

Robust Surface Reconstruction

V. Estellers, M.A. Scott, S. Soatto.

October 26, 2015

Abstract

We propose a method to reconstruct surfaces from oriented point clouds by formulating the problem as a convex minimization that reconstructs the indicator function of the surface's interior. Our reconstruction is robust to noise and outliers because it substitutes the usual least-squares fidelity terms by robust Huber penalties that recover sharp corners and avoid the shrinking bias of least squares. We adopt an implicit parametrization to reconstruct surfaces of unknown topology and close gaps in the point cloud and design efficient discretizations that avoid the high memory and computational requirements of volumetric representations. To this purpose, we investigate three discretizations with a hierarchy of locally-supported functions adapted to the surface: two hierarchical B-spline bases of first and second degree polynomials, and a dictionary of quadratic hierarchical B-splines. The hierarchical structure of these discretizations speeds minimization through multiresolution and an efficient algorithm based on convex optimization that allows non-differentiable functionals. Our experiments show that our model improves reconstruction from both synthetic and corrupted data, while the choice of discretization affects both the accuracy of the reconstruction and its computational cost.

1 Introduction

New challenges to surface reconstruction from measurements emerge as datasets grow in size but lose in accuracy. The reduction in accuracy appears as sensors evolve from short to long range, low-cost commodity scanners become widely available, and computer vision is increasingly used to infer 3D geometry from point sets. As a result, surface reconstruction methods must be robust to noise and outliers, and scale favorably in terms of computation and memory use. This impacts the choice of parametrization for the surface and the inference technique that reconstructs a surface from an oriented point cloud.

Point positions suffer from non-uniform sampling, scan misregistration, and gaps that appear as accessibility constraints leave regions devoid of data. The normal vectors that describe the orientation of the surface, estimated from the point positions or from direct measurements, also suffer from noise and artifacts. Normals estimated from the points are unreliable for thick and noisy point clouds or surfaces with touching sheets, while normals measured by scanning devices – like photometric stereo – are corrupted by illumination artifacts. Noise in the oriented normals is especially detrimental because normals locally define the surface to first order and identify its topology. While correcting the normals in an oriented point cloud is relatively easy, correcting the topology of the

reconstructed surface is an expensive and delicate operation. For this reason, it is critical that surface reconstruction methods estimate the topology correctly.

We propose a robust but simple algorithm to reconstruct a water-tight surface from an oriented point cloud. We formulate the reconstruction as a convex optimization that recovers the indicator function of the interior of the surface. Our objective function penalizes deviations in the orientation and location of the surface with a Huber loss function to robustly recover the topology of the surface and allow for sharp corners; this makes our model more robust to noise and avoids the “shrinking bias” of least-squares models [23, 24]. This is our first contribution, explained in Section 3.

Our second contribution investigates three different discretizations to solve the minimization problem described in Section 4. In each case, the solution is parametrized by a linear combination of basis functions. The basis functions are all scaled and translated versions of a piecewise polynomial and create a hierarchical space that adapts the resolution of the representation to the resolution of the point cloud. The three discretizations investigate different polynomial orders and hierarchical constructions: a quadratic hierarchical B-spline, a linear hierarchical B-spline equivalent to the discretization of [10], and a dictionary of quadratic B-splines that have a more flexible hierarchy, do not form a basis, and are equivalent to the discretization of the Poisson reconstruction [23, 24]. Investigating the different discretizations used in surface reconstruction allows us to distinguish the improvements caused by the model from the ones caused by the discretization. Usually reconstruction methods are evaluated as a whole, making it difficult to determine whether improvements in accuracy or speed are due to the model, the algorithm, or the discretization.

Our third contribution is an efficient algorithm that exploits the convexity of the objective function to derive a first-order minimization algorithm that is easy to parallelize and scales well with the size of the point cloud. To this purpose, in Section 5 we adapt the primal-dual algorithm of [12] to our objective and derive closed-form solutions for each proximal update.

Compared to our conference paper [18], we have improved the model and have investigated two new discretizations. The new model does not require normal interpolation and is robust to noise and outliers in both the location and orientation of the points, while the new discretizations over octrees are lighter in terms of memory and computation.

2 Related Methods. Choice of Representation

Surface reconstruction methods can be classified as either parametric or implicit. Parametric techniques represent the surface as a topological embedding of a 2D parameter domain into 3D space. They include approaches based on computational geometry that partition space into Voronoi cells from the input samples and exploit the intuitive idea that eliminating facets of Delaunay tetrahedra provides a triangulated parametrization of the surface [3, 17, 4, 35, 9, 40, 25] and describes a combinatorial problem solved by local analysis of the cells [4, 35, 9], eigenvector decomposition [40], or graph cuts [25]. The reconstructed surface thus interpolates most of the input samples and requires post-processing to smooth the surface and correct the topology. Parametric methods generally require clean data as they assume the topology of the surface to be known, while

implicit methods are designed to reconstruct surfaces from noisy point clouds with unknown topology.

Implicit representations both reconstruct the surface and estimate its topology, but increase the dimension of the problem by representing the surface as the zero-level set of a volumetric function. Their accuracy is thus limited by the resolution of the grid, with efficient representations requiring non-uniform grids [23, 30, 10, 24].

Implicit representations can be formulated as either global or local. Local methods consider subsets of nearby points one at a time and handle large datasets efficiently. Earlier methods [22, 15] estimate tangent planes from the nearest neighbors of each sample and parametrize the surface by the signed distance to the tangent plane of the closest point in space. Moving least squares (MLS) techniques [2, 28, 39, 5] reconstruct surfaces locally by solving an optimization that finds a local reference plane and then fit a polynomial to the surface. The least-squares fit of MLS, however, is sensitive to outliers and smooths out small features; for this reason variants robust to outliers [34, 19] and sharp features [29, 16] appeared. [33] also constructs implicit functions locally but blends them together with partitions of unity. Common to these methods is their locality (i.e., partitioning into neighborhoods and merging local functions) that makes them highly scalable but sensitive to non-uniform sampling and point-cloud gaps.

Global methods define the implicit function as the sum of basis functions (RBFs [11], splines [23, 24], wavelets [30]) and consider all the data at once without heuristic partitioning. Kazhdan et al. [23, 24] solve a Poisson problem that aligns the gradient of the indicator function to the normals of the point cloud with a least-squares fit, not robust to outliers. Manson et al. [30] similarly approximate the indicator function with wavelets efficiently designed to compute basis coefficients with local sums over an octree. Calakli and Taubin [10] use a signed-distance function to represent the surface, but also rely on least squares to fit the normals and include a screening term that improves accuracy by fitting the input points to the zero-level set of the implicit function. For this reason, our model includes a screening term together with a robust Huber penalty to fit the normal field and allow for sharp edges. Existing methods account for sharp features by explicit representations [1, 21, 29] or anisotropic smoothing [14, 42, 13]; they are fast but depend on local operators that do not seek a global optimum.

Our reconstruction combines benefits of global and local schemes. It is global in the sense that it does not involve heuristics on neighborhoods, while it preserves locality by requiring the basis functions to be locally supported and adapt to the input point cloud. Our discretizations adapt to the input point cloud by means of an octree and are motivated by the octree representations of [10] and [23, 24], while the use of a hierarchical B-spline [44] is inspired by [8] and generalizes well-known uniform B-splines [43, 6, 41].

3 Variational Model

The reconstruction of a surface S from oriented points can be cast as a minimization problem that estimates the indicator function χ of the interior of the surface. Let $\{(x_k, \mathbf{n}_k)\}_{k=1}^{n_p}$ be the oriented point cloud, with $x_k \in \mathbb{R}^3$ the point

location and $\mathbf{n}_k \in \mathbb{S}^2$ its associated normal; our goal is to estimate a continuous function $\chi: \Omega \subset \mathbb{R}^3 \rightarrow \mathbb{R}$ such that $S = \{x: \chi(x) = 0\}$, χ takes negative values in the interior enclosed by S and χ positive outside.

The key to estimating the continuous function χ from the set of discrete point samples is to observe that each oriented point (x_k, \mathbf{n}_k) is a sample of the gradient of the indicator function, that is, $\nabla\chi(x_k) = \mathbf{n}_k$. As a result, we can reconstruct S by finding the scalar function whose gradient best matches these samples.

To account for noise in the data, we formulate the reconstruction as a minimization, instead of interpolation, problem:

$$\min_{\chi} \alpha \sum_{k=1}^{n_P} f(\chi(x_k)) + \beta \sum_{k=1}^{n_P} g(\mathbf{n}_k - \nabla\chi), \quad (1)$$

where α, β are positive model parameters, and f and g are Huber loss functions in \mathbb{R} and \mathbb{R}^3 with parameters ϵ_x, ϵ_n . In particular $f = h_{\epsilon_x}$, $g = h_{\epsilon_n}$, with

$$h_{\epsilon}(\mathbf{v}) = \begin{cases} \frac{1}{2\epsilon} |\mathbf{v}|_2^2 & |\mathbf{v}|_2 < \epsilon \\ |\mathbf{v}|_2 - \frac{\epsilon}{2} & |\mathbf{v}|_2 \geq \epsilon \end{cases}. \quad (2)$$

The Huber loss function h is a convex and differentiable penalty that avoids two artifacts of least-squares models [23, 10, 24] due to overpenalizing outliers: shrinkage of thin structures and smoothing of sharp edges. It overcomes these limitations by using different penalties for small errors and outliers, but results in a minimization harder to solve than the linear systems derived from least-squares fits.

The first term in (1) sets the points as soft interpolation constraints and fixes the surface parametrization to the zero-level set of χ ; it is a generalization of the screening term of [32], but defined over a sparse set of points rather than the integration domain Ω and with a robust Huber penalty instead of a least-squares fit. The second term penalizes errors in the sampled normals with a Huber loss function that make our model robust to noise in the normals; this is particularly important in reconstruction because errors in the normals cause errors in the surface topology that are difficult and expensive to correct once the surface is extracted.

