artificial and real-world data sets to examine the reconstruction performance in practice.

2. Related Work

Surface reconstruction from point clouds is meanwhile a *classic* problem in computer graphics. A variety of techniques have been proposed, which can be classified roughly by the internal representations used for reconstruction:

Implicit functions: One class of surface reconstruction techniques uses implicit functions to represent the surface. This approach was pioneered by the work of Hoppe et al. [HDD*92]. First, normals are estimated by a local principal component analysis (PCA), followed by a neighborhood graph search to unify their inside/outside directions. Then a marching cubes algorithm [LC87] is employed to reconstruct the surface as the zero-levelset of an implicit signed distance function defined by the estimated surface normals. We will use a similar procedure as final step of our algorithm, applied to the reconstructed point cloud. Many extensions of the basic approach have been proposed: A subdivision-based technique to handle piecewise smooth objects is described in [HDD*94]. Dinh et al. [DGS01] employ anisotropic basis functions to improve the representation of sharp features. Another option for defining the implicit function is to use radial basis functions (RBF). Early work used globally supported functions (e.g. [TO99]), leading to quadratic complexity. The complexity can be reduced using a fast multipole method [CBC*01]. Ohtake et al. [OBA*03] define the surface locally via quadratic functions that are blended together globally by weights summing to one (partition of unity). Spatial subdivision is employed to adapt the resolution to the data; sharp features are detected by normal clustering and represented using multiple sets of coefficients.

Moving Least Squares (MLS): MLS approaches are a special case of implicit function-based-techniques. They define a surface as an invariant set of a projection operator. The projection operator is defined as a numerical optimization step on a locally constructed implicit function. Levin [Lev03] presents the projection strategy for general surface interpolation. Alexa et al. [ABCO*03] have introduced the technique to the field of computer graphics. Many variants of the technique have been developed in the meantime: Reuter et al. [RJT*05] propose special basis functions for use in the MLS projection to account for (known) sharp features. Fleishman et al. [FCOS05] use robust statistics to exclude outliers from the fitting process. Different pieces of a piecewise smooth surface are treated as outliers, preserving those feature lines. In comparison with our approach this technique has the advantage (as all MLS techniques) of being fully locally defined so that it can be applied to different small pieces of the model independently. A disadvantage is that considering sharp features as fitting outliers does not allow for a more specific statistical model of the shape for such feature lines. Additionally, the topology of the feature lines is not reconstructed explicitly, which is useful for certain types of further processing, such as meshing. Another interesting MLS variant is proposed by Shen et al. [SOS04]: Considering the limit of an MLS-kernel with increasing local weight allows a faithful reconstruction of discrete objects such as even triangle meshes, without artificial smoothing. Involved numerical integration techniques are used to compute the corresponding integrals.

Voronoi / Delaunay techniques: Amenta and colleagues approached the reconstruction problem from a computational geometry point of view [ABK98, ACK01], focusing on topology reconstruction. Their method gives provable guarantees for reconstructing the true topology given a minimum sampling density dependent on the local feature size. The original approach fails under the influence of noise, but extensions to the algorithm that a capable of handling noisy input have recently been developed [MAV*05].

Mesh Smoothing: One aspect of the reconstruction is noise removal. For this purpose, a number of filtering techniques for triangle-meshes have been proposed, in analogy to image filtering operations [Tau95, DMSB99, DMSB00]. These techniques are usually very efficient in terms of running time and yield good results, but do not provide an explicit statistical model based on assumptions on measurement errors and/or general prior expectations.

Statistical learning: Recently, statistical data analysis and machine learning techniques have gained some attention in the graphics community (see e.g. [DHS01] for an introduction). Ivrissimtzis et al. [IJS03] employ neural networks for surface reconstruction, representing the network via a triangle mesh. The network reacts to point sample signals and updates the mesh respectively. Steinke et al. [SSB05] use support vector machines for reconstruction, hole filling and morphing between datasets. [SBS05] use locally defined kernels that analyze the point neighbourhood to compute a global surface probability distribution. Pauly et al. [PMG04] propose a technique to quantify uncertainty in point cloud data, which can then be used for surface reconstruction. They explicitly avoid the problem of parametrizing the space of all scenes and giving priors on this space. Instead, they quantify uncertainty for points in $\mathbb{R}^3$ by analyzing in how far a point agrees with locally weighted planes. In image processing and computer vision Bayesian techniques are very common, see e.g. [Sto96, KW99, WK01]. However, up to now, to our knowledge, no complete 3D surface reconstruction technique from arbitrary point clouds has been examined. An technique very similar to ours has recently been published by Diebel et al. [DTB06]: They propose a Bayesian framework for the reconstruction of noisy triangle meshes. Similar to our approach, they use a Gaussian error model in conjunction with surface priors and perform numerical optimization to maximize the posterior probability of the model. The most important difference to our work is that their technique assumes to be already given a triangle mesh from which noise has to be removed. The topology and connectivity of the mesh is not altered and must be known beforehand. The priors are also slightly different: Instead of using a mixed discrete/continuous model, Diebel et al. employ a sub-quadratic normal potential that preserves sharp features. This technique is motivated by differential statistics of natural images and yields good results for a variety of natural objects. The technique has the advantage of being able to handle features of different sharpness; however, the sharp feature information is not reconstructed explicitly so that it cannot be used in further processing.

