

## Markov Random Field Surface Reconstruction

**Paulsen, Rasmus Reinhold; Bærentzen, Jakob Andreas; Larsen, Rasmus**

*Published in:*  
I E E Transactions on Visualization and Computer Graphics

*Link to article, DOI:*  
[10.1109/TVCG.2009.208](https://doi.org/10.1109/TVCG.2009.208)

*Publication date:*  
2010

*Document Version*  
Publisher's PDF, also known as Version of record

[Link back to DTU Orbit](#)

*Citation (APA):*  
Paulsen, R. R., Bærentzen, J. A., & Larsen, R. (2010). Markov Random Field Surface Reconstruction. I E E Transactions on Visualization and Computer Graphics, 16(4), 636-646. DOI: 10.1109/TVCG.2009.208

## DTU Library

Technical Information Center of Denmark

---

### General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

# Markov Random Field Surface Reconstruction

Rasmus R. Paulsen, Jakob Andreas Bærentzen, and Rasmus Larsen

**Abstract**—A method for implicit surface reconstruction is proposed. The novelty in this paper is the adaption of Markov Random Field regularization of a distance field. The Markov Random Field formulation allows us to integrate both knowledge about the type of surface we wish to reconstruct (the prior) and knowledge about data (the observation model) in an orthogonal fashion. Local models that account for both scene-specific knowledge and physical properties of the scanning device are described. Furthermore, how the optimal distance field can be computed is demonstrated using conjugate gradients, sparse Cholesky factorization, and a multiscale iterative optimization scheme. The method is demonstrated on a set of scanned human heads and, both in terms of accuracy and the ability to close holes, the proposed method is shown to have similar or superior performance when compared to current state-of-the-art algorithms.

**Index Terms**—Bayesian approach, implicit surface, Markov random field, mesh generation, surface reconstruction.

## 1 INTRODUCTION

SCATTERED point interpolation is a well-studied problem and application of this technique is found in a variety of fields; from the geological prediction of mineral reserves based on a few drill holes to the reconstruction of objects in microscopy images. In this paper, we will focus on the problem of robustly constructing surfaces from sampled point sets. The 3D data used in this paper are from a clinical environment, where patients (typically babies) are scanned pre- and postoperatively. Since the subjects can typically only be scanned once, there will be cases where points or anatomical structures are missing or incomplete. The criteria that are stated below are based on these assumptions. However, the algorithm presented is general and can handle point sets from different modalities. The following criteria were used in the design and implementation of the presented algorithm:

- **Robust:** No self-intersections and nonmanifold edges in the reconstructed surface. Can handle surfaces that are not topologically equivalent to spheres.
- **Accurate:** The reconstructed surface follows the input data faithfully. Possible to control the accuracy in the presence of noise.
- **Hole filling:** Plausible interpolation in case of missing data and holes. Control of hole-filling strategy.
- **Flexible:** Possible to adapt the algorithm based on knowledge of the nature of the data and the physical and statistical properties of the data capture modality.

The method proposed is based on the computation of a distance field, which is later regularized using a statistical framework. Three optimization schemes are demonstrated. Finally, the surface is extracted as a level set of the regularized distance field.

- *The authors are with the Informatics and Mathematical Modelling, Technical University of Denmark, Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby, Denmark. E-mail: {rrp, jab, rl}@imm.dtu.dk.*

*Manuscript received 9 Dec. 2008; revised 13 Aug. 2009; accepted 14 Sept. 2009; published online 6 Nov. 2009.*

*Recommended for acceptance by R. Whitaker.*

*For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org, and reference IEEECS Log Number TVCG-2008-12-0200. Digital Object Identifier no. 10.1109/TVCG.2009.208.*

## 2 RELATED WORK

The work presented here falls within the class of algorithms that fit implicit 3D functions to an input point set. The approach was pioneered in the work by Hoppe [1], where a distance field is computed for each point by finding the distance to the closest plane fitted locally to the input data. In Hoppe's work, the triangulated surface is extracted using the marching cubes algorithm [2]. Today, alternative implicit triangulators are available [3], [4]. A similar method was used to align data sets using knowledge of the physics of the scanning modality [5]. A recent method uses radial basis functions (RBF) to represent the underlying 3D function [6]. It uses estimated point normals to create a *density* representation of the input data. The normals are used to place points with positive values outside the (at the time unknown) object and points with negative values inside the object. The RBF is then fitted to these *offset* points and the surface is extracted as the zero level of the 3D function. The fitting is done by solving a large and dense linear system, which is far from trivial. An alternative approach is described in [7], where local quadratic functions are fitted locally and weighted globally using a partition of unity method. The approach is inherently multiscale since an octree subdivision is used for representing the basic functions. An octree approach is also used by Kazhdan et al. [8], where the surface is found by solving a Poisson problem. In physics, the Poisson equation is used to solve problems involving diffusion of, for example, heat. The solution is found by blurring an indicator function, and solving a sparse linear system. It is based on the work described in [9], where the Fast Fourier Transform is used to compute the solution. This method indirectly minimizes the membrane energy of the distance field and, as will be shown later, this is not always optimal. The method has later been extended by others [10]. The method described in this paper resembles the method described in [11], where a distance field representation of the input data is regularized by minimizing a bending energy using a double Laplacian operator. However, their method is designed for implementation on the GPU and has been optimized for speed.

Furthermore, it is not obvious how the parameters of the methods can be adapted to surfaces and data with different characteristics. The prior art in this field is abundant and we have only mentioned a selection of the most relevant.

The novelty in this paper is the use of Markov Random Field (MRF) regularization in a Bayesian framework. Since the paper by Geman and Geman [12] on stochastic relaxation, Gibbs distributions, and Bayesian restoration, the Bayesian paradigm has been used in a variety of domains. Early work typically focuses on restoring noisy 2D images. In the work by Besag [13], [14] local models that favor similar pixel neighbor intensities are applied, and the image probabilities are optimized using stochastic methods. Another example is image labeling, where each pixel is classified based on both scene-specific models and knowledge of the physical and statistical properties of the capture device [12], [15]. MRF regularization has also been used on nonscalar fields. A specific inspiration for our work is found in [16], where it is described how optical flow fields computed from time series of radar images are regularized. Each pixel contains a 2D flow vector and several models that operate on these multidimensional fields are described. These models are similar to the ones we propose. Recently, a method to regularize 3D vector fields has been described in [17]. The vector fields are constructed as correspondences between two meshes, and the MRF regularization is used in an alignment framework. The involved probabilities are based on an assumed similarity between neighboring vectors and correspondences between regions with similar curvature on the two shapes. The MRF approach presented in this paper is based on a 3D scalar field and uses energy functions and optimization strategies described in, for example, [16], [17]. A thorough introduction to Markov Random Fields can be found in [18].

Statistical approaches have previously been applied to surface reconstruction and surface manipulation. An early example can be found in [19], where a Bayesian approach is used to classify regions of voxels in volumetric images. A Bayesian approach is also used in [20], where a dense point cloud is computed based on a measured point cloud. The observation model is modeled as a truncated Gaussian based on point distances. The prior term includes both a smoothness term and a density term. Furthermore, sharp features are included in the probability function. While not mentioned in the paper, the formulation is close to the Markov Random Field approach. In contrast to our work, which is based on an implicit description of the surface, the method in [20] exclusively reconstructs the point cloud, and the surface is extracted using the method in [1]. An alternative method working exclusively with triangular surfaces is presented in [21], where it is assumed that a surface has already been computed. The first part of the paper describes a Bayesian regularization of the surface, where an oriented Gaussian distribution is used as an observation model, and a square root prior is used as smoothness prior. The second part of the paper presents a Bayesian framework for surface decimation. Even though the statistical framework presented in [21] is similar to our framework, there is a difference in that we use an implicit surface representation and that we do not only regularize

but also reconstruct the surface from input points. A statistical approach to combined alignment and reconstruction of point clouds is described in [22]. Here, a set of point clouds is roughly aligned. Second, a *prototype* surface is computed using a local quadratics approach that is very similar to the method described in [7]. In the next step, the poses of the input point clouds are changed to match the prototype. The optimization is formulated in a Bayesian framework, where the prior defines a smoothness of the surface and the probability is explained by a projected distance of points to the reconstructed surface. The parameters optimized are the collection of rigid transformations of the point clouds and the parameters of the fitting quadratics. As will be demonstrated later, the method from [7] is fast and accurate, but has problems with regards to hole filling. In [23], surfaces are extracted from multiple camera views using a probabilistic model and a Bayesian framework. As will be demonstrated, the priors used in our work are linear. Nonlinear priors for surface normals are demonstrated in a surface reconstruction scheme in [24]. This approach is similar to anisotropic diffusion.

