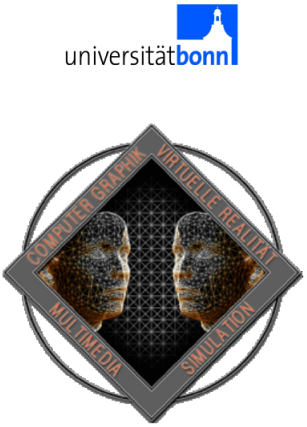

Smooth Signed Distance Surface Reconstruction

Seminar Report

Aljosa Osep
Mentor: Michael Weinmann

February 5, 2012

Contents

Chapter 1

Introduction

Surface reconstruction from (oriented) point samples is a very important and well-studied problem in computer graphics. There are various applications of surface reconstruction of real models in industry, archaeology and art, medical imaging and entertainment. For example, there is a need for digitized objects in 3D video games and movies. Scanning of mechanicals parts enables us to capture, analyse, visualize and even virtually modify them, do reverse engineering and physical simulations. Geometry reconstruction is essential also for digitization of cultural heritage and artistic works. It has also applications in medical imaging (visualization and segmentation of bones, tissues etc.).

1.1 Surface Reconstruction

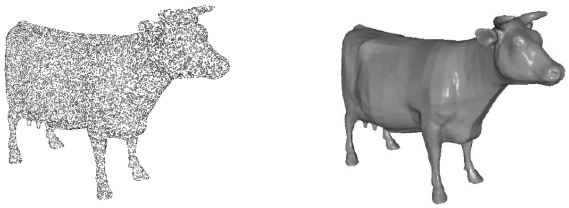

Figure 1.1: Oriented points and reconstructed surface (image credits: M. Kazhdan, M. Bolitho, H. Hoppe)

The aim of surface reconstruction is to construct as accurate approximation of a real surface (model) as possible. Several steps are necessary to obtain point clouds that are suitable for surface fitting. For capturing the full shape of an object, scans have to be taken from several views. Afterwards, these scans need to be registered into a common coordinate frame resulting in a point cloud describing the object.

However, in many applications it is desired to represent an object surface via a polygon-mesh instead of using point clouds. In this seminar, we consider surface reconstruction from oriented point samples. That means, that samples of a surface $s \in S$ have already been captured by one of the geometry acquisition approaches, such as structured light techniques, laser scanning or multi-view stereo methods. We also assume that scans are properly registered and additionally, normal information is assigned to each sample. This normal information could either be obtained in the acquisition or could be estimated from the point set. Given a dense point cloud the goal is to find a closed i.e. watertight surface for representing the object for which an example is shown in figure 1.1.

1.2 Challenges

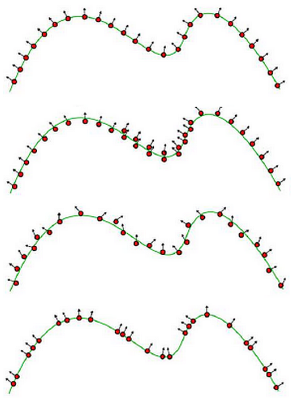

Figure 1.2: Uniform sampling, registration errors, noisy data and non-uniform sampling (image credits: F. Calakli, G. Taubin)

In the area of surface reconstruction, we are dealing with numerous challenges that arise from the development of inexpensive scanners and the acquisition procedure. There are various approaches to capture geometry (laser-range scanners, multi-camera systems, structured lighting systems, etc.) and even those that usually provide most accurate results (laser scanner) can completely fail on materials exhibiting complex reflectance behaviour. Furthermore, even with accurate laser scans, registration errors may occur. Many precise scans registered together can also result in very large data sets, so time efficiency and memory efficiency issues must be addressed. Due to a mixture of materials, some areas may not be well captured which results in holes in the scans. Surface reconstruction algorithms should reasonably fill such holes. Usually, scans are also noisy and sampling can be uneven, especially when using inexpensive setups to capture data (e.g.

structured light) or when dealing with challenging materials.

1.3 Overview of Methods

There is a large variety of techniques regarding the surface reconstruction problem. A survey on recent developments is given in [1].

In general, we can divide these methods into **interpolating** and **approximating** approaches. The first category aims at producing interpolating polygon meshes, i.e. all or just a subset of the given points are vertices in the resulting mesh. Many techniques of this group are combinatorial and e.g. Voronoi diagrams, Delaunay triangulations or alpha shapes can be used to partition the space based on the samples. Furthermore, some of these methods offer a guaranteed reconstruction quality [2].