The minimization problem (1) is undetermined for reconstruction because the model only constrains the gradient of χ close the sampled points, but lets χ vary freely away from them. From all the functions whose gradient approximates the samples, we are interested in those that lead to implicit representations of the surface and only evaluate to zero on it. We achieve this with a regularizer that ensures that χ does not evaluate to zero far from the samples and produce spurious surface sheets. In level-set parametrizations, this is usually accomplished by enforcing an approximate distance function with constant $|\nabla\chi|$ far from the surface or, equivalently, by penalizing changes in $|\nabla\chi|$ with a penalty on the Hessian $\mathcal{H}\chi$. The reconstruction model reads

$$\min_{\chi} \alpha \sum_{k=1}^{n_P} f(\chi(x_k)) + \beta \sum_{k=1}^{n_P} g(\mathbf{n}_k - \nabla\chi) + \frac{1}{2} \int_{\Omega} w |\mathcal{H}\chi|_F, \quad (3)$$

where $|\cdot|_F$ is the Frobenius norm, and w is a weighting function defined on the bounding box of the point cloud Ω . The first two terms in (3) are data terms

that should dominate the cost function close to the input samples and make χ an implicit parametrization of the surface, while the regularization term should dominate the objective far from the points to encourage an approximate signed distance parametrization. To achieve this balance, the weighting function w is a binary mask with 0 value in a neighborhood of samples and 1 elsewhere.

Our method is related to the Poisson reconstruction of [24] and the smooth signed distance of [10], but our model, representation of the surface, and minimization techniques are different. First, in terms of the model, we propose a robust Huber penalty on both the normals and the screening term to be resilient to outliers, instead of the least-squares penalty of [24] and [10]. Compared to [24] and our conference paper [18], we also avoid the interpolation of the sampled normals into a field by the inclusion of the Hessian regularizer; this requires the computation of additional second order derivatives but avoids pre-processing the normal samples, and in general leads to more accurate reconstruction of corners, where normal interpolation smears sharp changes in orientation. Compared to [10], our regularizer uses the Frobenius norm of the Hessian, instead of its square, to better approximate a signed distance function. Second, we adopt multiple discretizations to represent χ , two of them equivalent to the ones proposed by [10] and [24], and investigate the effects of discretization on the quality of the reconstruction. Finally, our minimization exploits the convexity of (1) to develop an efficient primal-dual algorithm, instead of finite-element methods that cannot handle the Huber loss functions.

4 Discretization

The discretization of (3) exploits the fact that its solution only needs to be accurate near the zero isolevel that parameterizes the surface, that is, in the neighborhood of the point samples. This calls for representations over irregular grids with high-resolution around the surface and coarser resolutions far from it. To this purpose, we discretize the space of functions over Ω with a finite dimensional vector space given by the span of basis functions ϕ_1, \dots, ϕ_n with higher spatial resolution near the point samples.

In the following, a function in Ω is discretized by the linear combination

$$\chi(x) = \sum_{\mathbf{A}} c_{\mathbf{A}} \phi_{\mathbf{A}}(x), \quad (4)$$

where \mathbf{A} is the finite index that sorts the basis functions, and $\mathbf{c} = (c_1, \dots, c_n)$ are the coefficients that represent χ and the variables in the optimization.

The accuracy of the representation is determined by the shape and resolution of the basis functions $\{\phi_{\mathbf{A}}\}$, while its computational cost is determined by the evaluation of the basis functions and their derivatives. These two criteria guide our choice of discretization: first, the basis functions must have derivatives up to second order that are integrable, either analytically or well approximated by quadrature rules; second, they must be compactly supported so that only a few functions are non-zero at each point and the linear combinations that define χ , $\nabla\chi$, $\mathcal{H}\chi$ can be evaluated fast. To accomplish this, we adopt a hierarchical B-spline (HBS) representation and investigate the impact of different polynomial degrees and smoothness with three discretizations:

1. a basis of quadratics HBS,

2. a lighter dictionary of quadratic HBS that do not form a basis,
3. a linear HBS basis that approximates second-order derivatives with finite differences.

4.1 Hierarchical B-splines

Hierarchical B-splines [20] are constructed through tensor products of splines over a hierarchy of grids of varying spatial resolution. Figure 1 shows a hierarchical B-spline in one dimension. B-spline functions are defined over the underlying knot partition at each level to define functions with varying spatial resolution and, to define a basis, linear dependencies are eliminated between levels. In Figure 1, the basis functions are denoted by solid lines, while those functions that are linearly dependent on higher levels are denoted by hatched lines. Hierarchical bases are a natural choice for our parametrization because of the compact representation, smoothness, and efficiency in performing local adaptivity due to the simple hierarchical structure.

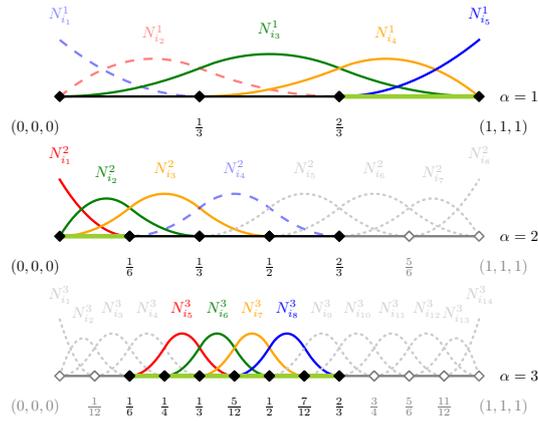


Figure 1: Basis functions for a 3-level quadratic HBS space. Basis functions are plot by solid colored lines, functions that are linearly dependent on higher-level functions by dashed colored lines, and inactive functions by grey dotted lines.

B-Spline Spaces

At each level of the hierarchy, a trivariate tensor-product B-spline space is defined by specifying the polynomial degree p and a knot vector that partitions the domain Ω into a regular hexahedral grid. For a unit volume $\Omega = [0, 1]^3$, the space is spanned by the tensor-product B-splines

$$N_{i,j,k}^p(x, y, z) = N_i^p(x) N_j^p(y) N_k^p(z) \quad 0 \leq i, j, k \leq m, \quad (5)$$

where the univariate B-spline functions are parameterized by knot vectors $[\xi_0, \xi_1, \dots, \xi_m]$ with $\xi_0 = 0, \xi_i \leq \xi_{i+1}, \xi_m = 1$ in each dimension and defined recursively as

$$N_i^0(x) = \begin{cases} 1 & \xi_i \leq x < \xi_{i+1} \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

$$N_i^p(x) = \frac{x - \xi_i}{\xi_{i+p} - \xi_i} N_i^{p-1}(x) + \frac{\xi_{i+p+1} - x}{\xi_{i+p+1} - \xi_{i+1}} N_i^{p-1}(x), \quad (7)$$

where fractions with zero denominators evaluate to zero. In our case, we have uniform knot vectors with $\xi_{i+1} - \xi_i = s$ for $0 \leq i < m$, and the B-spline space is parametrized by the degree p of the polynomial and the scale parameter s .

From this definition, each B-spline function $N_{i,j,k}^p$ is positive, differentiable p -derivatives at the knots, and has compact support

$$\text{supp } N_{i,j,k}^p = [\xi_i, \xi_{i+p+1}] \times [\xi_j, \xi_{j+p+1}] \times [\xi_k, \xi_{k+p+1}]. \quad (8)$$

The set $\mathcal{N}_s^p = \{N_{i,j,k}^p\}$ is linearly independent and forms a basis for the space of piecewise polynomials of degree p with uniform knot intervals of scale s .

Hierarchical B-Spline Spaces

A basis \mathcal{B}^p of a hierarchical B-spline space is constructed recursively from a set of nested volumetric patches $\Omega = \Omega^0 \supseteq \Omega^1 \supseteq \dots \supseteq \Omega^L$ of increasing spatial resolution, as follows:

1. Initialization: $\mathcal{B}_0^p = \mathcal{N}_1^p$.
2. Recursion: $\mathcal{B}_l^p = \{\phi \in \mathcal{B}_{l-1}^p : \text{supp } \phi \not\subseteq \Omega^l\} \cup \{\phi \in \mathcal{N}_{2^{-l}}^p : \text{supp } \phi \subseteq \Omega^l\}$.
3. At level L , $\mathcal{B}^p = \mathcal{B}_L^p$ defines the basis functions $\{\phi_A\}$ in our discretization.

We initialize $\mathcal{B}_0^p = \mathcal{N}_1^p$ and at each recursive step l we replace the basis functions whose support is entirely contained in the refined patch Ω^l by the refined basis functions in $\mathcal{N}_{2^{-l}}^p$ over this patch. With this construction, the basis functions of each \mathcal{B}^p are linearly independent and form a basis for $\text{span } \mathcal{B}^p$ with nested spaces $\text{span } \mathcal{B}_l^p \subseteq \text{span } \mathcal{B}_{l+1}^p$ [44].

4.2 Discretization 1: Quadratic HBS Basis

Our first discretization is a quadratic hierarchical B-spline space where the basis functions $\{\phi_A\} = \mathcal{B}^2$ define a basis. We choose $p = 2$ because it provides the twice-differentiable basis with minimal support and allows us to have analytical expressions for the Hessian. This is important because it allows us to use interpolation to evaluate χ , $\nabla \chi$ and $\mathcal{H} \chi$ in our objective functional.

In particular, the linearity of the discretization lets us efficiently compute the value of χ at any point x in the domain by evaluating the basis functions at this point. In other words,

$$\chi(x) = \sum_A c_A \phi_A(x) = [\phi_1(x) \dots \phi_n(x)] \mathbf{c}. \quad (9)$$

The procedure is efficient for two reasons: first, only a few basis functions are active at this point and thus the sum involves only a few terms and, second,

there are a number of efficient algorithms to evaluate the basis functions such as the Cox-De Boor algorithm[26]. Similarly, the gradient and Hessian $\nabla\chi$, $\mathcal{H}\chi$ are evaluated efficiently as

$$\nabla\chi(x) = \sum_A c_A \nabla\phi_A(x) = [\nabla\phi_1(x), \dots, \nabla\phi_n(x)] \mathbf{c} \quad (10)$$

$$\mathcal{H}\chi(x) = \sum_A c_A \mathcal{H}\phi_A(x) = [\mathcal{H}\phi_1(x), \dots, \mathcal{H}\phi_n(x)] \mathbf{c}. \quad (11)$$

Finally, we discretize the integral in our objective functional (1) as the weighted sum

$$\int_{\mathbb{R}^3} w(x) |\mathcal{H}\chi(x)|_F dx \approx \sum_{i=1}^{n_Q} w_{q_i} w(q_i) |\mathcal{H}\chi(q_i)|_F \quad (12)$$

with Gauss quadrature rules, where q_i is a quadrature point and w_{q_i} its associated quadrature weight. For efficiency, we use mid-point quadrature with a single quadrature point at the middle of each cell and weight given by its area.