Other related techniques: Our technique has also been inspired by additional techniques, such as the feature preserving Marching Cubes variant of Kobbelt at al. [KBSS01]. Gumhold et al. [GWM01] extract edges from point cloud data via a graph minimization on the neighbour-graph. The point based representation has been motivated by the point-based modelling approach of Szeliski and Tonnesen [ST92], which they also apply to the problem of interpolation and extrapolation of sparse 3D data in [STT93]. The definition of priors via local linear transforms is similar to the point cloud processing technique of Pauly and Gross [PG01].

3. Bayesian Reconstruction

In this section, we discuss the details of our Bayesian reconstruction model. We assume that the following random experiment is taking place. First, we assume that some original scene S consisting of n points has been chosen for measurement according to the probability density $p(S)$. Then, a measurement process deletes some of the original points, leaving $m \leq n$ measured points D. Lastly, random noise is added to the remaining points according to a density $p(D|S)$. To invert the measurement process in a statistical sense, we will construct an estimate point cloud $\widetilde{S}$ with size $n \geq m$. Each of the first m points will be associated with a point from D, i.e. we assume they are distorted versions of the original points. This means they affect $p(\widetilde{S}|D)$ both in terms of the priors $p(\widetilde{S})$ and the measurement model $p(D|\widetilde{S})$. The remaining $n-m$ points correspond to the deleted points; those points are only controlled by the prior $p(\widetilde{S})$. For initialization, the additional points will be distributed close to measurement points with probability inversely proportional to the measurement sampling density. In case of larger holes, the user may also place additional initial points to the data manually.

After this initialization, the algorithm tries to find the reconstruction $\widetilde{S}$ that maximizes the posterior probability $p(\widetilde{S}|D)$ by numerical optimization. More specifically, we will perform a (conjugate) gradient descent on the negative log-likelihood

$$\widetilde{S}_{MAP} = \arg\min_{\widetilde{S}}(-\log p(D|\widetilde{S}) - \log p(\widetilde{S})). \qquad (3)$$

This transformation into log-space is a standard technique; it does not change the solution and makes handling easier (esp. computing derivatives) and numerically more stable.

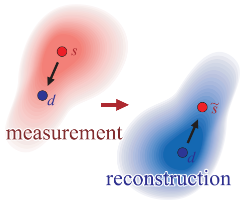


Figure 1: *The measurement error probability density has to be mirrored for the reconstruction.*

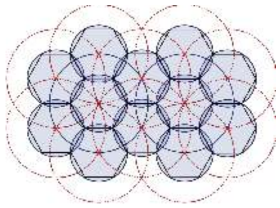


Figure 2: *A minimal circle coverage of the plane is used to estimate the point density.*

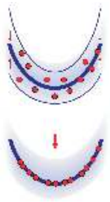


Figure 3: *Smoothness prior: f_{onSurf}*

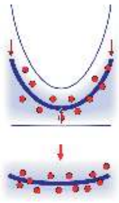


Figure 4: *Smoothness prior: f_{curv}*

3.1. The Measurement Model

The measurement model $p(D|\widetilde{S})$ must specify the probability of a candidate reconstruction $\widetilde{S}$ agreeing with measured data D. If we assume that a measured point d_i has been created from an original point s_i with measurement errors described by a probability density $p_i(s_i + \Delta x)$, the probability density for the location of the reconstructed original $\widetilde{s}_i$ is given by $p_i(d_i - \Delta x)$, which is the mirrored error density (see Figure 1; for symmetric distributions such as Gaussians, the distribution is unchanged). Additionally, we assume that all measurement errors are independent of each other, so that $\log p(D|\widetilde{S})$ is merely the sum of the logarithms of all per-point error densities.

In literature, there are several papers that analyze sensor errors both analytically and empirically, see for example [Cur97,LP99]. The model of choice depends strongly on the scanning technique. While time-of-flight scanners could for example be described by standard Gaussian distributions, triangulation based scanners often show asymmetrically biased error distributions influenced by local reflectivity gradients [Cur97]. As sensor modeling is not the main topic of our paper, we assume just a simple Gaussian model: Each point is associated with a 3×3 covariance matrix, the expected error is assumed to be zero (unbiased measurement). Of course, any other smooth probability density could be employed at this point. Using the independence assumption, we obtain the following log-likelihood (omitting additive constants):

$$-\log p(D|\widetilde{S}) = \frac{1}{2}\sum_{i=1}^{n}(\widetilde{s}_i - d_i)^T \Sigma_i^{-1}(\widetilde{s}_i - d_i) \qquad (4)$$

The resulting quadratic potential function is not very robust in the presence of outliers. In order to limit their influence, we limit the value of the quadric for each point to a user defined constant, leading to a mixture of truncated Gaussians and a uniform distribution with small density values.