Instead of producing meshes supported on the input points, there is also a large interest in getting approximating surfaces because interpolation of noisy data leads to overfitting. In this class, several techniques produce implicit surfaces which has the advantage that the implicit representation guarantees watertight surfaces. In the following, there is a list of some of the approaches:

- Poisson surface reconstruction [3]
- Streaming surface reconstruction using wavelets [4]
- Graph-cuts [5]
- Multi-level Partition of Unity Implicits [6]

For obtaining the approximation of the desired level set, the estimated implicit function is evaluated on a regular grid and subsequently, the isosurface is extracted via techniques such as Marching Cubes [7].

Additionally, the use of octree-based schemes [3, 4, 8, 9] helps in reducing computational and storage complexity.

1.4 Implicit Approach

The Poisson Surface Reconstruction method and the Smooth Signed Distance Surface Reconstruction method presented in this seminar fall into the category of implicit function fitting. This category can be divided further into **global** and **local** methods. While local methods (e.g. [10]) consider only a local neighbourhood of samples at surface points to estimate the surface, global methods (e.g. Radial Basis Function approaches, Graph-Cuts) produce a globally consistent solution.

The basic idea of the implicit approach is to fit an implicit function to the data having a value less than zero outside of the model and greater

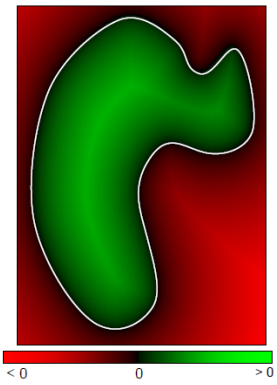

Figure 1.3: Implicit function and its zero set (image credits: M. Kazhdan, M. Bolitho, H. Hoppe)

than zero inside (see figure 1.3). Usually it is discretized on a regular voxel grid or adaptive data structure (e.g. octree). Then, the surface can be reconstructed by extracting its zero-level set using e.g. Marching Cubes or Dual-Contouring algorithms.

Chapter 2

Poisson Surface Reconstruction

The Poisson surface reconstruction method introduced in 2006 [3], has recently become one of most popular methods for surface fitting. This is mainly due to the availability of an implementation which is provided by the authors. There are also implementations suited for streaming [9] as well as parallel or GPU-based implementations [8, 11], which offer a drastic speed-up. The Poisson Surface Reconstruction belongs to the category of implicit function fitting, it is global, robust to noise, works well in case of non-uniform sampling and it is efficient in terms of memory consumption and speed.

We briefly present the method here for motivation and comparison. Beside the results, we also discuss similarities of this approach with regard to the SSD Reconstruction. In a nutshell, this method utilizes an octree and finds the implicit function by solving a Poisson equation.

2.1 Problem Statement

Both in the Poisson and the Smooth Signed Distance Reconstruction [12] approach the goal is to first reconstruct a 3D implicit function and then to obtain the reconstructed surface by extracting an appropriate isosurface. In particular, the Poisson method reconstructs an indicator function χ that has a value of one inside the model and zero outside, as it is shown in Figure 2.1.

The key idea lies, similarly as in the SSD Reconstruction, in the observation that an oriented point set $\vec{V}$ is in fact a sparse sampling of a continuous vector field, i.e. the gradient of the indicator function, $\nabla\chi$, should agree with the normal samples $s.n \in S$, where $s.n$ is the normal of the sample at the point $s.p$. Hence the problem seems to be solvable by the functional

equation $\nabla\chi = \vec{V}$. There is, however, a problem in this formulation. Its solution may not exist because not every vector field is a gradient of a function. A necessary condition for a vector field to be gradient of a function is to be conservative, i.e. its curl $\nabla \times \vec{V}$ has to be equal to zero.

Hence, the authors of both articles approach the problem by solving it in a least-square sense by finding an implicit function whose gradient approximates the vector field best. In case of the Poisson Reconstruction, the authors optimize the problem by solving the Poisson equation $\Delta\chi = \nabla \cdot \vec{V}$ by applying the divergence operator to $\nabla\chi = \vec{V}$ (that results in $\nabla \cdot \nabla\chi = \nabla \cdot \vec{V}$), since solving the Poisson equation amounts in finding a function whose gradient is closest to some prescribed vector field in the L_2 norm sense.

The main advantage of this formulation is that the result is a global solution, which is robust to noise in data. Unlike many global implicit reconstruction approaches such as approaches based on globally supported radial basis functions, this formulation admits locally supported functions and reduces to a sparse linear system.