Surface reconstruction can also be framed in a variational and partial differential equation (PDE) framework. In [25], the level set method is used to reconstruct surfaces from scattered point data, and in [26], the level set method is used in a probabilistic framework to reconstruct 3D shapes from range scans. These methods have also been used in stereo reconstruction [27]. As in our method, the level set methods typically work on an implicit volumetric representation of the surface. There are obvious similarities in the results that can be obtained using the level set methods and in the results obtained using our method. In [25], a membrane-like surface is computed in the same way as seen in our approach. The major difference is the formulation of the models and the optimization methods employed. Our method is also characterized by a decoupling of the behavior of the surface in high-confidence areas and the behavior where no samples are present.

Our method defines the desirable behavior when there is no data. This indirectly makes the method close holes in a predictable way. Recently, inpainting has been used for hole filling [28], [29], [30]. As with the previously mentioned PDE methods, these are typically based on a variational definition of the behavior of the surface where there are holes in the surface. In [30], functionals are defined using the mean curvature of the isosurfaces of an implicit surface representation and the solution is found using a PDE framework. In [29], a finite element method is employed to minimize the Willmore energy over the surface. The results obtained using inpainting seem to be comparable to our results. However, the formulation and the optimization are different.

The preprocessing of the point cloud employed in this paper resembles the approach used when computing point set surfaces [31]. Here, local maps of the geometry are computed using the moving least squares (MLS) method. Points in a local neighborhood are first fitted by a plane and then a low-order polynomial is fitted to the residuals from a plane projection. This allows both up and downsampling of the point cloud and can also be used for noise removal. It can be considered as a low-pass filter of the point cloud.

Our approach is based on fitting a plane and using this as input to a distance calculation. Since a small neighborhood with few samples is used, it is doubtful whether we would achieve more accurate results using the full MLS fitting. In [32], an extension to the method in [31] is presented. By using robust statistics the method is able to identify sharp edges and thereby reduce the low-pass filter effect of the original approach.

### 3 METHOD

The proposed method consists of four steps. Initially, an oriented point set with consistent normal directions is created from the input data. Second, a signed distance field is computed based on the oriented point set. In the next step, a Markov Random Field based regularization is applied to the distance field. Note that the word *distance field* is used in a loose sense since we do not enforce unit length gradients. In addition, the multiscale methods re-estimate distances in between the regularization steps. Finally, the surface is extracted using an isosurface extraction algorithm, and the resulting mesh is optimized.

#### 3.1 Preprocessing the Point Set

The point set is processed to remove outliers and create consistent normal estimates. Initially, a sampling density estimate is computed by finding the average  $\mu_l$  and standard deviation  $\sigma_l$  of the distances to the closest point neighbors. The distance is calculated for each point by locating the closest point and computing the Euclidean distance between the two. This estimate is used as a parameter in the next steps. An estimate of the point normals is computed using local principal component analysis (PCA) using a point search radius of  $2.5\mu_l$ . A point is considered noise, if the third Eigenvalue explains more than 10 percent of the local variation or if the distance from the point to the plane indirectly estimated by the PCA is greater than  $\mu_l$ . Second, a graph-based search algorithm is used to group points into connected components, where two points are considered neighbors if their distance is less than  $\mu_l + 6\sigma_l$  and their mutual normal angle is less than 15 degree. Connected components containing fewer points than 1 percent of the total number of input points are removed. During the graph search, the normals in the connected components are forced to have consistent directions. Finally, the normal directions in the connected components are compared to the position of the capture device. If a majority of normals are pointing away from the capture device, all the normals in the connected component are flipped. The result is a point set with few outliers and consistent normal directions.

#### 3.2 Computing the Signed Distance Field

The distance field is represented as a uniform voxel volume, where the value in each voxel is the signed distance to the nearest oriented sample point. A uniform space division is deliberately chosen in favor of a more advanced approach, for example, octrees. The reason is that if the input point cloud is used as the target when the octree subdivision is computed, areas with no points or sparse samples are represented by very big octree cubes, thus lowering the achievable resolution in these areas, for example, this can be observed with surfaces reconstructed by the Poisson

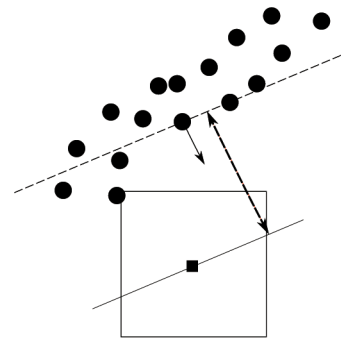


Fig. 1. Projected distance. The distance from the voxel center (small solid square) to the point with the normal is shown as the dashed double-headed arrow.

approach [8]. Since our focus is on hole filling, it is important that holes are reconstructed at the same spatial resolution as the densely sampled areas. We, therefore, believe that a uniform spatial division is important, even though it limits the achievable resolution. The initial distance field is computed using a distance method similar to the one described in [11]. The input points are stored in an octree for faster localization. For each voxel, the five closest (to the voxel center) input points are found using the standard Euclidean metric. Second, the distance to the five points is computed as the projected distance from the voxel center to the line spanned by the point and its associated normal as seen in Fig. 1. Finally, the distance is chosen as either the average or the median of the five distances. Using the average is equivalent to minimizing the  $L_2$  norm and should be used if the points are influenced by Gaussian noise, while the median that is equivalent to minimizing the  $L_1$  norm should be used for data with outliers. While Gaussian noise is typically correlated with the quality of the mechanical and optical components of the capture device, outliers can, for example, be caused by a stereogrammetry algorithm finding false matches. The range scanner used in this study has very high quality components and the Gaussian noise is found to be negligible, but some outliers are present due to the stereogrammetry algorithm. In the following, the  $L_1$  norm is, therefore, used. An approximate surface can be extracted from the distance field using a standard Bloomenthal isosurface extractor [4]. However, at this point, the surface is neither smooth nor does it fill holes in a satisfactory way. The next section describes how regularization is used to create improved surfaces.

#### 3.3 Markov Random Field Regularization

To obtain a better and more faithful surface reconstruction, the problem of finding the distance field is cast in a Bayesian framework of Markov Random Field restoration. In the following, the value at voxel indexed with  $i$  is described by a random variable  $D_i$ . The probability of voxel  $i$  having a specific value  $d_i$  is, therefore,  $p(D_i = d_i)$ . In the following, the event  $p(D_i = d_i)$  is abbreviated  $p(d_i)$ . Furthermore, the entire distance field is described by a multivariate random variable  $\mathbf{D}$  with the same number of elements,  $N$ , as voxels in the field. The event  $p(D_1 = d_1, \dots, D_N = d_N)$  is, therefore, written as  $p(\mathbf{D} = \mathbf{d})$ . However, in the following, the abbreviation  $p(\mathbf{d})$  is used for this event.



The four successive stages of the Bayesian paradigm are followed:

1. Construction of a prior probability distribution  $p(\mathbf{d})$  for the distance field  $\mathbf{D}$ .
2. Formulation of an observation model  $p(\mathbf{d}^o|\mathbf{d})$  that describes the distribution of the observed distances  $\mathbf{D}^o$  given any particular realization of the prior distribution.
3. Combination of the prior and the observation model into the posterior distribution by Bayes theorem

$$p(\mathbf{d}|\mathbf{d}^o) = p(\mathbf{d}^o|\mathbf{d})p(\mathbf{d})/p(\mathbf{d}^o). \quad (1)$$

$p(\mathbf{d}^o)$  is the prior probability of the observed distance field. This term is constant and can be ignored in the optimization.

4. Drawing inference based on the posterior distribution.

Here the observed distances  $\mathbf{D}^o$  are the initial estimates of the distances in the distance field. In other words, the goal is to compute the distance field  $\hat{\mathbf{d}}$  that maximizes the posterior probability

$$\hat{\mathbf{d}} = \arg \max_{\mathbf{d}} p(\mathbf{d}|\mathbf{d}^o). \quad (2)$$

To describe a probability distribution on a spatial arrangement of voxels some useful definitions from graph theory are necessary.