4.3 Discretization 2: Linear HBS Basis

Our second discretization is a linear hierarchical B-spline space where the basis functions $\{\phi_A\} = \mathcal{B}^1$ define a basis. Choosing $p = 1$ reduces the support of the basis functions and speeds up evaluation of B-spline functions and their derivatives, but reduces the smoothness of the reconstruction.

The value of a B-spline function and its gradient are computed in the same way as for the quadratic HBS basis, but evaluating the Hessian requires a finite-difference approximation. Indeed, with a linear basis the gradient is constant within each cell and discontinuous at the cell boundaries, that is, the Hessian is a Dirac distribution supported on cell faces that is only integrable in the general sense. To compute the regularizer we substitute the integral over the volume by a sum over cell faces and compute the Hessian at the joint face of cells \mathcal{C}_A and \mathcal{C}_B with the finite difference of their gradients:

$$\mathcal{H}_{AB}\chi = \frac{1}{\Delta_{AB}} (\nabla_{A\chi} - \nabla_{B\chi}), \quad (13)$$

where Δ_{AB} is the Euclidean distance between the centers of the cells and $\nabla_{A\chi}$ is the constant value of $\nabla\chi$ over cell \mathcal{C}_A [10]. The integral in the objective functional simplifies into the sum

$$\int_{\Omega} |\mathcal{H}\chi|_F^2 = \sum_{(A,B)} w_{AB} |\mathcal{H}_{AB}\chi|_F^2, \quad (14)$$

where w_{AB} is the area of the face shared by cells \mathcal{C}_A and \mathcal{C}_B .

4.4 Discretization 3: Dictionary of quadratic HBS

Our third discretization is equivalent to the hierarchical spline space proposed in [23, 24] for surface reconstruction. This representation is designed to be as light as possible while providing smooth second derivatives and the same – but

not higher- spatial resolution than the input point cloud. To this purpose, the hierarchical structure of the space is defined by the octree of the input samples, and a quadratic B-spline basis is created at the center of each cell A with its scale set to match the width of the cell

$$\phi_A(x) = N^2 \left(\frac{x - x_A}{s_A} \right), \quad (15)$$

where ϕ is the unit trivariate quadratic B-spline basis with support $[-1.5, 1.5]^3$, x_A is the center of the cell and s_A its width. As a result, the basis functions are again translated and scaled versions of a quadratic uniform B-spline and hierarchically organized by their scale but, unlike in a quadratic HBS space, each basis function is defined independently of the others with no guarantees on their linear independence.

Dropping the linear-independence condition allows to replace basis functions at coarser levels more aggressively because there is no constraint on the size of the patch to be refined. Indeed, in the hierarchical construction of Section 4.1, only basis functions whose support is completely contained in the finer patch can be refined to ensure linear independence over this patch[20]. As a result, the size of the refined patch must exceed the support of the basis functions or, equivalently, any region needing refinement that is smaller than the support of the active basis functions must be expanded to ensure linear independence. This condition is more restrictive with quadratic than linear splines because their support is larger, motivating the definition of our quadratic B-spline dictionary. In particular, with the hierarchical structure of the octree and the quadratic spline (15), basis functions centered at a finer patch are refined even when their support exceeds the patch. This allows us to refine smaller patches than the quadratic HBS basis and reduce the size of the hierarchical representation at the cost of representation power and the linear-independence property.

The order of the spline basis is again quadratic because it has the minimum support for a twice differentiable basis with non-zero Hessian. As a result, the integral in the objective functional is again discretized with a weighted sum with the mid-point Gauss quadrature rule (12). A minimum support is important to ensure that a small number of basis overlap at each point and a linear combination of them only involves summing over a small subset.

This discretization provides higher-resolution discretizations at higher hierarchical levels, but it does not provide a basis as the other discretizations. This limits a multiresolution representation because it is only possible to approximately project the solution at depth l into the solution at depth $l + 1$, but it does not affect the convergence¹ of our optimization algorithm because we only need an approximate multiresolution representation to initialize the iterative algorithm at each scale (see Section 6).

4.5 Spatial Refinement with Octrees

In our HBS discretizations, it is necessary to define the set of nested volumetric patches $\Omega = \Omega^0 \supseteq \Omega^1 \supseteq \dots \supseteq \Omega^L$ that determine the spatial resolution of the

¹In the least-squares models of [23, 24], a basis representation is required to ensure that the system of equations can be solved. This is not a restriction with our formulation because our solution is guaranteed by convexity and our algorithm does not solve a linear system requiring linear independence to have a unique solution.

discretization. We do so by scaling² the bounding volume to a unit cube Ω and constructing the minimal octree that ensures that every point sample falls into a leaf node at depth $L = 7$. The hierarchical structure and spatial refinement of the octree is then translated into the three discretizations of Section 4.2-4.2 by simply mapping the tree structure of its cells into a corresponding nesting and refinement of patches: $\Omega^l \supset \Omega^{l+1}$ is defined by setting Ω^l to the coarser cell of the parent node and Ω^{l+1} the set of refined cells of its children.

This partitioning naturally leads to a finer resolution near the surface as described by the point cloud and provides a mesh where locating point samples and finding neighboring cells reduces to checking Morton codes [31]. With this refinement, our second discretization is equivalent to the mixed finite-element and finite-differences over an octree proposed by [10] for surface reconstruction, while the third discretization corresponds to the quadratic splines used in the Poisson reconstruction of [23, 24].

5 Minimization Algorithm

In all our discretizations, the integral in (1) becomes the weighted sum

$$\int_{\mathbb{R}^3} w(x) |\mathcal{H}\chi(x)|_F dx = \sum_{i=1}^{n_Q} w_i |\mathcal{H}\chi(q_i)|_F. \quad (16)$$

With quadratic basis functions, this sum is the result of applying standard quadrature rules where $w_i = w_{q_i} w(q_i)$ is the product of the quadrature weight w_{q_i} and the value of the weighting function w at quadrature point q_i . In the case of the linear HBS, the sum results from the definition of the Hessian as a Dirac distribution supported on the grid faces, and q_i, w_{q_i} are the position and area of the shared grid faces.

By restricting χ to the span of the basis $\{\phi_A\}_{A=1}^n$, we confine the minimization to coefficients $c \in \mathbb{R}^n$.

$$\min_c \alpha \sum_{k=1}^{n_P} g(\chi(x_k)) + \beta \sum_{k=1}^{n_P} f(\nabla\chi(x_k) - \mathbf{n}_k) + \frac{1}{2} \sum_{i=1}^{n_Q} w_i |\mathcal{H}\chi(q_i)|_F \quad (17)$$

$$\text{s.t. } \chi(x_k) = \sum_A c_A \phi(x_k) \quad 1 \leq k \leq n_P \quad (18)$$

$$\nabla\chi(x_k) = \sum_A c_A \nabla\phi(x_k) \quad 1 \leq k \leq n_P \quad (19)$$

$$\mathcal{H}\chi(q_i) = \sum_A c_A \mathcal{H}\phi(q_i) \quad 1 \leq i \leq n_Q. \quad (20)$$

The constraints (18) can be written more compactly by defining the row vector $P_k = (\phi_1(x_k) \dots \phi_n(x_k)) \in \mathbb{R}^{1 \times n}$ for each sample x_k and stacking them into a sparse matrix $P \in \mathbb{R}^{n_P \times n}$ with $P_{ij} = \phi_j(x_i)$, that is,

$$\chi(x_k) = P_k \mathbf{c} \quad (21)$$

²If the scaling is anisotropic, then the normal vector coordinates need to be scaled as well and then re-normalized to unit length.

Similarly, we can write the linear constraints on the gradient and Hessian (19)-(20) as multiplications with sparse matrices N and Q by stacking the components of each $\nabla\chi(x_k)$ and $\mathcal{H}\chi(q_i)$ into 3 and m -dimensional³ column vectors, that is,

$$\begin{aligned} \nabla\chi(x_k) &= N_k \mathbf{c} \in \mathbb{R}^3 & \text{with} & & N_k &= [\nabla\phi_1(x_k), \dots, \nabla\phi_n(x_k)] \in \mathbb{R}^{3 \times n} \\ \mathcal{H}\chi(q_i) &= Q_i \mathbf{c} \in \mathbb{R}^9 & \text{with} & & Q_i &= [\mathcal{H}\phi_1(q_i), \dots, \mathcal{H}\phi_n(q_i)] \in \mathbb{R}^{m \times n} \end{aligned}$$

where the sparse matrices $N \in \mathbb{R}^{3n_P \times n}$ and $Q \in \mathbb{R}^{mn_Q \times n}$ are constructed by stacking the block components $N_{kj} = \nabla\phi_j(x_k) \in \mathbb{R}^3$ and $Q_{ij} = \mathcal{H}\phi_j(q_i) \in \mathbb{R}^m$ for each sample and quadrature point. The minimization problem (17) then simplifies into the constrained minimization

$$\begin{aligned} \min_{c, u, V, W} \quad & \underbrace{\sum_{k=1}^{n_P} \alpha g(u_k)}_{G(u)} + \underbrace{\sum_{k=1}^{n_P} \beta f(V_k - \mathbf{n}(x_k))}_{F(V)} + \underbrace{\frac{1}{2} \sum_{i=1}^{n_Q} w_i |W_i|_F}_{H(W)} \quad \text{s.t.} \quad \begin{cases} u = Pc \\ V = Nc \\ W = Qc \end{cases} \\ \min_{c, X=[u, V, W]} \quad & G(u) + F(V) + H(W) \quad \text{s.t.} \quad X = Kc \quad \text{with matrix} \quad K = \begin{bmatrix} P \\ N \\ Q \end{bmatrix}. \end{aligned} \quad (22)$$

The problem now has the standard form of many convex minimization problems that are solved with splitting techniques. Among them, we adopt a primal-dual formulation of [12] by rewriting (22) as the saddle-point associated with its Lagrangian

$$\max_{\nu, \lambda, \mu} \min_c -F^*(\lambda) - G^*(\nu) - H^*(W) + \langle \nu, Pc \rangle + \langle \lambda, Nc \rangle + \langle \mu, Qc \rangle. \quad (23)$$

with the dual variable $Y = [\nu, \lambda, \mu]$ associated with the constraint $Y = Kc$, and F^* , G^* and H^* the convex conjugates of F , G and H .