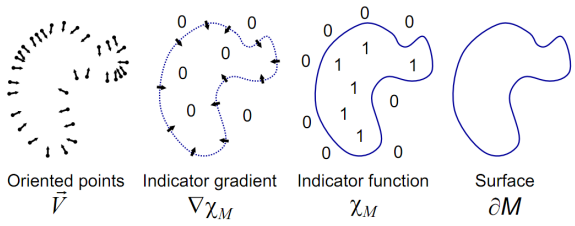

Figure 2.1: Illustration of the Poisson reconstruction in 2D

2.2 Implementation

A drawback of employing an indicator function χ is that it is discontinuous and its gradient does not exist at the sample points. The authors of the [3] address this problem by computing a smoothed indicator function $\widetilde{\chi}$ of the smoothed surface normal field, i.e. a discrete point set is used to compute an approximation of the continuous vector field. The following lemma formalizes the relationship between the gradient of a smoothed indicator function and the surface normal field:

$$\nabla(\chi_M * \widetilde{F})(q) = \int_{\partial M} \widetilde{F_p}(q)\vec{N}_{\partial M}(p)dp \tag{2.1}$$

Here, $\widetilde{F}(q)$ denotes a smoothing filter, $\widetilde{F}_p(q)$ represents $\widetilde{F}_p(q) = F(q-p)$, ∂M is the boundary of a solid M and $\vec{N}_{\partial M}(p)$ describes the inward surface

normal at point $p \in \partial M$. In [3], the proof is given.

The integral over the surface ∂M is approximated by a discrete summation of patches centered at the sample points $s \in S$ i.e. the integral over a patch P_s is approximated using the value at the coordinates $s.p$ which is scaled by the area $|P_s|$ of the patch. Thus, the computation of the vector field reduces to

$$\nabla(\chi_M * \widetilde{F})(q) \approx \sum_{s \in S} |P_s| \, \widetilde{F}_{s.p}(q) s.n \tag{2.2}$$

For time and space efficiency reasons, the authors of [3] decided to use a multi-resolution basis for representing the solution. An accurate solution to the Poisson equation is only necessary near the surface ∂M. Specifically, the authors discretize the problem on an octree O that is built on a point set S. Hence, the function space consists of base functions F_o, chosen to be the n-th convolution of the box filter (which approximates the Gaussian as n is increased). Basis functions, associated with each leaf node $o \in O$ of the octree, are translated to the center of the node, afterwards scaled with respect to the size of the node and have a large support in the local neighbourhood. Therefore, the vector field can be computed as the linear combination of these (smooth) basis functions which are associated with the octree cells:

$$\vec{V}(q) \equiv \sum_{s \in S} \sum_{o \in Nbhd_D(s)} \alpha_{o,s} F_o(q) s.n \tag{2.3}$$

Here $Nbhd_D(s)$ denotes the neighbourhood with depth D that are closest to the sample s. The parameter $\alpha_{o,s}$ represents the trilinear interpolation weights. Interpolation across eight nearest neighbours allows for sub-node precision.

Then, the Poisson equation is solved in the function space defined above. Because the functions $\Delta\widetilde{\chi}$ and $\nabla \cdot \vec{V}$ are not necessarily in that function space, the problem is solved for $\widetilde{\chi}$ such that the projection of $\Delta\widetilde{\chi}$ to the function space is closest to the projection of $\nabla \cdot \vec{V}$:

$$\langle \Delta\chi, F_o \rangle = \left\langle \nabla \cdot \vec{V}, F_o \right\rangle, \quad \forall o \in O \tag{2.4}$$

The resulting linear system is sparse and can be solved using adaptive multigrid solvers. In case of non-uniform samples, the width of the smoothing kernel has to be adapted.

Chapter 3

Smooth Signed Distance Surface Reconstruction

A technique closely related to the popular Poisson Surface Reconstruction is introduced in [12]. This Smooth Signed Distance Surface Reconstruction is an implicit approach, produces a globally consistent solution and similarly to the case of the Poisson Reconstruction, the solution reduces to a sparse linear system of equations. Instead of an indicator function, the smooth approximation of the signed distance is considered and directly incorporated into the energy functional. There are many possible discretizations of the problem. In [12], the authors propose to use an efficient hybrid finite element-finite difference discretization on an octree structure.