Given a graph of connected sites  $S = \{s_1, s_2, \dots, s_N\}$ . A neighborhood system  $\mathcal{N} = \{N_s, s \in S\}$  is any collection of subsets of  $S$  for which 1)  $s \notin N_s$ , and 2)  $r \in N_s \Leftrightarrow s \in N_r$ .  $N_s$  are the neighbors of  $s$ . A clique  $C$  is a subset of sites  $S$  for which every pair of sites are neighbors. In the following, the distance field  $\mathbf{D}$  is treated as an undirected graph, where  $i \sim j$  is used to denote that voxel  $i$  and voxel  $j$  are neighbors. Note that voxels are indexed by simple numbering instead of the cumbersome  $(x, y, z)$  indexing. With these definitions the probability distribution of any family of random variables indexed by  $S$ , i.e.,  $\mathbf{D} = \{D_s | s \in S\}$  can be considered.

Let  $\Omega$  denote the set of all possible configurations  $\Omega = \{\mathbf{d} = \{d_i\}_{i=1}^N | d_i \in \mathbb{R}\}$ . A random field  $\mathbf{D}$  is a Markov Random Field with respect to  $\mathcal{N}$  iff

1.  $p(\mathbf{d}) > 0 \forall \mathbf{d} \in \Omega$ ,
2.  $p(d_s | d_r, r \neq s) = p(d_s | d_r, r \in N_s) \forall s \in S$ ,

The first constraint is the positivity condition and can be satisfied by specifying a neighborhood large enough to encompass the Markovianity condition in the second constraint. Although the second condition is on the state of neighboring sites only, it does not exclude long-range correlations in the probability distribution over the entire graph. In our case the second condition translates into that the probability of the value of a given voxel  $p(d_i)$  is only dependent on the values at the neighboring voxels.

Given a neighborhood system  $\mathcal{N} = \{N_s\}$ , let all cliques be denoted by  $\mathcal{C}$ . For all  $C \in \mathcal{C}$  it is assumed that a family of potential functions  $V_C$  exists. An energy function for any given configuration of  $\mathbf{D}$  can now be defined:

$$U(\mathbf{d}) = \sum_{C \in \mathcal{C}} V_C. \quad (3)$$

This leads to the definition of the Gibbs measure induced by the energy function  $U(\mathbf{d})$ :

$$p(\mathbf{d}) = \exp(-U(\mathbf{d})). \quad (4)$$

Normally, a partition function  $Z$  and a temperature  $T$  is used in (4). However, for our purpose  $Z$  cancels out,  $T$  is considered constant, and they are, therefore, both ignored in the following. The Hammersley-Clifford theorem gives the relation between MRF and Gibbs random fields and states that  $\mathbf{D}$  is a Markov random field with respect to  $\mathcal{N}$  iff  $p(\mathbf{d})$  is a Gibbs distribution with respect to  $\mathcal{N}$  [15], [33]. Later it will be described, how potentials that induce the Gibbs measure are defined and thus MRF properties of  $\mathbf{D}$  on the graph are encompassed.

MRFs are often specified with one-dimensional finite state spaces. However, the voxel values in the distance field are continuous. Luckily, it can be shown that MRFs generalize to multivariate continuous distributions as described in, for example, [16], [17].

### 3.3.1 Prior Models

The prior probability distribution  $p(\mathbf{d})$  should capture the general and scene-specific knowledge. In our case, the prior should describe the general properties of the surface we wish to reconstruct. As will be shown later, it can, for example, be if the surface acts like a membrane. As indicated earlier, the local prior distribution for a given voxel value should be based on the values of the surrounding voxels or more formally the voxels that share cliques with the current voxel. As a consequence of (4) the local probabilities are formulated as energy functions. A classic and much used prior is based on differences between neighbors [13]. Using the  $L_2$ -norm, the energy function for this prior is

$$U_G(d_i) = \frac{1}{n} \sum_{i \sim j} (d_i - d_j)^2, \quad (5)$$

where  $n$  is the number of neighboring voxels. In this paper, a 6-neighborhood system is used, so  $n = 6$ . This energy function has a simple maximum likelihood (ML) estimate and using this prior is equivalent to minimizing the membrane energy [34]. Plugging this energy function into (4) gives  $p(d_i) = \exp(-U_G(d_i))$  that obviously induces a standard Gaussian distribution on the voxel values. An alternative prior is based on differences between neighboring Laplacians:

$$U_{\mathcal{L}}(d_i) = \sum_{i \sim j} (\mathcal{L}(d_i) - \mathcal{L}(d_j))^2, \quad (6)$$

where  $\mathcal{L}(d_i)$  is the Laplacian estimated at voxel  $i$ . This choice is inspired by the method described in [11]. A discrete approximation of the Laplacian is used:

$$\mathcal{L}(d_i) = \frac{1}{n} \sum_{i \sim j} (d_i - d_j). \quad (7)$$

Obviously, this prior favors distance fields with smooth Laplacians. Since the Laplacian of a distance field is proportional to its mean curvature [35], the prior in (6) favors distance fields with small variations in mean curvature. With this approximation, it can be seen that

the energy function in (6) is defined using a set of cliques based on a neighborhood system where two voxels  $i$  and  $j$  are neighbors if  $(\mathbf{p}_i - \mathbf{p}_j)^2 \leq 4$ , where  $\mathbf{p}_i$  is the  $(x, y, z)$  index of voxel  $i$ .

### 3.3.2 Observation Model

Given a realization of the prior distribution, the observation model  $p(\mathbf{d}^o|\mathbf{d})$  describes the conditional distribution of the observed data  $\mathbf{D}^o$ . In other words, given the true distance field  $\mathbf{d}$ ,  $p(\mathbf{d}^o|\mathbf{d})$  is the distribution of the observed (measured) distance field. If, for example, the distance field is created based on the scanning of a perfect sphere, the *true distance field* is how the distance field would optimally be without any noise influence, etc. The observation model describes how deviations from *the true distance field* caused by noise or mechanical problems are distributed. The observation model should, therefore, reflect the physical or statistical properties of the capture device. In our case, the data are acquired using a ranger scanner, where it is possible to assign measures of confidence to each point. By specifying the observation model, we may favor a distance map that is true to the initial distance map in regions with high data confidence. As with the prior models, the observation model is formulated as an energy function. The observation model energy used is:

$$U_{\text{obs}}(d_i^o) = (d_i - d_i^o)^2, \quad (8)$$

where  $d_i^o$  is the original distance at voxel  $i$ . Using (4) with this energy function leads to  $p(d_i^o|d_i) = \exp(-U_{\text{obs}}(d_i^o))$ , that is, a Gaussian distribution with mean  $d_i$ .

### 3.3.3 Local Confidence Estimates

A local confidence measure  $\alpha_i \in [0 : 1]$  is used to balance the prior and the observation model. It is based on the Euclidean distance from the voxel center to the nearest input point  $d_i^E$ . Here  $\alpha_i = 1 - \min(d_i^E/d_{\text{max}}^E, 1)$ , where  $d_{\text{max}}^E$  is a user-defined maximum Euclidean distance. For the scans of the human faces, a value of  $d_{\text{max}}^E = 3\mu_l$  was found to be suitable. A discussion of other confidence measures can be found in [5].

### 3.3.4 Maximum a Posteriori Estimate

We wish to compute the distance field that maximizes the a posteriori conditional probability  $\hat{\mathbf{d}} = \arg \max_{\mathbf{d}} p(\mathbf{d}|\mathbf{d}^o)$ . As will be shown, the maximization of the posterior probability is transformed into the minimization of the weighted sum of the energy functions for the prior distribution and the observation model.