We can solve (23) efficiently with the primal-dual algorithm [12]. We choose a first-order method because the size of the problem makes second-order methods unfeasible. The convexity of each term in the objective allows us to solve the minimization as a sequence of proximal problems and updates:

$$\nu^{n+1} \leftarrow \min_{\nu} \sigma G^*(\nu) + \frac{1}{2} \|\nu - \nu^{n+1} - \sigma P \bar{c}^n\|^2 \quad (24)$$

$$\lambda^{n+1} \leftarrow \min_{\lambda} \sigma F^*(\lambda) + \frac{1}{2} \|\lambda - \lambda^n - \sigma N \bar{c}^n\|^2 \quad (25)$$

$$\mu^{n+1} \leftarrow \min_{\mu} \sigma H^*(\mu) + \frac{1}{2} \|\mu - \mu^n - \sigma Q \bar{c}^n\|^2$$

$$\begin{aligned} c^{n+1} &= c^n - \tau (P^* \nu^{n+1} + N^* \lambda^{n+1} + Q^* \mu^{n+1}) \\ \bar{c}^n &= c^n + \theta (c^n - c^{n-1}) \end{aligned} \quad (26)$$

where P^* , N^* , Q^* are the adjoints of P , N , Q and α_n, τ, σ are algorithm parameters.

The efficiency of the proposed algorithm comes from the spatial separability of F^* , G^* , H^* and from the ability to find closed-form solutions for minimization problems (24) - (26). The derivation of closed-form solutions is detailed next, and Algorithm 1 summarizes the resulting updates, which are easy to parallelize.

³ $m = 9$ for the spline discretizations, while $m = 3$ for the linear HBS discretization.

<p>Initialize variables to zero. Set $\tau, \sigma, \theta > 0$ according to [12].</p> <p>while $\ c^{n+1} - c^n\ > 1^{-4}$ do</p>	
$\nu_k^{n+1} = \alpha \min(1, \frac{ \hat{\nu}_k }{\alpha + \epsilon_x \sigma}) \text{sign}(\hat{\nu}_k)$	$\hat{\nu} = \zeta_\nu^n + \sigma P \bar{c}^n$
$\lambda_k^{n+1} = \beta \frac{\bar{\lambda}_k}{\max(\beta + \sigma \epsilon_n, \bar{\lambda}_k _2)}$	$\bar{\lambda} = \zeta_\lambda^n + \sigma(N \bar{c}^n - \mathbf{n}_k)$
$\mu_i^{n+1} = \frac{\gamma w_i}{\max(\gamma w_i, \hat{\mu} _2)} \hat{\mu}$	$\hat{\mu} = \zeta_\mu^n + \sigma Q \bar{c}^n$
$c^{n+1} = c^n - \tau(P^* \nu^{n+1} + N^* \lambda^{n+1} + Q^* \mu^{n+1})$	
$\bar{c}^n = c^n + \theta(c^n - c^{n-1})$	
<p>end</p>	

Algorithm 1: Primal-dual minimization algorithm.

5.0.1 Minimization in ν

Let $\hat{\nu} = \zeta_\nu^n - \sigma Q \bar{c}^n$, we solve the minimization in ν (24) through Moreau's identity [36]:

$$\nu \leftarrow \min_{\nu} \sigma G^*(\nu) + \frac{1}{2} \|\nu - \hat{\nu}\|^2 \iff \nu = \hat{\nu} - \sigma u^*, \quad u^* \leftarrow \min_u G(u) + \frac{\sigma}{2} \|u - \frac{\hat{\nu}}{\sigma}\|^2.$$

The minimization in u is decoupled in each of its components $u_i \in \mathbb{R}$ with a single term in the sum

$$\min_u \sum_{k=1}^{n_P} \alpha g(u_k) + 0.5 \sigma (u_k - \sigma^{-1} \hat{\nu}_k)^2. \quad (27)$$

The minimization is thus solved by independently minimizing each term in the corresponding u_k . Due to the convexity and differentiability of the Huber loss g , the optimality conditions are obtained by differentiating the objective function with respect to u_k .

If $|u_k^*|_1 \leq \epsilon_x$, the optimality conditions are

$$\frac{\alpha}{\epsilon_x} u_k^* + \sigma u_k^* = \hat{\nu}_k \iff u_k^* = \frac{\epsilon_x}{\alpha + \epsilon_x \sigma} \hat{\nu}_k, \quad (28)$$

and the condition $|u_k^*|_1 \leq \epsilon_x$ becomes $|\hat{\nu}_k|_1 \leq \alpha + \epsilon_x \sigma$.

If $|u_k^*|_1 > \epsilon_x$, the optimality conditions

$$\alpha \text{sign}(u_k^*) + \sigma u_k^* = \hat{\nu}_k \quad \text{whith} \quad |u_k^*|_1 > \epsilon_x \quad (29)$$

are solved by

$$u_k^* = \begin{cases} \sigma^{-1}(\hat{\nu}_k - \alpha) & \hat{\nu}_k > \alpha + \sigma \epsilon_x \\ \sigma^{-1}(\hat{\nu}_k + \alpha) & \hat{\nu}_k < -\alpha - \sigma \epsilon_x \end{cases}. \quad (30)$$

Combining these two cases, the dual variable $\nu_k^{n+1} = \hat{\nu}_k - \sigma u_k^*$ is updated by

$$\nu_k = \alpha \min\left(1, \frac{|\hat{\nu}_k|_1}{\alpha + \epsilon_x \sigma}\right) \text{sign}(\hat{\nu}_k) = \begin{cases} \alpha & \hat{\nu}_k > \alpha + \sigma \epsilon_x \\ \frac{\alpha}{\alpha + \sigma \epsilon_x} \hat{\nu}_k & |\hat{\nu}_k|_1 \leq \alpha + \sigma \epsilon_x \\ -\alpha & \hat{\nu}_k < -\alpha - \sigma \epsilon_x \end{cases}. \quad (31)$$

5.0.2 Minimization in λ :

Let $\hat{\lambda} = \zeta_\lambda^n - \sigma N \bar{c}^n$, we solve the minimization in λ (25) again through Moreau's identity [36] and set $\lambda = \hat{\lambda} - \sigma V^*$ with

$$V^* \leftarrow \min_V F(V) + \frac{\sigma}{2} \|V - \frac{\hat{\lambda}}{\sigma}\|^2.$$

The minimization in V is decoupled in each one of its block components $V_i \in \mathbb{R}^3$ with a different term in the sum

$$\min_V \sum_{k=1}^{n_p} \beta f(V_k - \mathbf{n}_k) + 0.5\sigma(V_k - \sigma^{-1}\hat{\lambda}_k)^2. \quad (32)$$

The minimization is thus solved by independently minimizing each term in the sum. As the Huber loss function is differentiable and convex, the minimizers are the zeros of the derivative of the objective function with respect to each V_k , that is,

$$\beta \frac{V_k^* - \mathbf{n}_k}{\max(\epsilon_n, |V_k^* - \mathbf{n}_k|_2)} + \sigma V_k^* - \hat{\lambda}_k = 0. \quad (33)$$

If $|V_k^* - \mathbf{n}_k|_2 \leq \epsilon_n$, the optimality conditions are solved as follows

$$\frac{\beta}{\epsilon_n}(V_k^* - \mathbf{n}_k) + \sigma V_k^* = \hat{\lambda}_k \iff V_k^* = \mathbf{n}_k + \frac{\epsilon_n}{\beta + \epsilon_n \sigma}(\hat{\lambda}_k - \sigma \mathbf{n}_k), \quad (34)$$

and the condition $|V_k^* - \mathbf{n}_k|_2 \leq \epsilon_n$ becomes $|\hat{\lambda}_k - \sigma \mathbf{n}_k|_2 \leq \beta + \epsilon_n \sigma$.

If $|V_k^* - \mathbf{n}_k|_2 > \epsilon_n$, the optimality conditions can be re-written as

$$\left[\frac{\beta}{|V_k^* - \mathbf{n}_k|_2} + \sigma \right] (V_k^* - \mathbf{n}_k) = \hat{\lambda}_k - \sigma \mathbf{n}_k, \quad (35)$$

and vectors $V_k^* - \mathbf{n}_k$ and $\hat{\lambda}_k - \sigma \mathbf{n}_k$ are co-linear. The problem is then reduced to finding the scalar ρ such that $V_k^* - \mathbf{n}_k = \rho(\hat{\lambda}_k - \sigma \mathbf{n}_k)$, that is

$$\frac{\beta}{|\hat{\lambda}_k - \sigma \mathbf{n}_k|_2} + \sigma \rho = 1 \iff \rho = \frac{1}{\sigma} \left[1 - \frac{\beta}{|\hat{\lambda}_k - \sigma \mathbf{n}_k|_2} \right]. \quad (36)$$

As a result $V_k^* = \mathbf{n}_k + \frac{1}{\sigma} \left[1 - \frac{\beta}{|\hat{\lambda}_k - \sigma \mathbf{n}_k|_2} \right] (\hat{\lambda}_k - \sigma \mathbf{n}_k)$ and the condition $|V_k^* - \mathbf{n}_k|_2 > \epsilon_n$ becomes $|\hat{\lambda}_k - \sigma \mathbf{n}_k|_2 > \beta + \epsilon_n \sigma$. Combining these two cases, we obtain the following closed-form update for the dual variable $\lambda_k^{n+1} = \hat{\lambda}_k - \sigma V_k^*$

$$\lambda_k = \beta \frac{\hat{\lambda}_k - \sigma \mathbf{n}_k}{\max(\beta + \epsilon_n \sigma, |\hat{\lambda}_k - \sigma \mathbf{n}_k|_2)}. \quad (37)$$