3.1 Problem Statement

In [12] the same problem has been addressed as in the case of the Poisson Reconstruction [3]. The goal is to reconstruct a surface ∂M of a solid M,

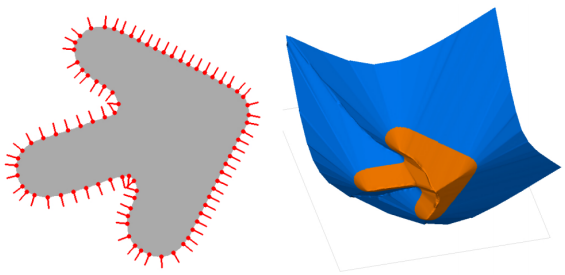

Figure 3.1: Indicator function and smooth signed distance function

which is defined by an implicit equation $Z = \{x : f(x) = 0\}$ from a finite set of oriented points $S = \{(p_1, n_1)...(p_N, n_N)\}$, where p_i denotes the location of

a surface sample and n_i expresses the corresponding surface normal which is oriented towards the outside of the object. Furthermore, it is assumed that outside of the object $f(x) > 0$ holds and inside we have $f(x) < 0$. One major difference is that the implicit function is forced to be a smooth approximation of the signed distance function instead of a discontinuous indicator function. This is shown in figure 3.1, where on the left oriented point set is visualized together with its indicator function (gray color represents value of 1 and white represents value of 0). On the right, the corresponding smooth signed distance function is visualized.

The gradient $\nabla f(x)$ of a function $f(x)$ is a vector field perpendicular to the level sets of the function. Intuitively, the gradient $\nabla f(x)$ of the function $f(x)$ should satisfy $\nabla f(p_i) = n_i$ and the point samples should satisfy $f(p_i) = 0$ for all points in the data set. The gradient is assumed to be of unit length at the surface, hence a direct comparison with the observed vector field is possible. Again, just as in [3], the functional $\nabla f = \vec{V}$ cannot be solved directly as its solution is not guaranteed. Therefore, the authors of [12] attempt to solve it in a least-square sense by minimizing an energy functional based on the interpolating conditions mentioned above. The details will be explained in the following section.

3.2 Energy Functional

In [12], the authors propose to handle the problem by minimizing the energy functional in a least-square sense. This energy functional consists of two main parts, the data term $E_D(f)$ and the regularization term $E_R(f)$:

$$E(f) = E_D(f) + E_R(f) \tag{3.1}$$

3.2.1 Data Term

In the data term the summation over all data points is considered. To optimally fit the data, the signed distance to the point samples p_i should be zero and the gradient of the signed distance function should match the normal samples n_i at the point samples p_i.

$$E_D(f) = \lambda_0(\frac{1}{N}\sum_{i=1}^{N} f(p_i)^2) + \lambda_1(\frac{1}{N}\sum_{i=1}^{N} \|\nabla f(p_i) - n_i\|^2) \tag{3.2}$$

The data term forces the implicit function to fit the data and the positive constants λ_0 and λ_1 influence the weight of each term. In order to be independent from the total number of points a normalization factor $\frac{1}{N}$ is used. The importance of the normalization will become clear in the next section, where the regularization term is considered.

3.2.2 Regularization Term

Including only a data term, the minimization problem might not be well-conditioned depending on the family of functions $f(x)$ considered. This is due to the fact that the energy functional does not specify how the function should behave away from the data points. For that reason, the additional regularization term

$$E_R(f) = \lambda_2(\frac{1}{|V|}\int_V \|Hf(x)\|^2\, dx) \tag{3.3}$$

is added, where the integral is over the volume V, i.e. the bounding box of data points, where the surface should be reconstructed. The term $Hf(x)$ denotes the Hessian matrix of f and the norm of the matrix is chosen to be the Frobenius matrix norm. The regularization term has two effects. One

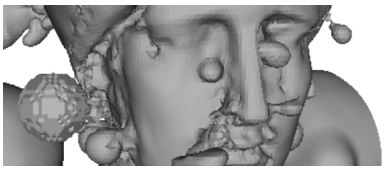

Figure 3.2: In case the implicit function is not constrained away from the data points, spurious surface sheets may occur

aspect of this term is the behaviour of $f(x)$ away from the data points, where this term will dominate over the data term and it will make the gradient nearly constant (to avoid the effects in Figure 3.2). The second effect is that with higher weights, it enforces a low curvature of the surface. That results in a smoother reconstruction of the object.