Using the Markovianity assumption and Bayes' rule, we can write the a posteriori probability of a single voxel as a product of the probability given by its prior and the observation model:

$$p(d_i|d_r, r \in N_i) = p(d_i)p(d_i^o|p_i),$$

where  $p(d_i^o)$  is left out since it is constant. Using the energies (5) or (6) and (8), we obtain

$$\begin{aligned} p(d_i|d_r, r \in N_i) \\ = \exp(-\alpha_i\beta U_{\text{obs}}(d_i) - (1 - \alpha_i\beta)U_{\text{prior}}(d_i)), \end{aligned}$$

where the global weight  $\beta$  and the local  $\alpha_i$ s are used to balance the prior and observation models. The ML estimate of the voxel value  $d_i$  can be computed as:

$$\begin{aligned} \frac{\partial \ln p(d_i|d_r, r \in N_i)}{\partial d_i} \\ = \frac{\partial(-\alpha_i\beta U_{\text{obs}}(d_i) - (1 - \alpha_i\beta)U_{\text{prior}}(d_i))}{\partial d_i} = 0. \end{aligned}$$

If the prior from (5) is used, the solution is:

$$d_i = \alpha_i\beta d_i^o + (1 - \alpha_i\beta) \frac{1}{n} \sum_{i \sim j} d_j. \quad (9)$$

The linear solution is due to the quadratic nature of the energy functions used. Furthermore, the solution is also found to be a linear combination of neighboring voxel values and the original distance estimate if the prior from (6) is used. Details can be found in Appendix. The next section describes how three different approaches can be used to compute the global solution.

## 3.4 Optimization

As seen in the previous section, the maximization of the global probability is equal to minimizing the global energy. Three different approaches have been tested.

### 3.4.1 Multiscale ICM

The Markovianity assumption states that local probabilities are defined using only a near neighborhood. A simple method to find the global optimum is to iteratively assign each site its local maximum likelihood estimate ((9) for the energy term in (5)). This approach is called the Iterative Conditional Modes (ICM) algorithm [14]. The visiting order is determined by a permuted index vector, ensuring that all voxels are visited for each iteration, but in a random order. This is done to avoid propagation of trends. In each iteration the total root mean square (RMS) change of voxel values is observed and the algorithm stops when this value drops below a fixed threshold. Alternatively, the algorithm stops after a fixed number of iterations. The ICM algorithm is equal to the Gauss-Seidel method for determining solutions of linear systems.

Due to the uniform space division, a multiscale ICM solver is fairly easy to implement. Initially, the solution is found in a small voxel volume and the results are propagated to the next level using linear interpolation. In each step, the side lengths of the voxel volume are doubled. Furthermore, a new initial distance estimate is recomputed for the voxels in a narrow band around the input points at each level. In the current implementation, the maximum side length of the initial volume is 16 and the typical number of levels used is four.

### 3.4.2 Conjugate Gradient

The Conjugate Gradient method [36], [37] used to solve unconstrained optimization problems can also be applied to our problem of finding the minimum of the global energy:

$$E_g = \sum_{i=1}^N -\ln p(d_i|d_r, r \in N_i) = \quad (10)$$

$$\sum_{i=1}^N (\alpha_i\beta U_{\text{obs}}(d_i) + (1 - \alpha_i\beta)U_{\text{prior}}(d_i)). \quad (11)$$

The main observation is that, due to the Markovianity condition, the partial derivative of  $E_g$  with respect to a

single voxel value  $d_i$  only depends on the change of the local energy terms involving the neighbors of  $d_i$ :

$$\frac{\partial E_g}{\partial d_i} = \frac{\partial E_i}{\partial d_i} + \sum_{i \sim j} \frac{\partial E_j}{\partial d_i}, \quad (12)$$

where  $E_i = \alpha_i \beta U_{\text{obs}}(d_i) + (1 - \alpha_i \beta) U_{\text{prior}}(d_i)$ . It is, therefore, reasonably quick to compute the gradient of the total energy  $\nabla E_g = (\frac{\partial E_g}{\partial d_1}, \dots, \frac{\partial E_g}{\partial d_N})$ . In our experiments, we have used a standard Fletcher-Reeves-based conjugate gradient implementation. Details can be found in [37]. In addition, the method is implemented in a multiscale scheme as described in Section 3.4.1. While it is simple to compute  $\frac{\partial E_g}{\partial d_i}$  for the energy in (5), it is quite complicated for the energy in (6) due to the larger neighborhood. Experiments showed that the multiscale ICM is superior to multiscale Conjugate Gradients for the energy formulation in (6).

### 3.4.3 Sparse Cholesky Factorization

An alternative approach is to formulate the problem as a large sparse linear system, since (9) can be rearranged into:

$$\frac{n_i d_i}{1 - \alpha_i \beta} - \sum_{i \sim j} d_j = \frac{n_i \alpha_i \beta}{1 - \alpha_i \beta} d_i^o, \quad (13)$$

that can be cast into the linear system  $\mathbf{A} \mathbf{x} = \mathbf{b}$ :

$$\begin{bmatrix} \frac{n_1}{1 - \alpha_1 \beta} & -1 & \dots & -1 & \dots \\ -1 & \frac{n_2}{1 - \alpha_2 \beta} & -1 & \dots & \\ \vdots & & & & \\ -1 & & & & \\ \vdots & & & & \ddots \end{bmatrix} \mathbf{x} = \begin{bmatrix} \frac{n_1 \alpha_1 \beta}{1 - \alpha_1 \beta} d_1^o \\ \frac{n_2 \alpha_2 \beta}{1 - \alpha_2 \beta} d_2^o \\ \vdots \\ \frac{n_N \alpha_N \beta}{1 - \alpha_N \beta} d_N^o \end{bmatrix},$$

where  $x_i = d_i$  and  $\mathbf{A}$  is a sparse tridiagonal matrix with fringes [38] having dimensions  $N \times N$ , where  $N$  is the number of voxels. The number of neighbors of a voxel determines the number of  $-1$  in each row in  $\mathbf{A}$ . The column indexes of the  $-1$  depend on the ordering of the voxel volume. In our case, the index is computed as  $i = x_t + y_t \cdot N_x + z_t \cdot N_x \cdot N_y$ , where  $(x_t, y_t, z_t)$  are the voxel displacement compared to the current voxel and  $(N_x, N_y, N_z)$  is the volume dimensions. Some special care is needed for edge and corner voxels that do not have six neighbors. For the energy term in (5), the maximum entries in a row are seven, which makes the system very sparse. Furthermore,  $\mathbf{A}$  is symmetric and positive definite making it suitable for sparse Cholesky decomposition. A standard sparse Cholesky solver (CHOLMOD) is used to solve the system [39]. Our experiments have demonstrated that the sparse Cholesky solver works well when the number of voxels in the volume is less than 200,000. On a standard PC with 2 GB of memory, the solver runs out of memory for larger volumes. However, the sparse Cholesky approach is well suited for computing the initial level used in the multiscale ICM solver described in the previous section. In the following, the results using the energy term in (5) are produced using the Sparse Cholesky solver to compute the initial estimate and the multiscale ICM solver is used for the next levels. With this approach, the

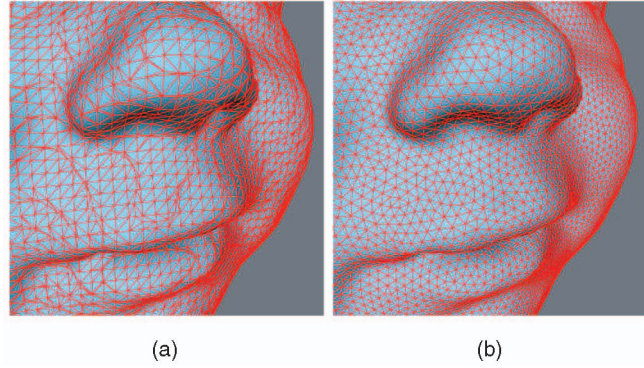


Fig. 2. (a) The result of the Bloomenthal polygonizer and (b) the remeshed result.

estimation and regularization of the distance field is done in less than two minutes for a final voxel volume of (150, 150, 150) on a standard dual core, 2.4 GHz, 2 GB RAM PC. For the energy in (6), the pure multiscale ICM solver is used, due to the difficulty in constructing  $\mathbf{A}$ . The solver is approximately 50 percent slower with (6) than with (5).

### 3.5 Meshing

The Bloomenthal polygonizer [4] is used to extract the isosurface from the MRF regularized distance field. Furthermore, a mesh-optimization scheme is applied to achieve a better triangulation. The method used is a modified version of the technique described in [40]. The modification is that we use the distance field for estimating surface tangents and reprojection of points. Initially, a goal edge length  $l$  is computed as the median value of the existing edge lengths. The approach consists of the following steps:

1. Split all edges that are longer than  $\frac{4}{3}l$  at their midpoint.
2. Collapse all edges shorter than  $\frac{4}{5}l$  into their midpoint.
3. Optimize vertex valence to 6 by edge flipping.
4. Weighted tangential relaxation of all vertices.
5. Backproject each vertex to the zero-level surface of the distance field.