3.2. Priors

Prior probabilities are the key to any Bayesian reconstruction technique. They define what artifacts are considered noise and thus what the reconstructed scene will look like. We assume that the object consists of piecewise smooth patches separated by sharp boundaries. Such a model is especially useful for reconstructing scans of man-made objects; for natural objects, other models such as the subquadric normal distribution of Diebel et al. [DTB06] might be more appropriate. Our current set of priors consists of three main ingredients: Density priors, smoothness priors and priors for estimating sharp features. We assume that all three are independent of each other, leading to:

$$p(S) = \frac{1}{Z} p_{density}(S)\, p_{smooth}(S)\, p_{discrete}(S) \cdot w(S) \qquad (5)$$

Z is a normalization constant, which is the integral over all other factors over $\Omega = \mathbb{R}^{3n}$. In order to make this expression integrable, we have to add a windowing function w, otherwise the integral would be infinite. For w we can choose a box function that is 1 inside a large bounding box containing the scene and 0 outside, limiting the range in which we expect to observe parts of the scene. This term mostly accounts for theoretical soundness. In a practical implementation, it can be omitted as the normalization of the probability density does not have an effect and our algorithm and will never attempt reconstructions far away from the scene. Next, we discuss the different terms that describe the expected surface properties:

Density priors: We want to obtain a reconstructed model which is well-sampled with a regular, constant sampling density all over the surface. Therefore, we estimate the surface area of the object by computing the sum of the area of all circles in which the k-nearest neighbors of each point lie, divided by k. This estimate will be recomputed at each iteration of the numerical optimization to make the approximation more accurate. Given the surface area, we can estimate the expected distance δ between two neighboring points: Assuming locally flat surfaces, the well-known minimal circle coverage in a plane (Figure 2) gives an estimate for an optimal packing of sample points. Next, we define a stochastic potential p_{dist} between all pairs of points that constrain the distance: Within a neighborhood radius proportional to the expected point to point distance, the probability density has a local maximum at the desired expected point distance while being lower otherwise. For points that are far away, the probability remains constant (please note that the windowing function will make this integrable over Ω_S). Let $N_\delta(x)$

denote the set of all points in S within radius δ of the point x. Then, we define the density prior as:

$$p_{density}(S) = \sum_{i=1}^{n} \sum_{i_j \in N_{2\delta}(s_i)} p_{dist}(s_i, s_{i_j}) \tag{6}$$

The heuristic variant we use and which works well in practice is to only consider the 6 nearest neighbors of each point s_i. The six neighbors are determined by finding the closest points after projecting it into a local fitting plane, considering one closest point for each of six 60° cones around s_i in this plane. We compare the distance towards these neighbors with the beforehand determined optimal distance and use the differences to determine the point in space that minimizes the error. This discrete choice of neighbors makes the probability function C_0-continuous only (rather than C_∞, as Equation 6). However, this does not appear to be an issue for our gradient-based optimization technique.

Smoothness: To quantify the smoothness of a surface, we use a general technique based on a linear basis transformation: First, we fix the size for a local neighborhood of diameter ε (typically about 20-40 points). We do not choose the corresponding set of nearest neighbors but a geometrically defined set to avoid different smoothness conditions due to sampling variations. For all points in the neighborhood $N_\varepsilon(s_i)$ of a point s_i, we compute a local coordinate frame using a principal component analysis of the local point cloud [GWM01]: The eigenvector with the smallest eigenvalue corresponds to the normal direction n while the two others span tangential coordinates u, v. Then, we consider all neighbors as a sparse heightfield in n direction over u, v. Let u_j, v_j and n_j be the corresponding (u, v, n)-coordinates of the points from $N_\varepsilon(s_i)$.

We fix a set of basis functions $\{b_j(u,v)\}_{j=1...k}$ and compute a least square fit to the corresponding points by solving the system of normal equations:

$$\begin{pmatrix} \langle b_1, b_1\rangle & \cdots & \langle b_1, b_k\rangle \\ \vdots & \ddots & \vdots \\ \langle b_k, b_1\rangle & \cdots & \langle b_k, b_k\rangle \end{pmatrix} \begin{pmatrix} c_1 \\ \vdots \\ c_k \end{pmatrix} = \begin{pmatrix} \langle b_1, \phi\rangle \\ \vdots \\ \langle b_k, \phi\rangle \end{pmatrix} \tag{7}$$

with

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

and ϕ being a function that returns the n-coordinates of the neighborhood points at their respective u, v coordinates. ω_{fit} is a weighting function that makes the solution continuous with respect to movement of points in S. Both, Gaussian and simple constant windowing functions lead to reasonable results.

The system of normal equations allows us to compute a set of coefficients of basis functions c_i, which will then be used to describe the prior. In general, the linear transform prior will have the following form:

$$p_{lt}(S) = \prod_{i=1}^{n} exp(-f(c(s_i), N(s_i))) \tag{8}$$

Here, $c(s_i)$ denotes the vector of coefficients computed for the neighborhood of a certain point s_i and f is an *evaluation* function that assigns a negative log-likelihood to each set of basis functions and their respective original points.