5.0.3 Minimization in μ :

Let $\hat{\mu} = \zeta_\mu^n - \sigma Q \bar{c}^n$, we solve the minimization in μ (26) again through Moreau's identity [36] and set $\mu = \hat{\mu} - \sigma W^*$ with

$$W^* \leftarrow \min_W H(W) + \frac{\sigma}{2} \|W - \frac{\hat{\mu}}{\sigma}\|^2.$$

The minimization in W is decoupled in each one of its block components $W_i \in \mathbb{R}^m$ with a different term in the sum

$$\min_W \sum_{i=1}^{n_Q} \gamma w_i |W_i|_F + 0.5\sigma(W_i - \sigma^{-1}\hat{\mu}_i)^2. \quad (38)$$

The minimization is thus solved by independently minimizing each term in the sum with respect to its W_i ; this correspond to the proximal operator of the Frobenius norm that results in the shrinkage operator

$$W_i^* = \frac{1}{\gamma w_i + \sigma} \hat{\mu}_i. \quad (39)$$

The dual variable $\mu_k^{n+1} = \hat{\mu}_i - \sigma W_i^*$ is updated by

$$\mu_i = \frac{\gamma w_i}{\max(\gamma w_i, |\hat{\mu}_i|_F)} \hat{\mu}_i. \quad (40)$$

6 Multiresolution

Given the hierarchy of the proposed discretizations, it is natural to accelerate the minimization with multiresolution. To this end, the basis functions are partitioned according to their hierarchical level $\mathcal{B}^p = \mathcal{B}_L^p \cup \mathcal{B}_{L-1}^p \cup \dots \cup \mathcal{B}_1^p$ and the optimization problem (17) is solved with increasing resolution as follows:

1. Initialize $\chi_0^0 = 0$.
2. Repeat
 - (a) Optimization at level l : find $\chi^l \in \text{span } \mathcal{B}_l^p$ that solves (17) with Algorithm 1 initialized at χ_0^l .
 - (b) Compute χ_0^{l+1} by projecting χ^l into $\text{span } \mathcal{B}_{l+1}^p$ with B-spline refinement [20, 23].
3. At level L , $\chi = \chi^L$.

At each level, the optimization converges with only a few iterations because the algorithm is initialized close to the optimum with the solution at a coarser resolution. This speeds up the optimization without changing the final solution because the convexity of the problem guarantees a unique minimum.

Multiresolution reduces the size of the discretization, but the size of the data terms in the minimization problem (3) remains fixed because the data terms

$$\alpha \sum_{k=1}^{n_P} f(\chi(x_k)) + \beta \sum_{k=1}^{n_P} g(\mathbf{n}_k - \nabla \chi) \quad (41)$$

loop over all the input samples. We can introduce multiresolution in the model and accelerate reconstruction by defining approximate data terms that cluster the samples to the resolution of the representation. To this purpose, we exploit the octree structure and substitute all the point samples inside an octree cell \mathcal{C} by their centroid $x_{\mathcal{C}}$ and weight their contribution by the number of samples in the cell $w_{\mathcal{C}}$. At intermediate resolutions we then substitute

$$\sum_{k=1}^{n_P} \alpha f(\chi(x_k)) + \beta g(\mathbf{n}_k - \nabla\chi) \approx \sum_{\mathcal{C}} w_{\mathcal{C}} [\alpha f(\chi(x_{\mathcal{C}})) + \beta g(\mathbf{n}_{\mathcal{C}} - \nabla\chi(x_{\mathcal{C}}))], \quad (42)$$

where $\mathbf{n}_{\mathcal{C}}$ is the average normal of the samples in cell \mathcal{C} . The complexity of the resulting minimization matches the complexity of the computational grid and improves speed at intermediate resolutions without compromising the accuracy of the reconstruction at the finest resolution.

7 Experimental Results

We perform experiments with two kinds of data: synthetic data with ground truth, and point clouds obtained from structured-light scanning with a Kinect camera [7] or a range sensor. Kinect and range data suffers from non-uniform noise, large scanning gaps, and artifacts. We use the synthetic point clouds for quantitative evaluation and the noisy ones to test reconstruction on data with real noise and artifacts⁴.

The reconstructed surfaces are obtained by extracting the zero isolevel of the implicit function with the Dual-Marching-Cubes algorithm of [38]. This algorithm discretizes the surface with a polygonal mesh and can be applied to any volumetric grid of hexahedral cells like the ones of our discretizations.

The model parameters α, β are normalized by the number of points and γ is normalized by the size of the bounding volume to make the model independent of size and range of the point cloud. After this normalization, they are manually set to values as they provide good reconstructions for all the experiments. The parameters ϵ_x, ϵ_n are fixed estimates of the noise in the location and orientation of point cloud that we set manually to 0 for clean data and 0.1 for noisy ones. Spatially varying parameters are possible, but we do not investigate them here.

Our first experiment compares the proposed model to a least-square fit in Figure 2 with synthetic, noisy and corrupted data with the same discretization (linear HBS basis) to focus on our first contribution, the use of robust penalties for both the normal field and the screening term. In Figures 2(a) - 2(d) the least-squares model rounds the cube’s corners because it blindly averages the normal samples from different cube sides on the corners, our robust model overcomes this limitation with a Huber penalty that considers the normal samples from different cube sides as outliers for the others. In Figures 2(e) - 2(h), a least-squares penalty on the screening term shrinks thin structures, like the ears or horns of the cow, because point samples from opposite sides act as outliers and pull the reconstruction to the center of the thins structure, leading to shrinkage.

⁴ All the datasets are publicly available at <http://graphics.stanford.edu/data/3Dscanrep/>, http://www.cc.gatech.edu/projects/large_models/, <http://people.sc.fsu.edu/~jburkardt/data/>, and <http://bitbucket.org/jbalzer/yas>.

Table 1: Average reconstruction time and Hausdorff distance distance, as a percentage of the bounding-box diagonal, between the point cloud and the reconstructed surface from 10% of its samples.

discretizations	average distance				running time (seconds)			
	dragon	bunny	cow	cube	dragon	bunny	cow	cube
quadratic HBS basis	0.387	0.538	0.402	1.125	672	403	432	326
HBS dictionary	0.105	0.111	0.088	0.808	81	75	46	24
linear HBS basis	0.098	0.143	0.068	0.157	58	53	45	5

Finally, our reconstruction is robust to outliers caused by scanning artifacts in the scanned head of Figures 2(i) - 2(k) without shrinking the ear by over-smoothing like the least-squares model. From this experiment, we conclude that the robust data-fidelity term improves reconstruction, both for the screening and normal components.

Our second experiment investigates the effects the discretization in the accuracy of the reconstruction. Results are shown in Figure 3 and Table 1. The richer but also more complex discretization is the first one (quadratic HBS basis), while the other two correspond to two possible simplifications in terms of polynomial degree (linear HBS basis) and the refinement strategy (quadratic HBS dictionary). In particular, the quadratic dictionary and the linear HBS defined by the octrees create the same hierarchical structure and have comparable complexity, but differ slightly in speed and accuracy. Quadratic splines are slightly slower because they require computing second-order derivatives with a Cox-DeBoor algorithm, while the linear HBS basis only needs to evaluate finite-differences of vertex values. In terms of accuracy, the linear HBS has constant derivatives within each cell that lead to smoother reconstructions when combined with the hierarchical basis, while the quadratic HBS dictionary produce very small and localized basis functions in textured areas that can cause wiggles, see the cow’s torso in Figures 3(m) and 3(o). The quadratic HBS basis produces smoother results but is an order of magnitude slower than the simplified discretizations, moreover the posterior use of marching cubes to extract the surface can lead to holes or cracks, as observed in the right ear of the cow or the lower left wing of the angel in Figures 3(n) and 3(h). Both the cracks in the quadratic HBS dictionary and the wiggles in the quadratic HBS dictionary can be eliminated by increasing the order of the quadrature rule at the price of a higher computational cost – in memory and computations. For this reason, we adopt the linear HBS discretization over the octree in the rest of our experiments.

A third set of experiments compares our model and discretizations to the state-of-the-art techniques [23, 24, 10]⁵. Figure 4 compares them visually for synthetic data, for which we have ground truth, and Table 2 quantitatively. Although the least-squares models [24, 10] have average Hausdorff distances comparable to our model, see Table 2, they produce errors that are not captured by the statistics of the point cloud but the geometry of the reconstruction. The models of [24, 10] round the cube’s corners and shrink the ears of the cow

⁵As [24] is a generalization of [23] and we can switch between the two models by adapting a parameter, we tune this parameter for the best reconstruction performance with each point cloud and only report results for [24] models.

Table 2: Average reconstruction time and Hausdorff distance $d_{\mathcal{H}}$, as a percentage of the bounding-box diagonal, between the point cloud and the clean reconstructed surface. Average running time

$d_{\mathcal{H}}$	10% subsampled point cloud				point cloud with noise of $\sigma = 0.1$				average time (s)
	dragon	bunny	cow	cube	dragon	bunny	cow	cube	
[10]	0.112	0.116	0.069	0.192	0.150	0.290	0.198	0.288	8
[24]	0.096	0.095	0.043	0.193	0.110	0.170	0.187	0.221	88
ours	0.098	0.143	0.068	0.158	0.112	0.172	0.171	0.225	51

because they rely on least-squares data terms, while our model overcomes these artifacts with a robust Huber penalty that recovers the sharp corner in the cube and avoids the shrinking bias. This bias does not only reduce the size of small structures, but it can lead to geometrically incorrect reconstructions, as shown in Figures 4(j)-4(l) where the least-squares models of [24, 10] merge the ears and horns of the cow into a single structure that misrepresents the high-curvature geometry of these areas. These experiments allow us to quantitatively evaluate the performance of our model in Table 2 by measuring the Hausdorff distance between clean point clouds and the surfaces reconstructed from perturbed point clouds, obtained by either sub-sampling the point cloud or adding white Gaussian noise of standard deviation σ to the point locations and normals.