The above steps are repeated 10 times. In the tangential relaxation step, the vertex  $\mathbf{p}$  is moved to  $\mathbf{p}' = \mathbf{q} + (\mathbf{nn}^T(\mathbf{p} - \mathbf{q}))$ , where  $\mathbf{n}$  is the surface normal estimated from the distance field and  $\mathbf{q}$  is the weighted barycenter of the neighbor vertices of  $\mathbf{p}$ . The weighted barycenter is calculated by weighting each neighbor with the area of their one-ring neighbors. Backprojecting a point to the zero-level surface of a distance field is trivial, since the distance field provides (with some computation) both the distance and the direction to the zero level at any given point in space. However, as mentioned earlier our distance field does not strictly enforce the unit gradient and, therefore, this method is not accurate. Experiments showed that a simple Runge-Kutta [38] like vector field tracing approach efficiently locates the closest zero-level location for a given point in space. Inspecting distributions of both edge lengths and triangle areas shows that they both follow a Gaussian distribution with a much smaller variance than before remeshing. An example of the result of the remeshing can be seen in Fig. 2, where clear improvements in both vertex valence and triangle uniformity is seen.



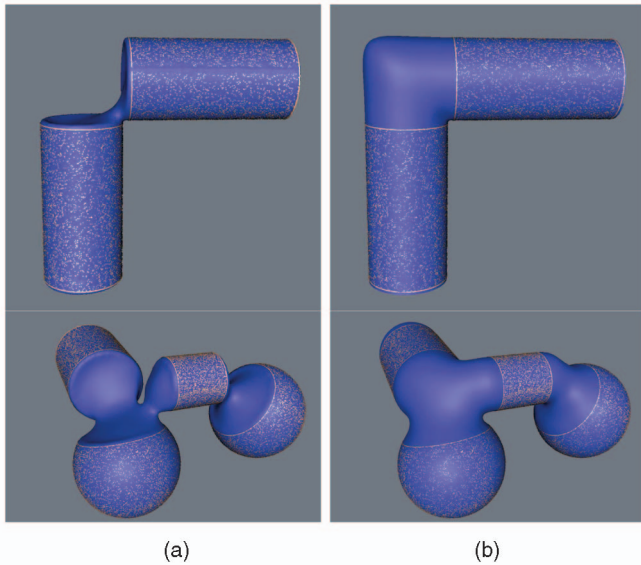


Fig. 3. Surface reconstruction on an artificial data set. The points are used as input. (a) The results using the energy term in (5) and (b) with (6).

## 4 RESULTS

In order to visualize the behavior of the different prior models, the method has been tested on two artificial data sets. The first consists of points randomly sampled over two separated tubes and the second is points randomly distributed over a more complex collection of primitives. There is a clear difference in the results when using the two different prior models as seen in Fig. 3. Fig. 3a clearly shows that the result of using (5) is equal to stretching a membrane over the point cloud. The higher order prior in (6) results in a surface where there are no abrupt changes in the curvature of the surface as seen in Fig. 3b.

The real data used to evaluate the method consist of 3D scans of human heads made using a 3dMD Cranial scanner. The scanner uses structured light and multiple cameras to capture stereo images of the head. Second, a proprietary stereogrammetry algorithm is used for extracting the 3D positions of the points on the head. As input to our method, the raw points found by the 3dMD stereogrammetry algorithm are used. Since the data typically consist of the human face and not the whole head, the surface to be reconstructed is not topologically equivalent to a sphere, but is a patch with open boundaries. Defining an inside and an outside is, therefore, ambiguous. In practice, our method does not have problems with this ambiguity. One aim is to use the resulting surfaces for 3D analysis of facial morphology [41] and boundary element modeling of sound fields [42]. It is, therefore, important that missing parts of the anatomy are reconstructed in an anatomically plausible way.

In Fig. 4, a surface reconstructed using the described MRF method with the prior in (6) is seen. The surface is smooth and interpolates the missing parts well. Especially, the area under the chin is reconstructed well. Furthermore, the algorithm is feature preserving, as can be seen in areas with high curvature, for example, around the nose. The local weights,  $\alpha_i$ , are visualized by projecting them on the reconstructed surface. It is seen that areas with underlying data are skin colored, meaning that the model is favoring the

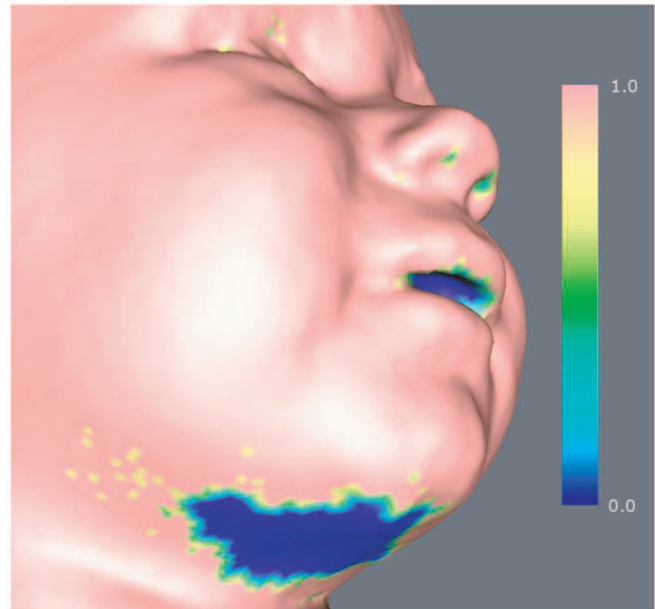


Fig. 4. Result of the MRF surface reconstruction using the prior in (6). The local voxel weights  $\alpha_i$  are shown on the resulting isosurface. Here, 1 (skin color) is complete confidence in original samples and 0 (blue) is no confidence. The blue part of the surface is, therefore, interpolated.

original distance estimate and the surface, therefore, follows the input points closely. The blue part of the surface has been regularized more since the model used here is closer to a pure prior model. As expected, the blue parts of the surface are shaped so the variation in curvature is minimized while at the same time the surface connects smoothly to the skin-colored areas. The example seen in Fig. 4 was created using a volume with uniform voxel side lengths of 1.8 mm and dimensions (121, 121, 105). There are 28,000 input points with an average neighbor distance of 1.1 mm. The surface reconstruction took 100 seconds. A value of  $\beta = 0.9$  was used. As will be described later, the choice of  $\beta$  is mostly critical when the input points are influenced by noise.

The robustness of the algorithm is tested by inspecting the generated surfaces. In all our test cases, no degenerate triangles or nonmanifold edges are found. Furthermore, it can be proved that, if the scalar field contains no singularities, the isosurface will be a manifold [43].

As described earlier, the choice of prior model should be based on knowledge of the nature of the surface. The prior models described in (5) and (6) have been tested by applying them to data with a large hole. The result can be seen in Fig. 5. As can be seen the prior from (6) seems well suited for this type of data. Furthermore, it can be seen that the results from the Poisson surface reconstruction algorithm [8] are similar to the results based on (5) that minimizes a membrane energy. In the following, the results are based on the prior from (6).

The accuracy of the reconstructed surface is estimated by computing the distances from the input points to the nearest points on the surface. The accuracy is expressed as the RMS value of the point distances.

The hole-filling capabilities are estimated by initially extracting a set of points from the input data and then fitting the surface to the remaining points. An example input data set, where some points are removed, can be seen



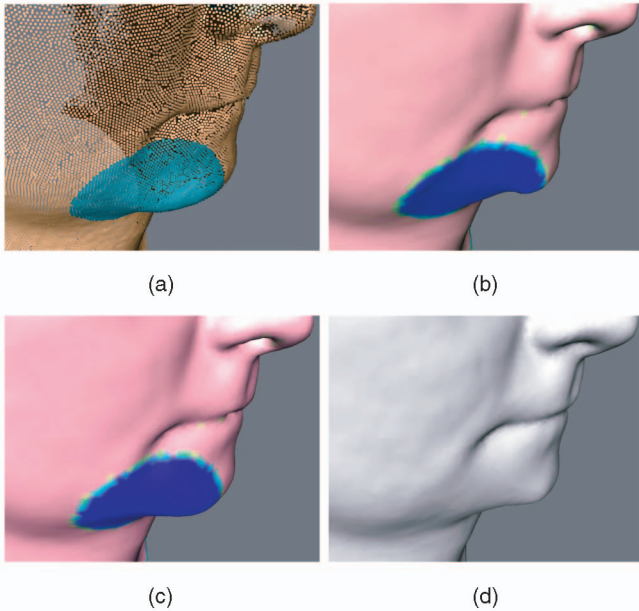


Fig. 5. Comparison of different priors. (a) The input data, where the blue points are removed to test hole filling. (b) The result of the MRF algorithm using the prior in (5). (c) The result using the prior in (6), and (d) the result of the Poisson surface reconstruction.

in Fig. 6. The gray points are used as input to the surface reconstruction. Second, the distances from the removed points to the reconstructed surface are calculated, as can be seen as the color-coded points in Fig. 6. Finally, a measure of the hole-filling capabilities is calculated as the RMS value of these distances.