3.3 Implementation Motivation

Considering the aspect of existence and uniqueness of the solution for the problem defined by equation (3.1) the authors of [12] restrict their analysis to families linearly parametrized by a finite number of parameters. That yields in solving a linear system with a unique solution. Members of this families can be expressed as

$$f(x) = \sum_{\alpha\in\Lambda} f_\alpha \phi_\alpha(x) \tag{3.4}$$

where the index α belongs to a finite set Λ with K elements. The K basis functions $\phi_\alpha(x)$ are chosen in advance and f_α are the corresponding coefficients. The function can also be expressed as an inner product

$f(x) = \Phi(x)^T F$, where $F = (f\alpha : \alpha \in \Lambda)$ is a $K-$dimensional vector containing the coefficients and $\Phi(x) = (\phi_\alpha(x) : \alpha \in \Lambda)$ is a $K-$dimensional vector of basis functions.

In [12], the authors first consider to restrict the energy function to basis functions with continuous derivatives up to second order (e.g. radial basis functions). This results in a positive semi-definite, non-homogeneous, quadratic function $F^T AF - 2b^T F + c$, where A is usually a symmetric and positive definite matrix, except for very singular configurations. The resulting finite dimensional minimization problem has a unique and global minimum which can be found by solving the linear system $AF = b$. Hence, the solution can be simply obtained by first determining the contributions of each term to the matrix A and the vector b and then solving the system. The first term of the data term contributes only to matrix A, its contribution is:

$$A^0_{\alpha\beta} = \frac{1}{N} \sum_{i=1}^{N} \phi_\alpha(p_i)^T \phi_\beta(p_i) \tag{3.5}$$

As the second term can be expanded as $\|\nabla f(p_i) - n_i\|^2 = \|\nabla f(p_i)\|^2 - 2n_i^T \nabla f(p_i) + \|n_i\|^2$, its first part contributes to the matrix A according to

$$A^1_{\alpha\beta} = \frac{1}{N} \sum_{i=1}^{N} \nabla\phi_\alpha(p_i)^T \nabla\phi_\beta(p_i) \tag{3.6}$$

and its second part to the vector b in the form

$$b^1_\alpha = \frac{1}{N} \sum_{i=1}^{N} n_i^T \nabla\phi_\alpha(p_i). \tag{3.7}$$

The third part can be neglected. This is due to the fact, that normal vectors are assumed to have unit-length. As a consequence, the contribution to the constant is λ_1, but additive constants have no influence on the optimization. Finally, the contribution of the regularization term to the matrix A is:

$$A^2_{\alpha\beta} = \frac{1}{|V|} \int_V \langle H\phi_\alpha(x), H\phi_\beta(x) \rangle \, dx \tag{3.8}$$

where the inner product is associated with the Frobenius norm. This leads to

$$A = \lambda_0 A^0 + \lambda_1 A^1 + \lambda_2 A^2 \tag{3.9}$$

and

$$b = \lambda_1 b^1. \tag{3.10}$$

The accumulation of the contributions A^0, A^1 and b^1 requires the evaluation of the basis functions and their derivatives on all the data points. Having the case of fixed basis functions which are known in analytic form

allows to pre-compute the contribution of A^2 of the regularization term and to store it into a static table. However, the integrals in equation (3.8) have to be evaluated during the runtime of the program if the basis functions are constructed as functions of the data set. Although there is the possibility to evaluate these integrals in closed form for many of linear families, in [12] it is hinted that numerical approximations are usually acceptable.

Generally for the purpose of obtaining a fine-reconstruction, a large degree of freedom is needed, leading to a large number of parameters K. In case of non-decaying basis functions (like RBF) that presents a problem since the matrix A will be large and dense, resulting in less efficiency with regard to storage requirements and speed. This problem can be solved efficiently by choosing basis functions that are compactly supported, which means that the support of only a small number of basis functions overlaps at any point in the volume, and hence, the matrix A will be sparse (the functions used in [3, 4] would lead to this case). Having no intersection of the support of basis functions ϕ_α and ϕ_β leads to $A^2_{\alpha\beta} = 0$. To avoid additional complications in evaluating integrals associated with the regularization term, the authors of [12] propose even a different discretization described in the next chapters.

3.4 Hybrid FE/FED Discretization with Discontinuous Gradient

Considering basis functions with derivatives continuous up to first order and with second order derivatives integrable in the volume V, it is also possible to extend technique described in the previous sections. As long as $f(x)$, its gradient $\nabla f(x)$ and the Hessian $Hf(x)$ can be expressed as a linear combination of parameter vector F, the formulation still leads to a non-homogeneous quadratic equation $F^T AF - 2b^T F + c$ with a global minimum obtained by solving $AF = b$. The authors of [12] even note that $f(x)$, $\nabla f(x)$ and $Hf(x)$ can be discretized independently as long as they are homogeneous linear functions of same parameter F.