As we are interested in reconstruction from corrupted point clouds suffering from outliers, we also perform experiments with noisy point clouds captured with a Kinect camera [7] and range data. In this case, the least-squares models [24] and [10] reconstruct surfaces that reproduce the scanning artifacts of the point clouds and lead to incorrect topology estimates, as shown in Figures 5-6. The two model contributions of our reconstruction are featured in these experiments. First, the use of a robust Huber function allows us to estimate the correct topology – the shoelace of Figure 6(k) or the human ear in Figure 5(l) – and eliminate spurious point clusters – the head of the cleaner in Figure 5(h) or the dancer’s stature of Figure 6(g). Estimating the correct topology is an important feature because it is expensive to recover the correct topology once an explicit parametrization, e.g., a triangulated surface, has been extracted. Second, our regularizer allows for sharp changes in orientation and avoids over-smoothing the normals in textured areas, as shown in Figures 5(c)-5(d) and 6(j)-6(k). In our regularizer, we weight the norm of the Hessian with a mask that is only active far from the samples to avoid smoothing the normals close to the point cloud, as [10] does in Figure 6(j), and do not square the Frobenius norm to allow for sharp changes in surface orientation that appear whenever two surface sheets with opposite orientations are spatially close. As a result, we are able to reconstruct the correct topology of the statue in Figure 5(d), while [10] merges the two surface sheets at the statue’s leg because it overpenalizes sharp changes in orientation and over-smooths the normals in Figure 5(c). This is an important feature for the reconstruction of surfaces with touching sheets and highlights the importance of both robust data terms and regularizers for the estimation of a continuous surface from a set of sparse points.

8 Conclusions

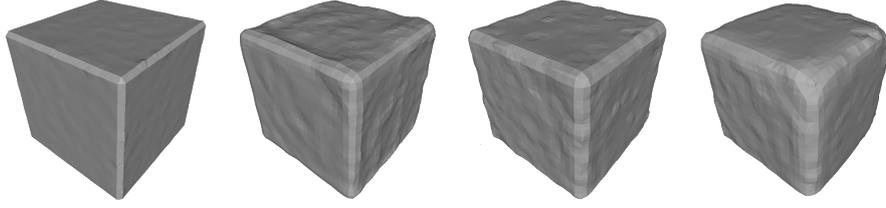
We reconstruct surfaces from corrupted point clouds by formulating the problem as a convex minimization that is robust to outliers, avoids the shrinking bias of least squares, and is able to recover sharp corners as well as smooth regions. To this purpose, we substitute the usual least-squares penalties for robust Huber loss functions and introduce a regularizer that allows sharp changes in the surface’s orientation and is only active far from the samples to avoid over-smoothing of textured regions. These properties are demonstrated experimentally with both clean and corrupted and synthetic and real data.

For an efficient parametrization, we approximate the implicit function with hierarchical basis functions that are spatially localized to match the resolution of the data. After investigating discretizations of different complexity and smoothness, we conclude that a linear hierarchical B-spline basis over an octree offers the best performance for our data resolution, allowing us to both represent smooth regions and sharp corners with piecewise linear elements.

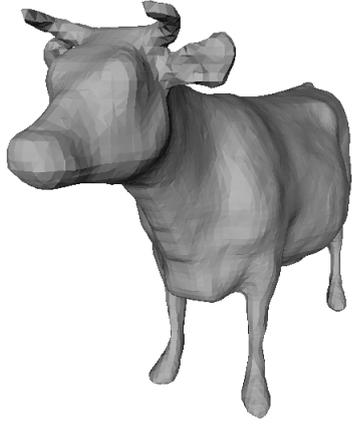
The proposed discretizations lend themselves to a multiresolution strategy for the minimization. This is particularly advantageous because the use of robust Huber penalties in our variational model leads to a more complex minimization problem that we solve efficiently with a primal-dual algorithm that exploits its convexity and is easy to parallelize. As a result, our method is just an order of magnitude slower than the least-squares models [23, 10, 24] for a more robust and accurate results.

Acknowledgments.

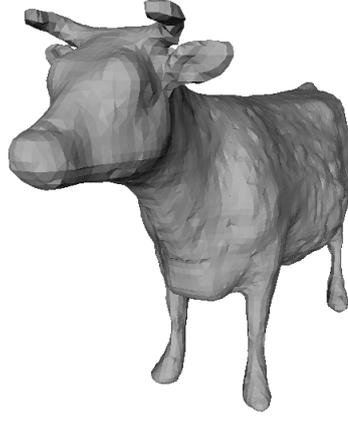
V. Estellers was supported by the SNSF grant SNSF-P300P2_161038 . M.A. Scott was partially supported by the AFOSR grant FA9550-14-1-0113.



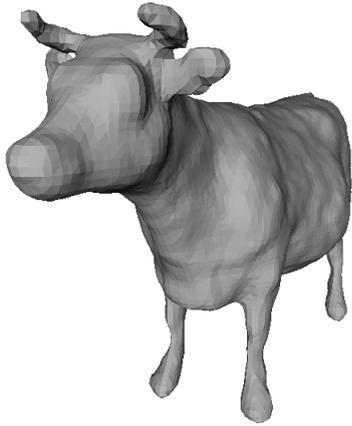
(a) Huber, clean cube (b) least squares, clean (c) Huber, noisy cube (d) least squares, noisy



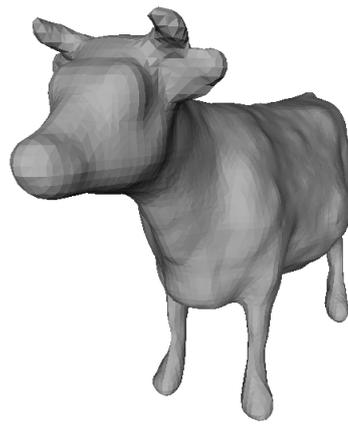
(e) Huber, clean samples



(f) least squares, clean samples



(g) Huber, noisy samples



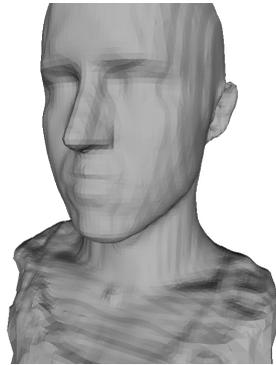
(h) least squares, noisy samples



(i) point cloud



(j) Huber



(k) least squares

Figure 2: Comparison of reconstruction models with least-squares and Huber data terms with a linear HBS discretization. The Huber penalty of our reconstruction model offers 3 advantages in comparison to a least-squares penalties: first, it is able to reconstruct sharp corners, as observed for the clean and noisy samples of cube in Figures 2(a)-2(d); second, it avoids the shrinkage of thin structures, like the ears of the cow in Figures 2(e)-2(h)), and third, it is robust to outliers due to scanning artifacts without over-smoothing and shrinking, as Figures 2(i)-2(k) show for the ear region.

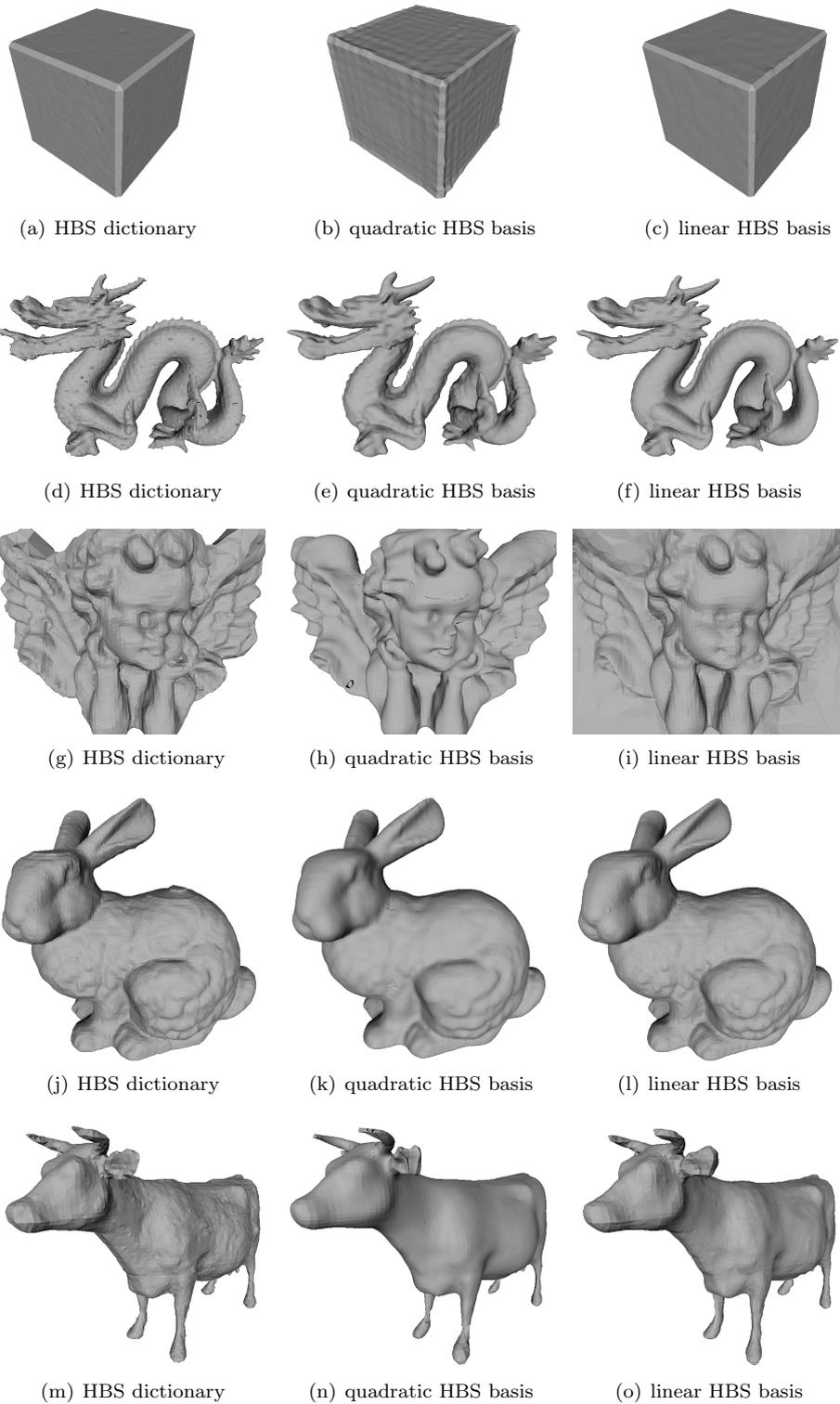


Figure 3: Reconstructed surface and execution time with different discretizations. The quadratic HBS basis produces smoother results but is an order of magnitude slower than the octree discretizations, while the octree representations differ in the reconstruction of smooth regions: the quadratic splines produce wiggles in textures areas, and the mixed FE-FD recovers smooth regions in the cow's torso.