The presented algorithm has been tested on 10 scanned human faces and the above measures of accuracy and hole-filling capabilities computed. Furthermore, the data have also been used as input to the multilevel partition of unity implicit method (MPU) presented in [7] (default parameters) and the Poisson surface reconstruction method (Poisson) by Kazhdan et al. [8] (default parameters and

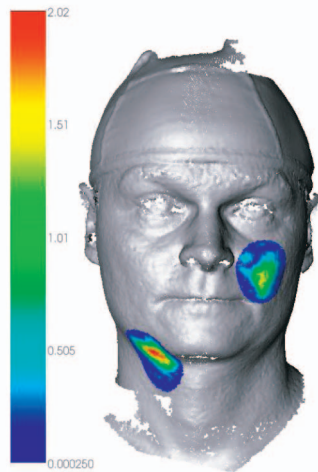


Fig. 6. The data used for testing hole filling. The point-to-surface distances are calculated using the points that were left out of the surface reconstruction. The points are color coded according to their distance to the resulting surface (blue to red). The gray points are used as input to the surface reconstruction.

TABLE 1  
Results from the Presented Algorithm and Two Alternative Surface Reconstructors Computed on 10 Scans

#	Points	MRF		Poisson		MPU	
		Acc	HF	Acc	HF	Acc	HF
1	69030	0.13	0.78	0.47	1.23	0.13	1.91
2	60396	0.15	0.83	0.38	2.58	0.13	2.28
3	57106	0.13	0.57	0.39	3.01	0.13	1.27
4	55693	0.15	0.65	0.50	2.62	0.14	3.39
5	62661	0.15	1.37	0.46	3.41	0.14	1.85
6	34466	0.14	0.25	0.56	0.43	0.13	1.42
7	32468	0.13	0.54	0.38	1.20	0.12	1.43
8	48056	0.15	0.38	0.44	0.49	0.14	1.14
9	77595	0.14	0.71	0.43	1.60	0.12	2.03
10	57956	0.14	0.42	0.47	1.32	0.12	2.09
Average		0.14	0.65	0.45	1.79	0.13	1.88

Accuracy (Acc) and hole filling (HF) are in RMS (mm) and lower is better.

depth = 11). The results can be seen in Table 1. It can be seen that the presented MRF method together with the MPU method has excellent accuracy of around 0.13 mm, while the Poisson method is consistently around 0.42 mm. The accuracy of all the methods is influenced by the choice of spatial resolution. The spatial resolution is difficult to assess for the Poisson and MPU method due to their octree-based implementation. The average volume size for the MRF results is (120,120,120) and the average voxel side length is 2.5 mm.

Surfaces reconstructed with the three methods can be seen in Fig. 7, where it can clearly be seen that they all do an

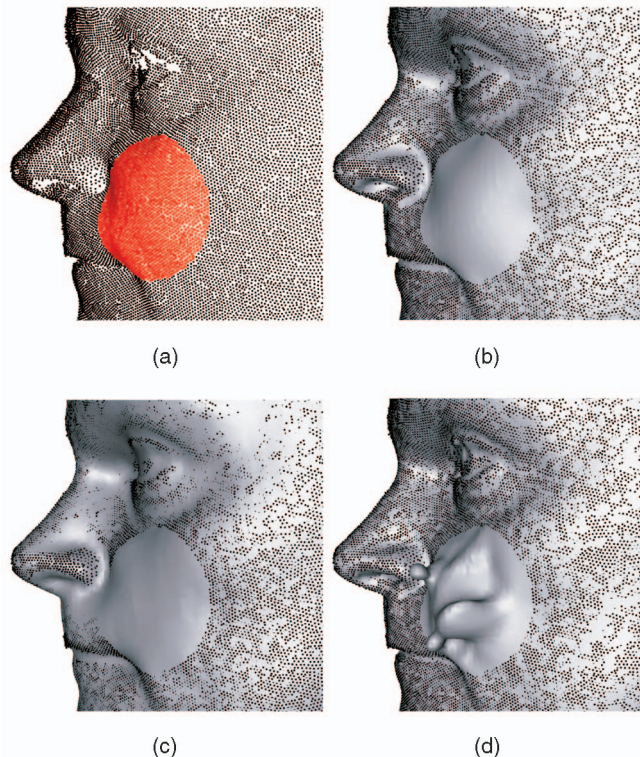


Fig. 7. Comparison of three different algorithms. (a) The input data, where the red points are removed to test hole filling. (b) The result of the MRF algorithm. (c) The result of the Poisson algorithm and (d) the MPU result.

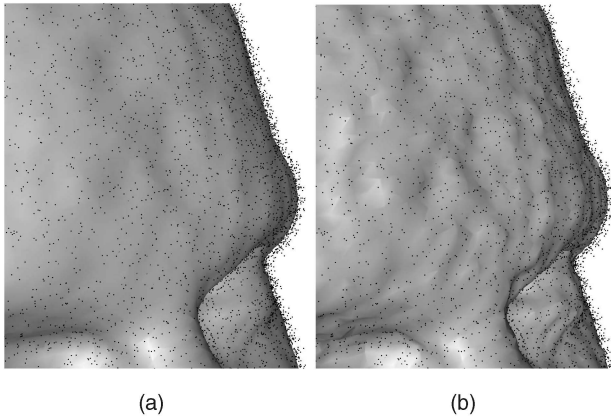


Fig. 8. Results with Gaussian noise ( $\sigma = 1$  mm). In (a) more weight is on the prior model ( $\beta = 0.1$ ). In (b) the observation model is favored ( $\beta = 0.5$ ) causing a more bumpy result.

excellent job in fitting the surface where there is good input data. The difference between the methods manifests itself in the hole filling. Notably, the MPU method has not succeeded in filling the hole in a plausible way. This is probably caused by the method used to fit the local quadratic functions, where the artifacts seen are due to the unconstrained tails of a set of quadratic functions. As described earlier, the MRF method with (6) fills the hole in a way that minimizes the variation in curvature while at the same time smoothly connecting the edges. The Poisson algorithm hole filling should be more closely related to a membrane stretched over the hole. As seen in Table 1, the MRF approach is superior with regards to hole filling with data of this nature. In contrast to the accuracy, the hole-filling capabilities are determined by the underlying principles of the method and less by the choice of spatial resolution.

The average computation times for the results in Table 1 are 20 seconds for the Poisson method, 24 seconds for the MPU method, 100 seconds for the MRF method with the prior in (6) and 70 seconds for the MRF method with the prior in (5). The proposed method is, therefore, not as fast as the state-of-the-art algorithms. However, several optimizations can be applied that will at least halve the computation time.

The noise robustness has been tested by adding Gaussian-distributed noise to a point cloud and visually inspecting the resulting surfaces. As can be seen in Fig. 8, the global parameter  $\beta$  can be used to adjust the balance between the degree of smoothing and the degree of trust in the data. The optimal estimation of  $\beta$  is far from trivial since it partly depends on the nature and scale of the noise. However, methods like *L-curve* analysis could probably be used to determine the regularization parameter  $\beta$  [44]. In this paper, visual inspection has been used to determine  $\beta$ .