In order to quantify the smoothness of a surface, we choose monomials of second order $\{1, u, v, uv, u^2, v^2\}$ as basis functions and employ a weighted sum (with user chosen weights) of two evaluation functions, f_{onSurf} and f_{curv}, as negative log-likelihood. The first describes how far the points are away from the quadratic surface:

$$f_{onSurf} = \sum_{i=1}^{|N_\varepsilon(s_i)|} ((\sum_{q=1}^{k} c_q b_q(u_j, v_j)) - n_j)^2 \tag{9}$$

The second penalizes high curvature of the fitted patch by computing the average squared norm of second derivatives of the patch, which is:

$$f_{curv} = c_{1,1}^2 + 2c_{2,0}^2 + 2c_{0,2}^2 \tag{10}$$

where $c_{i,j}$ is the coefficient of basis function $u^i v^j$. The two functions describe two different effects (see Figures 3 and 4): If we penalize large curvature only, we still obtain noise with frequency in the range above $O(\varepsilon^{-1})$. On the average, the cloud of noise lies close to a surface of low curvature. If we only penalize the distance to the local polynomial approximation, the noise vanishes but the surface can show undesired irregularities at a medium frequencies below $O(\varepsilon^{-1})$. We could think of using only one evaluation function f and increasing or decreasing ε correspondingly. However, this leads to excessive computational cost (increasing ε) or accuracy problems (decreasing ε), respectively. Instead, we use a medium value for ε (corresponding to some 10-40 neighbor points) and let the user choose a weighting parameter for the curvature component to determine his prior expectations concerning smoothness of the surface.

The concept of linear transforms of point neighborhoods is rather general and could be used to describe many probabilistic assumptions concerning local correlations between points. For example, we could think of using a Fourier basis to describe systematic *rippeling* errors of certain frequency (which are typical for structured light scanners) in the probabilities of the measurement model by assigning a large variance to the corresponding Fourier coefficients.

3.3. Discrete Properties and Sharp Features

Up to now our framework only allows for the reconstruction of smooth surfaces. However, in many applications it is desirable to handle sharp features. This could be done by modifying the evaluation function f, for example by using non-quadratic potentials [DTB06]. Nevertheless, there are still many applications in which an explicit reconstruction of these features is desirable. In our case, this is especially beneficial for the later conversion into a triangle mesh. To generalize this framework, we assign each point s_i of the original scene (and thus of its Bayesian reconstruction) a set of discrete attributes in addition to the continuous attributes. To estimate the discrete attributes, an accurate knowledge of the continuous attributes would be beneficial. However, the discrete attributes themselves influence the estimation of the continuous attributes. A general technique to deal with this kinds of problems is expectation maximization [DHS01]: The algorithm iteratively estimates the probabilities for the assignment of the discrete and continuous attributes in turn, using the estimate from the previous stage to improve the estimate of the corresponding other attribute set. For our sharp feature model, we use a simplified procedure inspired by this technique.

We employ the following discrete attributes per point: A type attribute which determines whether a point belongs to a region (smooth patch), an edge (border curve between regions) or a corner (where two or more edges meet), and an *ID-number* to identify the corresponding entity. We assume that smooth regions are always separated from each other by edges. This assumption makes the estimation task easier; however it also limits the algorithm to models without sharp features within one and the same region.

We augment our statistical model accordingly: We assume that the *original* model S provided all the discrete attributes, which were unfortunately lost during measurement. However, we have some general, a priori knowledge about the assignment of these attributes: The probability for being an *edge* point grows with the curvature of the local neighborhood. We could for example use the same f_{curv} evaluation function to quantify this probability (our current implementation uses the simpler PCA-based estimate of [GWM01]). Correspondingly, the probability for being a corner point depends on the number of edge points from different edges (different IDs) in the neighborhood. This defines probabilities for being a candidate for an edge or corner point but does not affect the shape. To define the effect on shape, we restrict the neighborhoods used for the other priors (smoothness, density) to points within the same region (edge points belong to both adjacent regions). Additionally, and this is very important for good results, we also impose priors on the shape of edges: In analogy to the region priors, we also demand uniform sampling (by looking at the distance towards the two neighbor edge points and defining corresponding pairwise probabilities) and smoothness (by applying the same measures as outlined in the previous subsection, but now locally fitting a polynomial curve of 2nd order to the set of edge points). Finally, we assume that corner points can only exist with points with two or more different edge-IDs in their ε-neighborhood, that the number of corner points in such a neighborhood is exactly one and assume a probability distribution for its position that peaks at the point were regression lines for all edges are closest to each other.

In addition to these attributes, we also employ a type attribute for each region; it can be locally polynomial or planar. In the first case, the prior probabilities are employed as described before. In case of *plane*-regions, the smoothness priors are replaced by a simpler prior that attracts the points to a single plane, constant for the complete region. The probability for choosing the *plane* attribute for a region depends on the least-squares error when fitting to one global plane (with user define allowable variance). Given the probabilistic model augmented by discrete attributes, we are looking for an $\widetilde{S}$ that maximizes the overall probability.

The resulting mixed discrete-continuous optimization problem does not seem to lend itself to an optimal solution with reasonable computational efforts. Thus, we employ a heuristic approximation strategy, as described at the end of the next section.

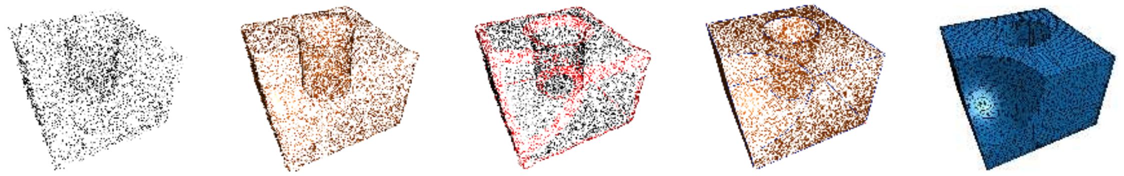
4. The Reconstruction

In this section, we describe the employed optimization technique for finding an approximate MAP-reconstruction (see Figure 5).