In [12], such a hybrid discretization is introduced by using a finite element discretization for $f(x)$ and finite difference discretization for $\nabla f(x)$ and $Hf(x)$. The advantage of such a discretization is that it is easy to implement, it can be applied to adaptive grids and allows to obtain accurate results fast.

First, the authors consider a bounding volume in the form of a unit cube $V = [0, 1] \times [0, 1] \times [0, 1]$ and divide it into hexahedral cells (finite elements). Each axis is split into M segments of equal length, which results in M^3 cells and $(M+1)^3$ vertices, which is shown in Figure 3.3. Grid vertices are noted with multi-indices $\alpha \in \Lambda, \alpha = (i, j, k)$ where $0 \leq i, j, k \leq M$. The cells C_α are also indexed with a multi-index α with $0 \leq i, j, k < M$. In addition, the values of the function $f(x)$ at the grid vertices $p_\alpha = (i/M, j/M, k/M)$ are denoted by f_α.

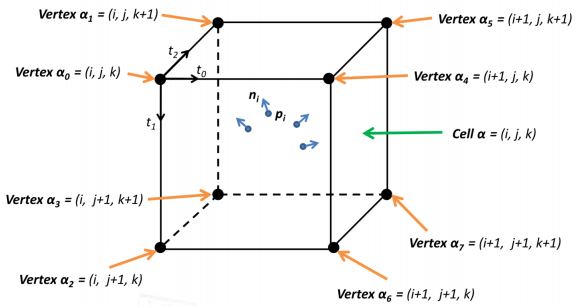

Figure 3.3: Vertices of hexahedral cell

Finite Element Discretization Within each cell C_α trilinear interpolation is used for discretizing $f(x)$:

$$f(p_i) \approx \sum_{h=0}^{7} w_h f_{\alpha_h} \tag{3.11}$$

The parameters w_h are the trilinear weights and the sum goes over eight cell vertices.

Finite Differences Discretization The discretized gradient $\nabla_\alpha f$ is computed within each cell using simple forward differences in each dimension:

$$\nabla_\alpha f = \frac{1}{4\Delta_\alpha} \begin{pmatrix} f_{\alpha_4} - f_{\alpha_0} + f_{\alpha_5} - f_{\alpha_1} + f_{\alpha_6} - f_{\alpha_2} + f_{\alpha_7} - f_{\alpha_3} \\ f_{\alpha_2} - f_{\alpha_0} + f_{\alpha_3} - f_{\alpha_1} + f_{\alpha_6} - f_{\alpha_4} + f_{\alpha_7} - f_{\alpha_5} \\ f_{\alpha_1} - f_{\alpha_0} + f_{\alpha_3} - f_{\alpha_2} + f_{\alpha_5} - f_{\alpha_4} + f_{\alpha_7} - f_{\alpha_6} \end{pmatrix} \tag{3.12}$$

Here $\Delta\alpha$ is the length of the cell's edge and it is assumed that cells are cubes. For different lengths of the cell dimensions the different lengths have to be used for the normalization of the coordinates of $\nabla_\alpha f$. For $p_i \in C_\alpha$, that yields $\nabla f(p_i) \approx \nabla_\alpha f$.

As the gradient is piecewise constant, the second derivative is equal to zero within each of the cells and, hence, the square norm of the Hessian is supported only on the faces shared by neighbouring voxels. Thus, the integral over the volume reduces to a finite sum over the faces:

$$E_R(x) \approx \frac{1}{|V|} \sum_{\alpha,\beta} |V|_{\alpha,\beta} \, \|H_{\alpha,\beta} f\|^2 \tag{3.13}$$

where $|V|_{\alpha,\beta}$ is the area of the face between two cells C_α and C_β, and the finite differences of discrete gradients are given by

$$H_{\alpha,\beta} f = \frac{1}{\Delta\alpha\beta} (\nabla_\alpha f - \nabla_\beta f) \tag{3.14}$$

and $\Delta_{\alpha\beta}$ is the Euclidian distance between the centres of the cells.

The concept described generalizes to the octree data structure without any changes, except that the term $\Delta_{\alpha\beta}$ is then the area of the common face between two octree nodes.