The discrete approximation of the energy functions can induce a grid bias. A detailed analysis of metrication effects can be found in [45], where the  $L_2$  norm used in a binary labeling problem is approximated using higher order terms. In our case, the most relevant question is the rotation invariance of the surface reconstruction. In particular, in regions where the prior model is dominant. To test the rotation invariance, an artificial data set is created. This consists of points sampled randomly on a sphere with radius 40. A cut has been made in the sphere as seen in

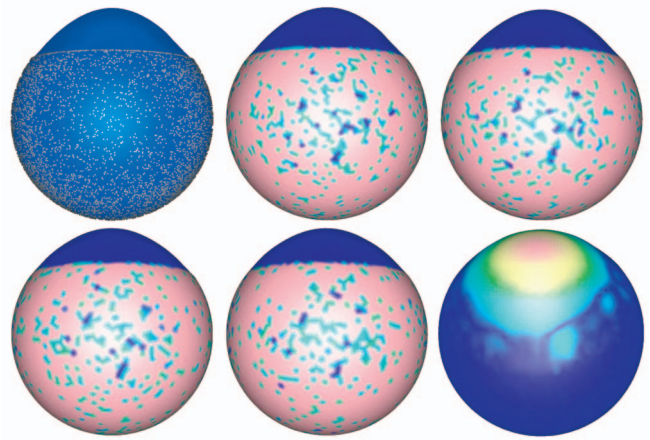


Fig. 9. From top left right to lower right. Input points generated from a cut sphere with radius 40 (together with a reconstructed surface). Results after input points were rotated 0, 10, 30, and 45 degree. The difference between the result from the 0 and 45 degree rotated input points.

Fig. 9 (top left). These points are then used as input to the MRF-based surface reconstruction. The point cloud is then rotated around a non-axis-aligned axis and the surface is reconstructed again. The resulting surface is rotated back using the inverse rotation. The surfaces can be seen in Fig. 9, where the point cloud has been rotated 0, 10, 30, and 45 degree. The surface is color coded so blue means a pure prior model and pink is a pure observation model. The patches in the pink part are the result of the random spacing of the point cloud. It is observed that for a rotation of 0 degree the part of the surface that consists of pure prior model is somewhat pointy, while for a rotation of 45 degree it is approximating a sphere. The last case is a vague indication that the Willmore energy is minimized by the approach. The maximum difference between the surfaces is 2.47 as seen in the lower right corner of Fig. 9. This error should be compared to the sphere radius of 40. However, the experiment indicates that our approach is not completely rotation invariant. A future solution could be to use higher order approximations of the energy functions.

## 5 SUMMARY AND CONCLUSIONS

A novel surface reconstruction method is proposed. The novelty is primarily due to Markov Random Field based regularization of a distance field. The Markov Random Field framework allows us to express the probability of a voxel value in terms of both a prior, which describes how likely the voxel is given its neighbors, and a data term that describes how likely it is given the original (observed) value. This is a great advantage over previous work since it allows us to integrate knowledge about data and desired properties of the distance field with greater ease and orthogonality.

The method has been tested using 3D scans of human faces captured in a real clinical environment. The algorithm proved to have excellent accuracy and hole-filling capabilities compared to two state-of-the-art algorithms. In conclusion, the presented method has the following benefits when compared to existing methods:

- **Robust and accurate:** The generated surfaces are of high quality in terms of both accuracy and geometrical



properties. Furthermore, the noise sensitivity can be handled by adjusting a single parameter.

- **Hole filling:** The algorithm has excellent hole-filling capabilities. The nature of the hole filling can be adapted using the prior energy term.
- **Flexible:** It is easy to incorporate knowledge of the nature of the surface to be reconstructed and information about the capture device in an orthogonal way.

Furthermore, the algorithm is reasonably fast. Future work will include speed optimization using a multicore implementation of the ICM algorithm. Furthermore, it will be tested if it is feasible to limit the updating of the distance field in a broad band around a previously estimated zero level of the distance field at the higher levels at the multiscale ICM. The software can be downloaded from [www.imm.dtu.dk/MRFSurface](http://www.imm.dtu.dk/MRFSurface).

## APPENDIX

In this appendix, it is demonstrated why the local maximum likelihood estimate using (6) can be found as a linear combination of voxel values. The Approximated discrete Laplacian is:

$$\mathcal{L}(d_i) = \frac{1}{n} \sum_{i \sim k} (d_i - d_k).$$

Partial differentiation with respect to center voxel value  $d_i$ :

$$\frac{\partial \mathcal{L}(d_i)}{\partial d_i} = 1. \quad (14)$$

Partial differentiation with respect to neighbor voxel value  $d_k$ :

$$\frac{\partial \mathcal{L}(d_i)}{\partial d_k} = -1/n. \quad (15)$$

Squared difference of Laplacians

$$U_{\mathcal{L}}(d_i) = \sum_{i \sim j} (\mathcal{L}(d_i) - \mathcal{L}(d_j))^2.$$

Partial differentiation:

$$\frac{\partial U_{\mathcal{L}}(d_i)}{\partial d_i} = 2 \sum_{i \sim j} \left( (\mathcal{L}(d_i) - \mathcal{L}(d_j)) \left( \frac{\partial \mathcal{L}(d_i)}{\partial d_i} - \frac{\partial \mathcal{L}(d_j)}{\partial d_i} \right) \right).$$

According to (14) and (15), it can be seen that:

$$\left( \frac{\partial \mathcal{L}(d_i)}{\partial d_i} - \frac{\partial \mathcal{L}(d_j)}{\partial d_i} \right)$$

is constant  $(1 + 1/n)$ , which leads to:

$$\frac{\partial U_{\mathcal{L}}(d_i)}{\partial d_i} = \sum_{i \sim j} (\mathcal{L}(d_i) - \mathcal{L}(d_j)) = 0,$$

for which the solution is a linear combination of voxel values in the neighborhood of  $d_i$ .

## ACKNOWLEDGMENTS

The authors would like to thank the Oticon Foundation for partly funding this work and the 3D Craniofacial Image

Research Laboratory at Copenhagen University for letting them use their scanner. They also thank Yutaka Ohtake and Michael Kazhdan for making their surface reconstruction algorithms publicly available. They would like to thank reviewers for their constructive and relevant feedback. Finally, they would like to thank Mario Botsch for explaining the inner workings of the Sparse Cholesky methods.

## REFERENCES

- [1] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Surface Reconstruction from Unorganized Points," *Proc. ACM SIGGRAPH*, pp. 71-78, 1992.
- [2] W.E. Lorensen and H.E. Cline, "Marching Cubes: A High Resolution 3D Surface Construction Algorithm," *Proc. ACM SIGGRAPH '87*, vol. 21, no. 4, pp. 163-169, July 1987.
- [3] G. Treece, R. Prager, and A. Gee, "Regularised Marching Tetrahedra: Improved Iso-Surface Extraction," *Computers & Graphics*, vol. 23, no. 4, pp. 583-598, 1999.
- [4] J. Bloomenthal, "An Implicit Surface Polygonizer," *Proc. Graphics Gems IV*, pp. 324-349, 1994.
- [5] B. Curless and M. Levoy, "A Volumetric Method for Building Complex Models from Range Images," *Proc. ACM SIGGRAPH*, pp. 303-312, 1996.
- [6] J.C. Carr, R.K. Beatson, J.B. Cherrie, T.J. Mitchell, W.R. Fright, B.C. McCallum, and T.R. Evans, "Reconstruction and Representation of 3D Objects with Radial Basis Functions," *Proc. ACM SIGGRAPH*, pp. 67-76, 2001.
- [7] Y. Ohtake, A. Belyaev, M. Alexa, G. Turk, and H. Seidel, "Multi-Level Partition of Unity Implicits," *Proc. Int'l Conf. Computer Graphics and Interactive Techniques*, pp. 463-470, 2003.
- [8] M. Kazhdan, M. Bolitho, and H. Hoppe, "Poisson Surface Reconstruction," *Proc. Symp. Geometry Processing*, pp. 61-70, 2006.
- [9] M. Kazhdan, "Reconstruction of Solid Models from Oriented Point Sets," *Proc. Third Eurographics Symp. Geometry Processing*, 2005.
- [10] O. Schall, A. Belyaev, and H. Seidel, "Error-Guided Adaptive Fourier-Based Surface Reconstruction," *Computer-Aided Design*, vol. 39, no. 5, pp. 421-426, 2007.
- [11] B. Jakobsen, J.A. Bærentzen, and N.J. Christensen, "Variational Volumetric Surface Reconstruction from Unorganized Points," *Proc. IEEE/EG Int'l Symp. Volume Graphics*, Sept. 2007.
- [12] S. Geman and D. Geman, "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 6, no. 6, pp. 721-741, Nov. 1984.
- [13] J. Besag, "Towards Bayesian Image Analysis," *J. Applied Statistics*, vol. 16, no. 3, pp. 395-407, 1989.
- [14] J. Besag, "On the Statistical Analysis of Dirty Pictures," *J. Royal Statistical Soc., Series B*, vol. 48, no. 3, pp. 259-302, 1986.
- [15] D. Geman, "Random Fields and Inverse Problems in Imaging," *Saint-Flour Lectures*, Springer-Verlag, pp. 113-193, 1990.
- [16] R. Larsen, K. Conradsen, and B.K. Ersbøll, "Estimation of Dense Image Flow Fields in Fluids," *IEEE Trans. Geoscience and Remote Sensing*, vol. 36, no. 1, pp. 256-264, Jan. 1998.
- [17] R.R. Paulsen and K.B. Hilger, "Shape Modelling Using Markov Random Field Restoration of Point Correspondences," *Proc. Information Processing in Medical Imaging*, vol. 2732, pp. 1-12, 2003.
- [18] G. Winkler, *Image Analysis, Random Fields and Markov Chain Monte Carlo Methods—A Mathematical Introduction*, second ed., vol. 27. Springer-Verlag, 2003.
- [19] G. Storvik, "Bayesian Surface Reconstruction from Noisy Images," *Proc. 28th Symp. Interface (Interface)*, vol. 28, pp. 91-9, 1997.
- [20] P. Jenke, M. Wand, M. Bokeloh, A. Schilling, and W. Strasser, "Bayesian Point Cloud Reconstruction," *Computer Graphics Forum*, vol. 25, no. 3, pp. 379-388, 2006.
- [21] J. Diebel, S. Thrun, and M. Brünig, "A Bayesian Method for Probable Surface Reconstruction and Decimation," *ACM Trans. Graphics*, vol. 25, no. 1, pp. 39-59, 2006.
- [22] Q. Huang, B. Adams, and M. Wand, "Bayesian Surface Reconstruction via Iterative Scan Alignment to an Optimized Prototype," *Proc. Fifth Eurographics Symp. Geometry Processing*, pp. 213-223, 2007.