Initialization: We initialize the estimate of the reconstructed point cloud $\widetilde{S}$ with original measurement points D. The additional $n-m$ points are distributed randomly in the neighborhood of the points from D, with probability inversely proportional to the local sampling density in D. We estimate the sampling density by the inverse of the averaged distance towards the nearest neighbors in each of the six 60° cones (the same as used for the density priors, see Section 3.2). An exact knowledge of the sampling density is of minor importance as points will be redistributed by the density prior later on. We have also implemented an interactive tool for semi-automatic hole-filling where the user can click on a set of border points and additional points will be distributed within the convex hull of the selection.

Numerical Optimization: First, we try to maximize the a posteriori probability neglecting the discrete components of our model. This is done by numerical descent: We compute the gradient of the posterior probability $P(\widetilde{S}|D)$ for the candidate point cloud $\widetilde{S}$ with respect to the positions of all of its points and perform a gradient descent. We have implemented three different techniques: simple gradient descent with manually chosen, fixed sized steps, gradient descent with automatically chosen step size, and conjugated gradient descent [She94, DTB06]. For the two latter techniques, the step sizes are chosen by fitting a one dimensional parabola to the objective function in gradient direction to perform a quasi-Newton line search; in case of divergence, the step size is automatically reduced or even inverted [DS96, She94].

In order to perform this optimization, it is crucial to com-



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

Figure 5: *The sequence (a-e) of reconstruction steps in an example dataset (*carved object*)*

pute the gradients of the posterior analytically. Besides instabilities, a numerical estimation would cause a significant computational overhead (for example, the least squares fit had to be repeated for each point within each neighborhood). Consequently, our model was chosen to allow for an easy gradient computation: the measurement likelihood $P(D|\tilde{S})$ (Equation 4) is a simple quadric with gradient $\frac{1}{2}\Sigma^{-1}x$. For the linear transform priors, we restrict the movement of points in each neighborhood to the normal direction, i.e. we only compute the partial derivatives of equations 9, 10 with respect to n_i. This turned out to yield more stable results in practice. The second derivatives for the line search algorithm are currently estimated numerically.

Discrete Optimization: The discrete optimization procedure builds on top of the continuous optimization. First, we perform a run of the continuous optimization, neglecting the discrete attributes. In this pass, we set the influence of the curvature penalty term in the priors to zero as its smoothing effect contradicts the estimation of edge probabilities. After convergence of the optimization, we start assigning probabilities for being an edge point to points of high curvature. Afterwards, we start a region growing algorithm on an ε-neighborhood graph of the point cloud for non-edge points and assign a unique region ID to each such point. The graph traversal is stopped by points that show a significant probability of being an edge, i.e. where the estimated curvature exceeds a user-defined threshold. For each point that has not been assigned to a region yet, its distance to an extrapolation of the adjacent regions is used to decide which region it belongs to. Points that lie on the border between adjacent regions are labeled as edges. This labeling step yields a rough and still noisy estimate of edges. Thus, we run a second pass of continuous optimization, enforcing smoothness priors on edges (and activating the curvature prior that has been deactivated before). This step optimizes the position of edge, region and corner points to (locally) maximize the model likelihood given the discrete attributes. The discrete/continuous estimation process could be iterated to refine the model; however, to our experience, one such iteration is sufficient as we already converge to a local optimum.

5. Triangulation

The final step is to create a triangle mesh from the point cloud, which might be a preferable representation for some applications. After having reconstructed a well-sampled and almost noise free point cloud, this is not very complicated any more: Meshing is done with an implicit surface reconstruction scheme, similar to the original algorithm of Hoppe et al. [HDD*92]. We just augment the algorithm to make use of the sharp feature information we obtain from the point cloud reconstruction. First, triangles are created using a marching cubes algorithm on a signed distance function defined by normals that are estimated from nearest neighbours in the reconstructed point cloud. An individual signed distance function is defined for every region in order to preserve the reconstructed features. To increase the stability of this step, we employ a variant of the MLS-function proposed in [SOS04] with a small, cube-sized Gaussian weighting kernel. Then the marching cubes algorithm is executed several times, once for each region located near the actual cube. This produces overlap at the edges. To clean this up, the resulting triangles for each region are clipped to an approximating plane of the points from all other regions. In the resulting mesh there are small gaps between two regions caused by the cutting process. Closing the mesh is done by snapping to close edge vertices. The correspondence of matching vertices can be determined by tracking the region IDs of the triangles and clipping planes.

6. Results

We have implemented the reconstruction algorithm within the framework of a scanner data processing software. The range queries for determining a local ε-neighborhood or the k-nearest neighbors, respectively, have been implemented using a hierarchical query algorithm on an octree-based data structure. The implementation has been done in C++; all experiments have been conducted on a system with an AMD Athlon 64 3500+ processor and 2 GB of RAM. Using the prototype, we have examined different aspects of the algorithm:

Noisy 3D models: We have applied the algorithm to a set of synthetic 3D models. The original triangle meshes were converted into point clouds by uniform random sampling and Gaussian noise of varying amplitude has been added. Figure 5 shows a test data set similar to that used in [FCOS05] with 2% noise added. The algorithm reconstructs both planar and curved surface regions and edges, the noise is not visible in the final reconstruction. We have also applied the algorithm to the standard *fandisk* benchmark scene (Figure 6). Again, the algorithm removes the noise and correctly reconstructs all sharp edges except one sharp edge at the back of the data set (Figure 6c): This edge cannot be detected by our approach because it ends in the middle of a smooth region; hence the current region growing approach does not recognize it.