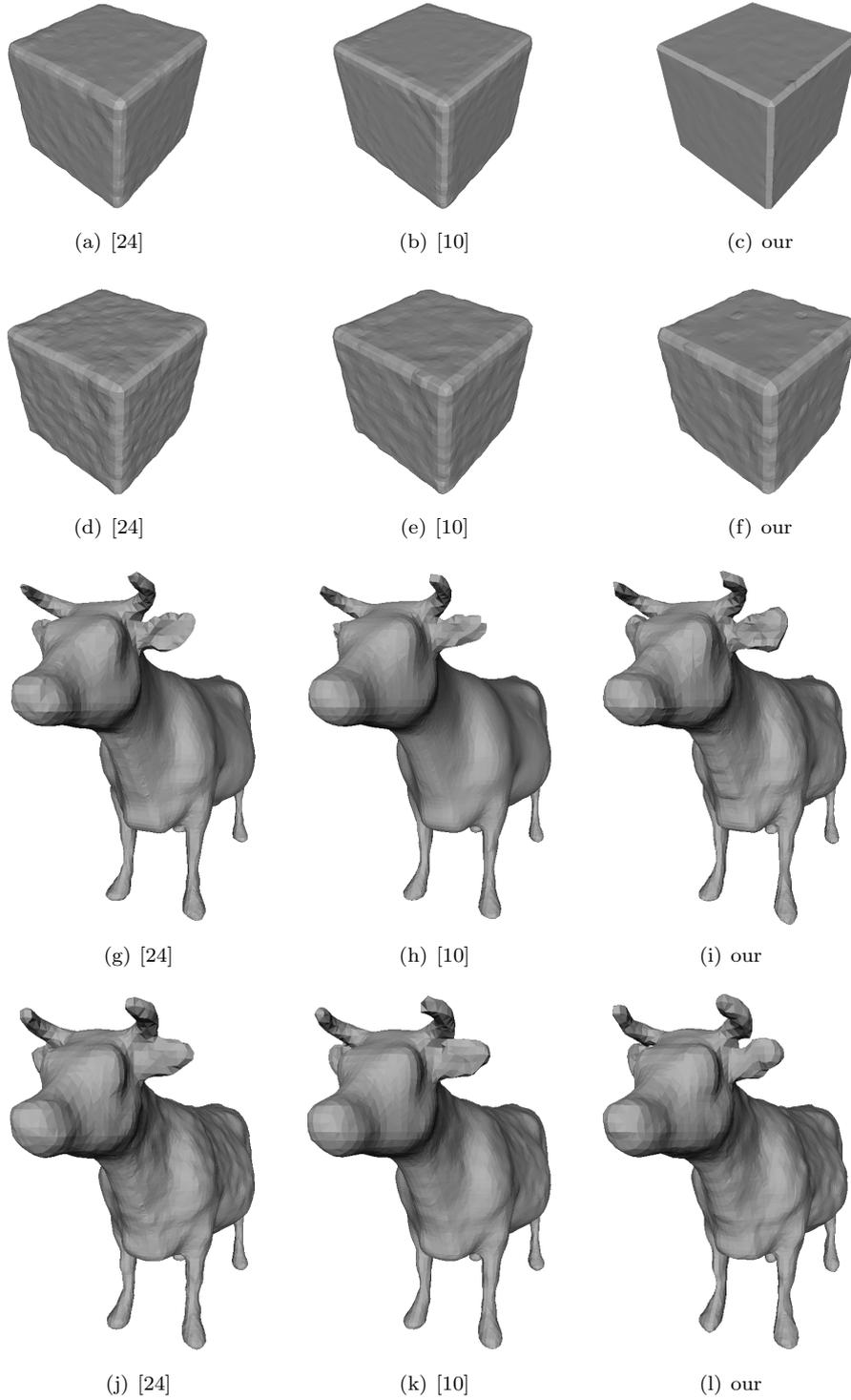


Figure 4: Comparison to state-of-the-art for synthetic data, with and without noise. The models of [24, 10] round the cube’s corners and shrink the ears of the cow because they rely of least-squares data terms, while our model overcomes these artifacts with a robust Huber penalty that recovers the sharp corner in the cube and avoids the shrinking bias that merges the ears and horns of the cow in least-squares reconstructions [24, 10] from noisy samples.

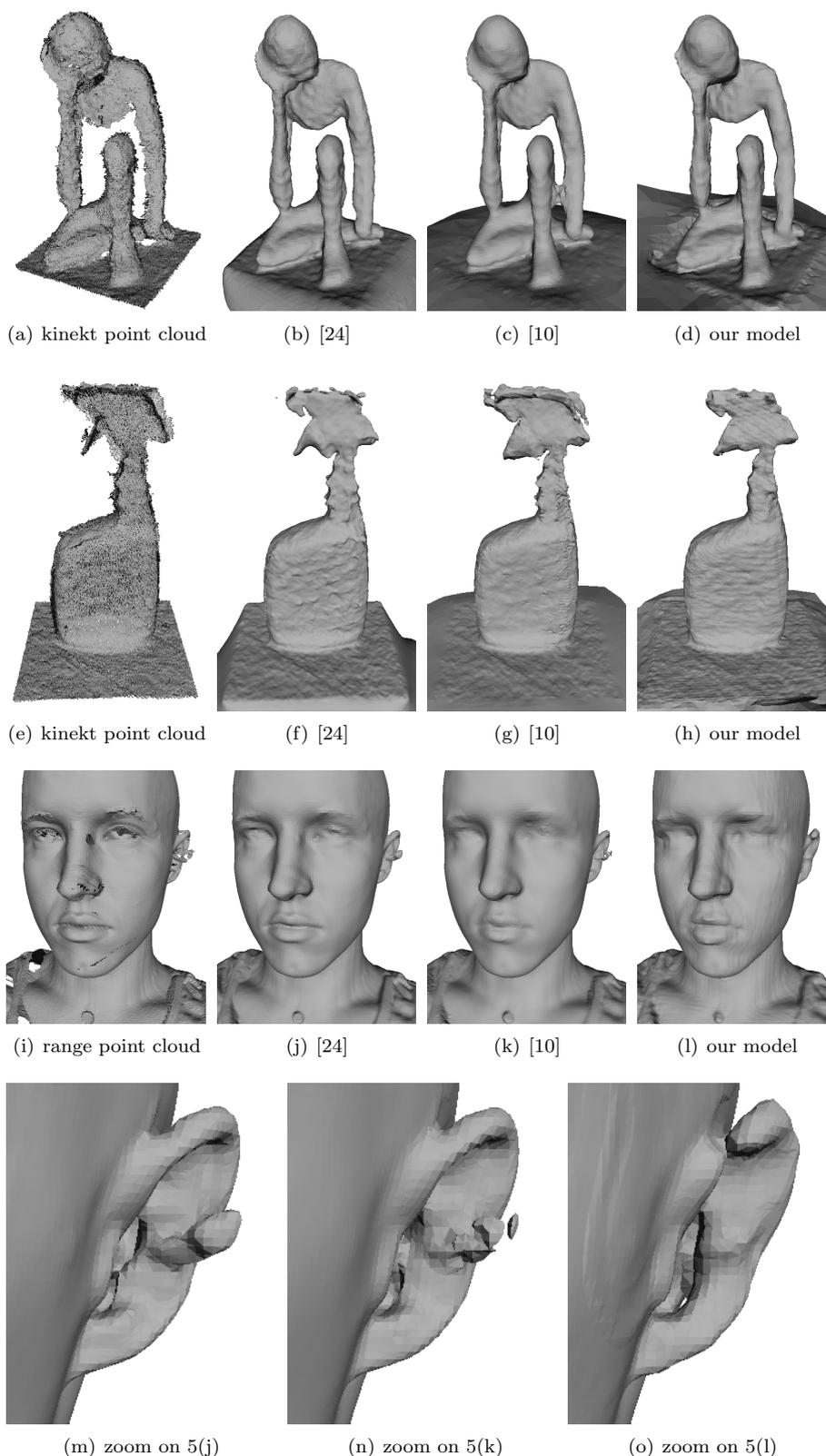


Figure 5: Comparison to state-of-the-art models [24, 10] with noisy point clouds captured with a kinect camera [7] and a range scanner. The models of [24, 10] are not robust to outliers and lead to reconstructions with incorrect topologies: [10] merges the two surface sheets in the statue’s knee because it relies on a least-squares regularizer of the Hessian that smooths the normal samples of close surface sheets, while both [24] and Calakli2011 reconstructions reproduce the scanning artifacts of the point cloud at the head of the cleaner bottle and the ear of the human head. Our model is robust to outliers and estimates the correct topology.