- [23] N. Moretto and R. Frezza, "Bayesian Surface Reconstruction," *Proc. Second Int'l Symp. 3D Data Processing, Visualization and Transmission (3DPVT '04)*, pp. 235-241, 2004.
- [24] T. Tasdizen and R. Whitaker, "Higher-Order Nonlinear Priors for Surface Reconstruction," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 26, no. 7, pp. 878-891, July 2004.
- [25] H.-K. Zhao, S. Osher, and R. Fedkiw, "Fast Surface Reconstruction Using the Level Set Method," *Proc. IEEE Workshop Variational and Level Set Methods in Computer Vision*, 2003.
- [26] R. Whitaker, "A Level-Set Approach to 3D Reconstruction from Range Data," *Int'l J. Computer Vision*, vol. 29, no. 3, pp. 203-232, 1998.
- [27] J. Gomes and O. Faugeras, "Level Sets and Distance Functions," *Lecture Notes in Computer Science*, pp. 588-602, Springer, 2000.
- [28] J. Verdera, V. Caselles, M. Bertalmio, and G. Sapiro, "Inpainting Surface Holes," *Proc. Int'l Conf. Image Processing (ICIP '03)*, vol. 2, 2003.
- [29] U. Clarenz, U. Diewald, G. Dziuk, M. Rumpf, and R. Rusu, "A Finite Element Method for Surface Restoration with Smooth Boundary Conditions," *Computer Aided Geometric Design*, vol. 21, no. 5, pp. 427-446, 2004.
- [30] V. Caselles, G. Haro, G. Sapiro, and J. Verdera, "On Geometric Variational Models for Inpainting Surface Holes," *Computer Vision and Image Understanding*, vol. 111, no. 3, pp. 351-373, 2008.
- [31] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C.T. Silva, "Point Set Surfaces," *Proc. Conf. Visualization (VIS '01)*, pp. 21-28, 2001.
- [32] S. Fleishman, D. Cohen-Or, and C. Silva, "Robust Moving Least-Squares Fitting with Sharp Features," *Proc. Int'l Conf. Computer Graphics and Interactive Techniques*, pp. 544-552, 2005.
- [33] J. Besag, "Spatial Interaction and the Statistical Analysis of Lattice Systems," *J. Royal Statistical Soc., Series B*, vol. 36, pp. 192-236, 1974.
- [34] D. Terzopoulos, "Regularization of Inverse Visual Problems Involving Discontinuities," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 8, no. 4, pp. 413-242, July 1986.
- [35] M. Jones, J. Bærentzen, and M. Sramek, "3D Distance Fields: A Survey of Techniques and Applications," *IEEE Trans. Visualization and Computer Graphics*, vol. 12, no. 4, pp. 518-599, July/Aug. 2006.
- [36] J. Shewchuk, "An Introduction to the Conjugate Gradient Method without the Agonizing Pain," technical report, School of Computer Science, Carnegie Mellon Univ., 1994.
- [37] P. Frandsen, K. Jonasson, H. Nielsen, and O. Tingleff, "Unconstrained Optimization," third ed., technical report, Informatics and Mathematical Modelling, Technical Univ. of Denmark, Mar. 2004.
- [38] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge Univ. Press, 2002.
- [39] T.A. Davis and W.W. Hager, "Row Modifications of a Sparse Cholesky Factorization," *SIAM J. Matrix Analysis and Applications*, vol. 26, no. 3, pp. 621-639, 2005.
- [40] M. Botsch and L. Kobbelt, "A Remeshing Approach to Multi-resolution Modeling," *Proc. Eurographics/ACM SIGGRAPH Symp. Geometry Processing*, pp. 185-192, 2004.
- [41] P. Hammond, T. Hutton, J. Allanson, L. Campbell, R. Hennekam, S. Holden, M. Patton, A. Shaw, I. Temple, and M. Trotter, "3D Analysis of Facial Morphology," *Am. J. Medical Genetics*, vol. 126, no. 4, pp. 339-348, 2004.
- [42] B. Katz, "Boundary Element Method Calculation of Individual Head-Related Transfer Function. I. Rigid Model Calculation," *J. Acoustical Soc. of Am.*, vol. 110, pp. 2440-2448, 2001.
- [43] L. Velho, J. Gomes, and L. de Figueiredo, *Implicit Objects in Computer Graphics*. Springer, 2002.
- [44] P. Hansen, *Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion*. Soc. for Industrial Math., 1998.
- [45] M. Klodt, T. Schoenemann, K. Kolev, M. Schikora, and D. Cremers, "An Experimental Comparison of Discrete and Continuous Shape Optimization Methods," *Proc. 10th European Conf. Computer Vision: Part I*, pp. 332-345, 2008.



**Rasmus R. Paulsen** received the MSc(Eng) degree from the Technical University of Denmark in 1998, and the PhD degree in medical image analysis from the Technical University of Denmark in 2004. His employment experience includes research and development in the Danish hearing aid company Oticon A/S from 2004 to 2008. He is now an associate professor in the Department of Informatics and Mathematical Modelling at the Technical University of Denmark.



**Jakob Andreas Bærentzen** received the MSc and PhD degrees from the Technical University of Denmark in 1998 and 2003, respectively. He is now an associate professor in the Department of Informatics and Mathematical Modelling at the Technical University of Denmark. His research interests are mostly related to the representation, modeling, synthesis, and manipulation of 3D geometry (shapes) but his interests extend to other areas of graphics, in particular within the realm of real-time graphics. Many of his contributions have been in the field of volumetric shape representations and distance fields.



**Rasmus Larsen** received the MSc(Eng) degree from the Technical University of Denmark in 1991, and the PhD degree in statistical image analysis from the Technical University of Denmark in 1994. Until 1997, he was an assistant professor in the Department of Mathematical Modelling, Technical University of Denmark. He worked as an expert statistician for Novo Nordisk A/S until 1999. In 2000, he returned to the Technical University of Denmark as an associate professor in medical image analysis. Since 2008, he has been a full professor of image analysis in the Department of Informatics and Mathematical Modelling at the Technical University of Denmark. He is the leader of the Section for Image Analysis and Computer Graphics and the head of the department graduate school programme. His research interests are centered on modeling analysis of biological geometries with applications particularly in medicine and food technology.

► For more information on this or any other computing topic, please visit our Digital Library at [www.computer.org/publications/dlib](http://www.computer.org/publications/dlib).