Different noise levels: Next, we have varied the level of noise and recomputed the reconstruction. Figure 8 shows the reconstruction results for a simple box scene and a variant of the well-known *mechanical part* data set from [HDD*92]. For very large noise levels, our current curvature-based edge detector is not able to reliably detect edges anymore. Nevertheless, a reasonable smooth surface is reconstructed in these cases. This shows that the combination of discretized smoothness and density priors still converges to smooth surfaces for unfavorable initial estimates with large variance in the measurement model. However, the topology and discrete features may be reconstructed incorrectly. Please note that the local feature size value ε and the curvature threshold had to be chosen manually and separately for each case. This is the case for all data sets, these two main parameters have to be determined by the user, they are currently not inferred automatically. The other parameters worked for all data sets without manual intervention.

Holes: The algorithm has not specifically been designed for hole filling. However, holes smaller than the ε-parameter are filled automatically by the density prior. Slightly larger holes can be handled by enlarging ε during the initialization phase of the algorithm. The result can be seen in Figure 7. For larger holes, the user can insert support points inside the hole manually. This allows to remove at least minor defects from scanned data sets (Figure 9).

Real-world scenes: We have also applied the algorithm to two real-world scenes from range scanners (assuming a uniform Gaussian error model). Figure 9 shows the reconstruction of a scanned face from a structured light scanner. Measurement noise has been smoothed out. However, the sharp feature reconstruction is less useful for such natural object scenes. Finally, we have also reconstructed an architectural model, a part of the floor of our department acquired with a mobile time-of-flight scanner. Here, the algorithm automatically reconstructs all important features (walls and doors, see Figure 10).

Numerical optimization: We have applied three variants of gradient descent: Simple, fixed step-width, adaptive step width and conjugated gradients (also with automatically chosen step width). The adaptive and conjugated gradient techniques needed significantly fewer steps for convergence than the non-adaptive version. In our experiments, the CG-method showed by far the best performance when applied to purely quadratic problems it has been designed for. However, in the reconstruction settings, the number of steps have been comparable to non-conjugated, adaptive gradient descent. A problem is, however, that the two adaptive methods need several evaluations of the objective function per step. As the result, they are currently slightly slower than the very simple gradient descent technique. This problem could probably be fixed using analytically computed 2nd order line derivatives. As currently being the fastest, all running times have been measured using simple plain gradient descent.

	# 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.*

Running time: The overall running time for the test scenes is given in Table 1. In comparison with local reconstruction techniques such as MLS, the running times are significantly longer, as we optimize a global statistical model. The most expensive part is the numerical optimization. For example, for the *fandisk* data set, the initial smoothing took 691 sec., initial edge estimation 148 sec., and the second smoothing step 3042 sec. The second step is more expensive because more priors are evaluated and more iterations are performed for final convergence. The final triangle mesh reconstruction took 102 sec for a 200^3 grid.

7. Conclusions and Future Work

In this paper, we have presented a surface reconstruction technique from noisy point clouds based on Bayesian statistics. The main idea is to perform a search for a local maximum of posterior probability in the space of all possible reconstructions, which are parameterized as point clouds, i.e. as points in the high dimensional space $\mathbb{R}^{3n}$. It turns out that the approach works quite robustly in practice and allows to reconstruct models of general, a priory unknown topology and remove noise artifacts while preserving and explicitly reconstructing sharp features. The resulting representation lends itself for a straightforward conversion into a triangle mesh, preserving the explicit sharp feature information. The main limitation of our current technique is the reconstruction technique for the sharp features. The edge probability being based on curvature estimates does only work if the noise level is smaller than the curvature threshold that distinguishes