3.5 Solving the Linear System

To fill the matrix A and the vector b, the octree and its dual graph must be traversed. First, for each non-empty cell C_α the terms $f(p_i)$ and $\nabla f(p_i)$ are added to $A_{\alpha_i\alpha_j}$ and b_{α_i}, where α_i and α_j are indices of the eight vertices of a cell.

For the regularization term, the dual graph is traversed, so for each dual edge between the cells C_α and C_β values of $|V|_{\alpha,\beta} \|H_{\alpha,\beta} f\|^2$ are added to $A_{\alpha_i\beta_j}$, where α_i is a vertex of the cell C_α and β_j is a vertex of C_β.

For solving the system, the authors of [12] propose an iterative cascading multigrid approach. First, the system is solved at a coarse level. The result is then used to initialize the solution on the next level and solved again using conjugate gradient solver.

3.6 Isosurface Extraction

After computing the coefficients f_α, the polygonal approximation of the zero-set of $f(x)$ should be computed. The authors employ the method described in [13], a Dual Marching Cubes algorithm. Using this approach, the dual graph of the octree is computed first and then the standard Marching Cubes algorithm is performed over the dual graph. This approach requires the values of the implicit function at the dual graph, that is at the centroids of the cells, where the simple average of associated cell corners is taken.

3.7 Results

The method presented in [12] was compared to the Multi-Level Partition of Unity Implicits (MPU) [6], Poisson surface reconstruction (Poisson) [3] and Streaming Surface Reconstruction Using Wavelets (D4 Wavelets) [4]. From the results shown in Table 3.2, which is provided in [12], it is clear that the SSD Reconstruction method is comparable to the others in terms of speed and memory efficiency. In terms of the quality of the reconstruction, the SSD Reconstruction method mostly outperforms the others, especially on the challenging data sets, see Table 3.1, where the Hausdorff distance is computed between the 3D model of David's head and its reconstructions of sampled model.

Additionally, the authors [12] conducted four tests. The first one is carried out on the Michelangelo's David set ($\sim$ 1 million samples) where

Figure 3.4: David's data set and reconstructions (MPU, Poisson, D4 Wavelets, SSD)

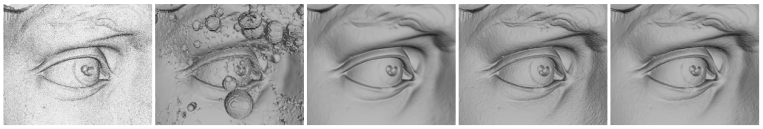

Figure 3.5: David's eye data set and reconstructions (MPU, Poisson, D4 Wavelets, SSD)

point samples are uniformly distributed and come together with accurate normals. All methods operated in this case at the same maximal octree depth of 8. The produced results (see Figure 3.4) are comparable in terms of quality while in terms of speed of reconstruction other methods are faster than the SSD.

For the second test, the eye fraction of the original David's data set was used (187,526 samples). In this case, the sampling is also uniform but the normals were estimated on the noisy triangles provided by the scanner and, hence, are inaccurate. For the reconstruction, octrees with maximal depth of 10 were used. The results are compared in Figure 3.5. While SSD, Poisson and the D4 Wavelets approach produces accurate results, the MPU method produce many spurious surface sheets.

For the third test, the authors of [12] used a data set that was obtained by sampling a virtual horse. The data set consists of 100,000 oriented point samples, the sampling density is proportional to the curvature of the model and the normal samples are accurate. Figure 3.6 compares the reconstructions obtained by the different methods. While the D4 Wavelets approach

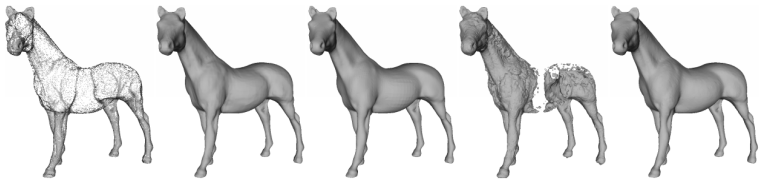

Figure 3.6: Virtual horse's data set and reconstructions (MPU, Poisson, D4 Wavelets, SSD)

Figure 3.7: Chiquita's data set and reconstructions (MPU, Poisson, D4 Wavelets, SSD)

Model	MPU	Poisson	D4	SSD
Armadillo	1.0000	0.4617	0.2383	0.3514
Dragon	0.8779	1.0000	0.5310	0.6810
Horse	0.0752	0.0827	1.0000	0.0551
Igea	1.0000	0.7761	0.5701	0.4018

Table 3.1: Hausdorff distance between real surfaces and their reconstructions, each row is normalized by the maximum error. Lower is better.

fails on this data set, all other approaches produce accurate results with minor differences.