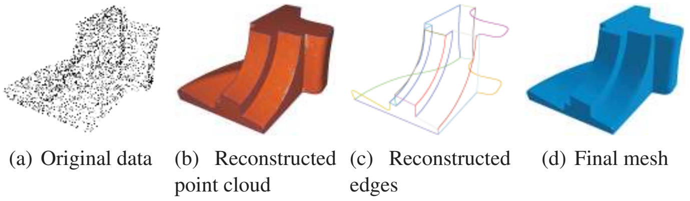


Figure 6: *Reconstructed* fandisk *data set*

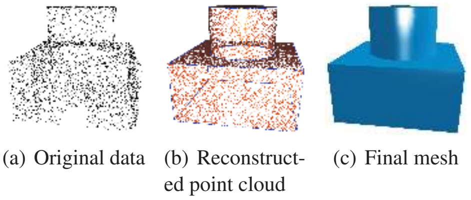


Figure 7: *Data set with small holes*

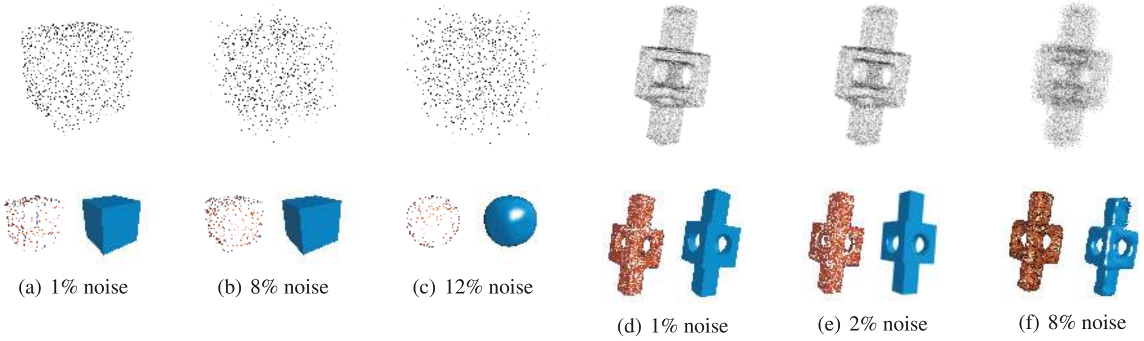


Figure 8: *Reconstruction for different noise levels (Gaussian noise, standard deviation relative to bounding box size)*

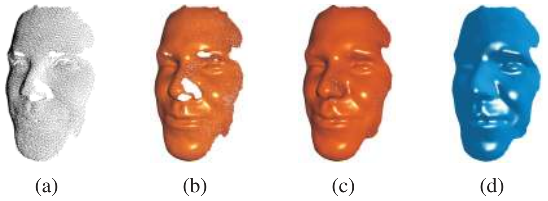


Figure 9: *A scanned face: original data (a), reconstruction (b), manual initialization (c), triangulation (d)*

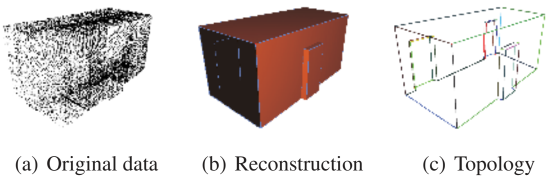


Figure 10: *A building scan*

the edges. In future work, we would like to work on a more elaborated statistical model. A promising research direction is also the investigation of alternative priors, for example transformations to other linear function systems such as the Fourier basis. We are currently working on a statistical model based on self-similarity in order to fill large holes more meaningfully. Another interesting direction could be an application to time-variant data, by adding additional priors on time-dependent scene behavior. Finally, several technical aspects could be improved, such as employing a higher order numerical optimization scheme, which could improve the computation time significantly.

8. Acknowledgments

The authors would like to thank Peter Biber and Sven Fleck for providing the *floor* dataset and Benjamin Maier for his *face* scan. We appreciated the valuable comments by the anonymous reviewers.

References

[ABCO*03] Alexa M., Behr J., Cohen-Or D., Fleishman S., Levin D., Silva C. T.: Computing and rendering point set surfaces. *IEEE Transactions on Visualization and Computer Graphics 9*, 1 (2003), 3–15.

[ABK98] Amenta N., Bern M., Kamvysselis M.: A new voronoi-based surface reconstruction algorithm. In *ACM SIGGRAPH '98* (New York, NY, USA, 1998), ACM Press, pp. 415–421.

[ACK01] Amenta N., Choi S., Kolluri R. K.: The power crust. In *SMA '01* (New York, NY, USA, 2001), ACM Press, pp. 249–266.

[CBC*01] Carr J. C., Beatson R. K., Cherrie J. B., Mitchell T. J., Fright W. R., McCallum B. C., Evans T. R.: Reconstruction and representation of 3d objects with radial basis functions. In *ACM SIGGRAPH '01* (New York, NY, USA, 2001), ACM Press, pp. 67–76.

[Cur97] Curless B.: *New Methods for Surface Reconstruction from Range Images*. PhD thesis, Stanford University, 1997.

[DGS01] DINH H. Q., G.TURK, SLABAUGH G.: Reconstructing surfaces using anisotropic basis functions. In *ICCV '01* (2001), vol. 2, pp. 606–613.

[DHS01] DUDA R. O., HART P. E., STORK D. G.: *Pattern Classification*. John Wiley and Sons, Inc., 2001.

[DMSB99] DESBRUN M., MEYER M., SCHROEDER P., BARR A. H.: Implicit fairing of irregular meshes using diffusion and curvature flow. In *ACM SIGGRAPH '99* (New York, NY, USA, 1999), ACM Press/Addison-Wesley Publishing Co., pp. 317–324.

[DMSB00] DESBRUN M., MEYER M., SCHROEDER P., BARR A. H.: Anisotropic Feature-Preserving denoising of height fields and images. In *Graphics Interface '00* (2000), pp. 145–152.

[DS96] DENNIS J. E., SCHNABEL R. B.: *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Society for Industrial and Applied Mathematics, 1996.

[DTB06] DIEBEL J. R., THRUN S., BRÜNIG M.: A bayesian method for probable surface reconstruction and decimation. *ACM TOG 25*, 1 (2006), 39–59.

[FCOS05] FLEISHMAN S., COHEN-OR D., SILVA C. T.: Robust moving least-squares fitting with sharp features. *ACM TOG 24*, 3 (2005), 544–552.

[GWM01] GUMHOLD S., WANG X., MACLEOD R.: Feature extraction from point clouds. In *10^{th} International Meshing Roundtable* (Oct 2001), Sandia National Laboratories, pp. 293–305.

[HDD*92] HOPPE H., DEROSE T., DUCHAMP T., MCDONALD J., STUETZLE W.: Surface reconstruction from unorganized points. In *ACM SIGGRAPH '92* (New York, NY, USA, 1992), ACM Press, pp. 71–78.

[HDD*94] HOPPE H., DEROSE T., DUCHAMP T., HALSTEAD M., JIN H., MCDONALD J., SCHWEITZER J., STUETZLE W.: Piecewise smooth surface reconstruction. In *ACM SIGGRAPH '94* (New York, NY, USA, 1994), ACM Press, pp. 295–302.

[IJS03] IVRISSIMTZIS I., JEONG W.-K., SEIDEL H.-P.: Using growing cell structures for surface reconstruction. In *SMI '03* (Seoul, Korea, May 2003), Kim M.-S., (Ed.), IEEE, pp. 78–86.

[KBSS01] KOBBELT L. P., BOTSCH M., SCHWANECKE U., SEIDEL H.-P.: Feature sensitive surface extraction from volume data. In *ACM SIGGRAPH '01* (New York, NY, USA, 2001), ACM Press, pp. 57–66.

[KW99] KEREN D., WERMAN M.: A full bayesian approach to curve and surface reconstruction. *J. Math. Imaging Vis. 11*, 1 (1999), 27–43.

[LC87] LORENSEN W. E., CLINE H. E.: Marching cubes: A high resolution 3d surface construction algorithm. In *ACM SIGGRAPH '87* (New York, NY, USA, 1987), ACM Press, pp. 163–169.

[Lev03] LEVIN D.: Mesh-independent surface interpolation. *Advances in Computational Mathematics* (2003).

[LP99] LANG J., PAI D. K.: Bayesian estimation of distance and surface normal with a time-of-flight laser rangefinder. In *3DIM '99* (1999), pp. 109–117.

[MAV*05] MEDEROS B., AMENTA N., VELHO L., , DE FIGUEIREDO L. H.: Surface reconstruction for noisy point clouds. In *SGP '05* (2005), pp. 53–62.

[OBA*03] OHTAKE Y., BELYAEV A., ALEXA M., TURK G., SEIDEL H.-P.: Multi-level partition of unity implicits. *ACM TOG 22*, 3 (2003), 463–470.

[PG01] PAULY M., GROSS M.: Spectral processing of point-sampled geometry. In *ACM SIGGRAPH '01* (New York, NY, USA, 2001), ACM Press, pp. 379–386.

[PMG04] PAULY M., MITRA N. J., GUIBAS L. J.: Uncertainty and variability in point cloud surface data. In *PBG '04* (2004).

[RJT*05] REUTER P., JOYOT P., TRUNTZLER J., BOUBEKEUR T., SCHLICK C.: Surface reconstruction with enriched reproducing kernel particle approximation. In *PBG '05* (2005).

[SBS05] SCHALL O., BELYAEV A., SEIDEL H.-P.: Robust filtering of noisy scattered point data. In *PBG '05* (Stony Brook, New York, USA, 2005), Pauly M., Zwicker M., (Eds.), Eurographics Association, pp. 71–77.

[She94] SHEWCHUK J. R.: *An Introduction to the Conjugate Gradient Method Without the Agonizing Pain*. Tech. rep., Carnegie Mellon University, Pittsburgh, PA, USA, 1994.

[SOS04] SHEN C., O'BRIEN J. F., SHEWCHUK J. R.: Interpolating and approximating implicit surfaces from polygon soup. *ACM TOG 23*, 3 (2004), 896–904.

[SSB05] STEINKE F., SCHÖLKOPF B., BLANZ V.: Support vector machines for 3d shape processing. In *EUROGRAPHICS '05* (2005), vol. 24.

[ST89] SZELISKI R., TERZOPOULOS D.: From splines to fractals. In *ACM SIGGRAPH '89* (New York, NY, USA, 1989), ACM Press, pp. 51–60.

[ST92] SZELISKI R., TONNESEN D.: Surface modeling with oriented particle systems. In *ACM SIGGRAPH '92* (New York, NY, USA, 1992), ACM Press, pp. 185–194.

[Sto96] STORVIK G.: Bayesian surface reconstruction from noisy images. In *Interface '96* (1996).

[STT93] SZELISKI R., TONNESEN D., , TERZOPOULOS D.: Modelling surfaces of arbitrary topology with dynamic particles. In *IEEE CVPR '93* (New York, NY, 1993), pp. 82–87.

[Tau95] TAUBIN G.: A signal processing approach to fair surface design. In *ACM SIGGRAPH '95* (New York, NY, USA, 1995), ACM Press, pp. 351–358.

[TO99] TURK G., O'BRIEN J. F.: Shape transformation using variational implicit functions. In *ACM SIGGRAPH '99* (New York, NY, USA, 1999), ACM Press/Addison-Wesley Publishing Co., pp. 335–342.

[WK01] WERMAN M., KEREN D.: A bayesian method for fitting parametric and nonparametric models to noisy data. *IEEE TPAMI 23*, 5 (2001), 528–534.