For the final test the authors of [12] use the Chiquita dataset, captured by an inexpensive 3D structured lighting system. This dataset consists of 705,375 non-uniformly distributed oriented samples with inaccurate normals. For the reconstructions, octrees of maximal depth 8 were used. The results are compared in Figure 3.7. The MPU method again produces many spurious surface sheets and the D4 Wavelets approach is unable to fill the holes reasonably. Both the Poisson and the SSD approaches produce reasonable surfaces, but it can be noted that SSD approach extrapolates the surface better in the large areas that contain no samples.

Model	Time (sec)	Memory (MB)	Polygons
MPU	27	148	378925
Poisson	283	148	319989
D4	17	63	365974
SSD	72	264	346652

Table 3.2: Running time, peak memory usage and number of triangles of reconstructed mesh for David's head.

3.8 Conclusion

In the Smooth Signed Distance Reconstruction paper, a new variational method for surface reconstruction is introduced. In this approach, an implicit smooth signed distance function is fitted to the oriented point samples.

The formulation of the problem reduces to solving a sparse linear system with a global and unique solution. This new variational approach is built on top of ideas introduced in the Poisson Surface Reconstruction, but the energy optimization approach is simplified and it mostly produces better results in terms of surface reconstruction quality.

The efficient discretization of the given formulation is also proposed and it can be easily extended to adaptive data structures. The authors of the method also provide results and comparisons to other implicit function based methods, where SSD method performs especially well on challenging data sets.

Bibliography

[1] O. Schall and M. Samozino, "Surface from scattered points: A brief survey of recent developments," in *1st International Workshop on Semantic Virtual Environments* (B. Falcidieno and N. Magnenat-Thalmann, eds.), (Villars, Switzerland), pp. 138–147, MIRALab, 2005.

[2] T. Dey and Dey, *Curve and Surface Reconstruction: Algorithms with Mathematical Analysis*. Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, 2011.

[3] M. Kazhdan, M. Bolitho, and H. Hoppe, "Poisson surface reconstruction," in *Proceedings of the fourth Eurographics symposium on Geometry processing*, SGP '06, (Aire-la-Ville, Switzerland, Switzerland), pp. 61–70, Eurographics Association, 2006.

[4] J. Manson, G. Petrova, and S. Schaefer, "Streaming surface reconstruction using wavelets," *Computer Graphics Forum (Proceedings of the Symposium on Geometry Processing)*, vol. 27, no. 5, pp. 1411–1420, 2008.

[5] V. Lempitsky and Y. Boykov, "Global optimization for shape fitting," in *In CVPR, 2007. 6*, 2007.

[6] Y. Ohtake, A. Belyaev, M. Alexa, G. Turk, H.-P. Seidel, and M. SaarbrÃ$\tilde{}$$\frac{1}{4}$cken, "Multi-level partition of unity implicits," *ACM Transactions on Graphics*, vol. 22, pp. 463–470, 2003.

[7] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3d surface construction algorithm," *COMPUTER GRAPHICS*, vol. 21, no. 4, pp. 163–169, 1987.

[8] K. Zhou, M. Gong, X. Huang, and B. Guo, "Data-parallel octrees for surface reconstruction," *IEEE Trans. Vis. Comput. Graph.*, vol. 17, no. 5, pp. 669–681, 2011.

[9] M. Bolitho, M. Kazhdan, R. Burns, and H. Hoppe, "Multilevel streaming for out-of-core surface reconstruction," 2007.

[10] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Surface reconstruction from unorganized points," in *Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '92, (New York, NY, USA), pp. 71–78, ACM, 1992.

[11] M. Bolitho, M. Kazhdan, R. Burns, and H. Hoppe, "Parallel poisson surface reconstruction," in *Proceedings of the 5th International Symposium on Advances in Visual Computing: Part I*, ISVC '09, (Berlin, Heidelberg), pp. 678–689, Springer-Verlag, 2009.

[12] F. Calakli and G. Taubin, "Ssd: Smooth signed distance surface reconstruction.," *Comput. Graph. Forum*, vol. 30, no. 7, pp. 1993–2002, 2011.

[13] S. Schaefer and J. Warren, "Dual marching cubes: Primal contouring of dual grids," in *Proceedings of the Computer Graphics and Applications, 12th Pacific Conference*, PG '04, (Washington, DC, USA), pp. 70–76, IEEE Computer Society, 2004.