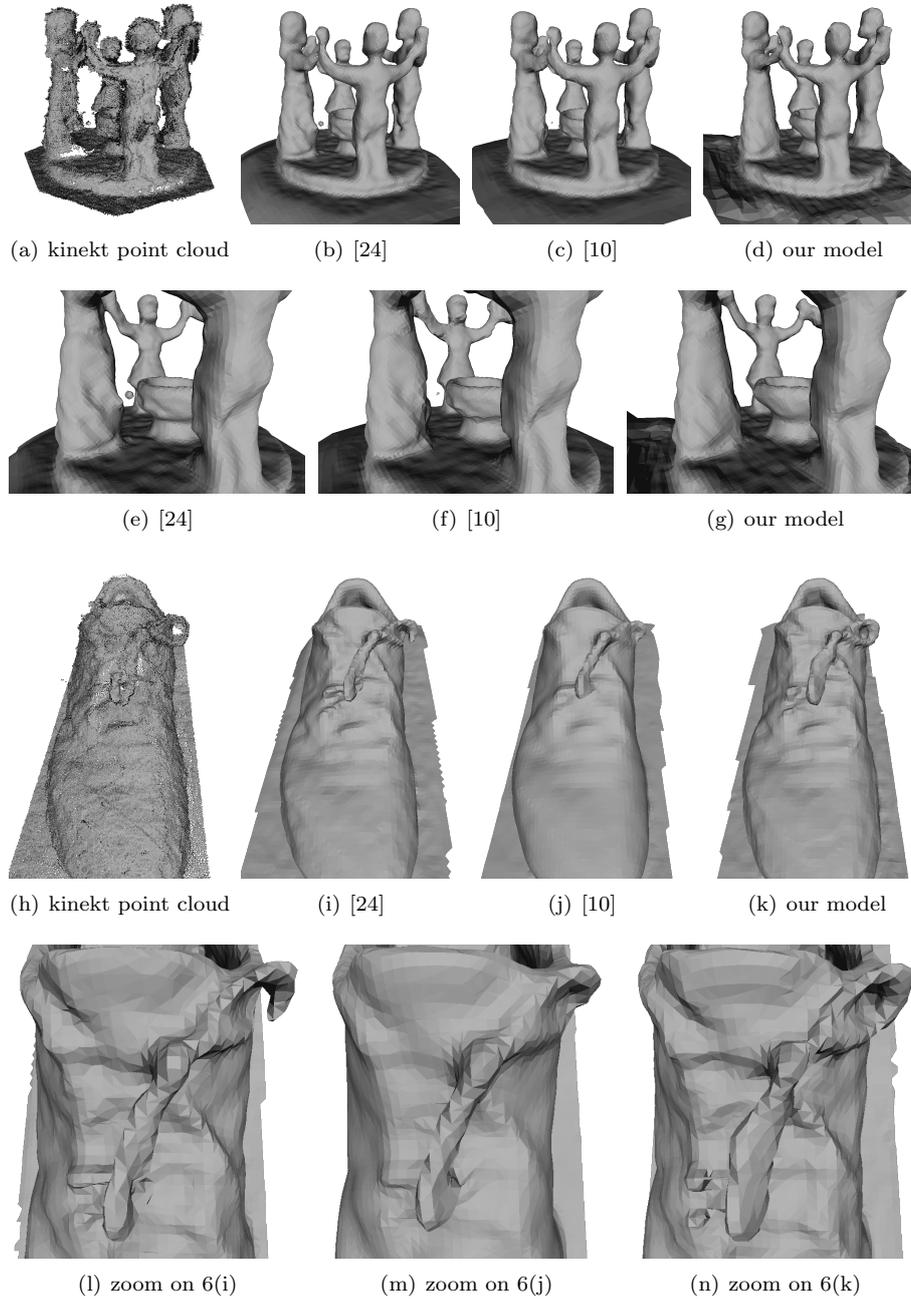


Figure 6: Comparison to state-of-the-art models [24, 10] with noisy point clouds captured with a kinect camera [7]. The models of [24, 10] are not robust to outliers and lead to reconstructions with incorrect topologies: there are residual spheres due to a cluster of outliers in the reconstruction of the dancers in Figures 6(b) and 6(c), more visible in the close-up views of Figures 6(e) and 6(f). Similarly, only our model is able to capture the right topology of the shoelace in the reconstruction of the scanned shoe of Figures 6(i)-6(n).

References

- [1] Anders Adamson and Marc Alexa. Point-sampled cell complexes. *ACM Transactions on Graphics (TOG)*, 1(212):671–680, 2006.
- [2] Marc Alexa, Johannes Behr, Daniel Cohen-or, Shachar Fleishman, David Levin, and Claudio T. Silva. Computing and rendering point set surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 9(1):3–15, January 2003.
- [3] Nina Amenta, Marshall Bern, and Manolis Kamvyselis. A New Voronoi-Based Surface Reconstruction Algorithm. *ACM SIGGRAPH*, pages 415–421, 1998.
- [4] Nina Amenta, Sunghee Choi, and Ravi Krishna Kolluri. The power crust. *ACM symposium on Solid Modeling and Applications*, pages 249–266, 2001.
- [5] Nina Amenta and Yong J. Kil. Defining point-set surfaces. *ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH*, 23(3):264–270, 2004.
- [6] Muthuvel Arigovindan, Michael Sühling, Patrick Hunziker, and Michael Unser. Variational Image Reconstruction from Arbitrarily Spaced Samples: a Fast Multiresolution Spline Solution. *IEEE Transactions on Image Processing*, 14(4):450–60, April 2005.
- [7] J. Balzer, M. Peters, and S. Soatto. Volumetric Reconstruction Applied to Perceptual Studies of Size and Weight. *IEEE WACV*, 704, November 2013.
- [8] Jonathan Balzer and Thomas Morwald. Isogeometric finite-elements methods and variational reconstruction tasks in visionA perfect match. *International Conference on Computer Vision and Pattern Recognition*, pages 1624–1631, 2012.
- [9] Jean-Daniel Boissonnat and Steve Oudot. Provably good sampling and meshing of surfaces. *Graphical Models*, 67(5):405–451, 2005.
- [10] F. Calakli and G. Taubin. SSD: Smooth Signed Distance Surface Reconstruction. *Computer Graphics Forum*, 30(7):1993–2002, September 2011.
- [11] J C Carr, R K Beatson, and T R Evans. Reconstruction and Representation of 3D Objects with Radial Basis Functions. *ACM SIGGRAPH*, pages 67–76, 2001.
- [12] Antonin Chambolle and Thomas Pock. A First-Order Primal-Dual Algorithm for Convex Problems with Applications to Imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145, December 2010.
- [13] Ming Chuang and Michael Kazhdan. Interactive and anisotropic geometry processing using the screened Poisson equation. *ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH*, 30(4):57, 2011.
- [14] U Clarenz, U Diewald, and M Rumpf. Anisotropic geometric diffusion in surface processing. *Proceedings of the conference on Visualization '00 (VIS '00)*, pages 397–405, 2000.
- [15] Brian Curless and Marc Levoy. A Volumetric Method for Building Complex Models from Range Images Volumetric integration. *ACM SIGGRAPH*, pages 303–312, 1996.

- [16] Joel Daniels II, Linh K Ha, Tilo Ochotta, and Claudio T. Silva. Robust Smooth Feature Extraction from Point Clouds. *IEEE International Conference on Shape Modeling and Applications (SMI'07)*, pages 123 – 136, 2007.
- [17] TK Dey and S Goswami. Tight cocone: a water-tight surface reconstructor. *Proceedings of the eighth ACM symposium on Solid modeling and applications*, pages 127–134, 2003.
- [18] Virginia Estellers, Michael A. Scott, Kevin Tew, and Stefano Soatto. Robust Poisson Surface Reconstruction. *Scale Space and Variational Methods in Computer Vision*, under revi.
- [19] Shachar Fleishman, Daniel Cohen-or, and Claudio T. Silva. Robust moving least-squares fitting with sharp features. *ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH 2005*, 44(3):544–552, 2005.
- [20] David Forsey and Richard Bartels. Hierarchical B-Spline Refinement. *Computer Graphics*, 22(4):205–212, 1988.
- [21] Gael Guennebaud and Markus Gross. Algebraic Point Set Surfaces. *ACM Transactions on Graphics*, 26(3):1–10, 2007.
- [22] Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle. Surface reconstruction from unorganized points. *Conference on Computer graphics and interactive techniques SIGGRAPH*, pages 71–78, 1992.
- [23] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. *Eurographics Symposium on Geometry Processing*, pages 61–70, 2006.
- [24] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Transactions on Graphics*, 32(3):1–13, June 2013.
- [25] Patrick Labatut, Jean-Philippe Pons, and Renaud Keriven. Robust and efficient surface reconstruction from range data. *Computer Graphics Forum*, 28(8):2275–2290, 2009.
- [26] Lee, E.T.Y. A simplified B-spline computation routine. *Computing*, 29(4):365–371, 1982.
- [27] P Les and T Wayne. The Nurbs Book. *Springer Verlag, Berlin*, 1997.
- [28] David Levin. Mesh-Independent surface Interpolation. *Geometric modeling for scientific visualization*, pages 37–49, 2004.
- [29] Yaron Lipman, Daniel Cohen-or, and David Levin. Data-dependent MLS for faithful surface approximation. *Eurographics Symposium on Geometry Processing*, pages 59–67, 2007.
- [30] J. Manson, G. Petrova, and S. Schaefer. Streaming surface reconstruction using wavelets. *Computer Graphics Forum*, 27(5):1411–1420, 2008.
- [31] G. M. Morton. A computer oriented geodetic data base and a new technique in file sequencing. *International Business Machines Company*, 1966.
- [32] Diego Nehab, Szymon Rusinkiewicz, James Davis, and Ravi Ramamoorthi. Efficiently combining positions and normals for precise 3D geometry. *ACM Transactions on Graphics*, 24(3):536, July 2005.

- [33] Yutaka Ohtake, Alexander Belyaev, Marc Alexa, Greg Turk, and Hans-Peter Seidel. Multi-level partition of unity implicits. *ACM Transactions on Graphics*, 22(3):463, July 2003.
- [34] A. C. Öztireli, Gael Guennebaud, and Markus Gross. Feature Preserving Point Set Surfaces based on Non-Linear Kernel Regression. *Computer Graphics Forum*, 28(2):493–501, April 2009.
- [35] Joshua Podolak and Szymon Rusinkiewicz. Atomic Volumes for Mesh Completion. *Symposium on Geometry Processing*, pages 33–41, 2005.
- [36] R. Tyrrell Rockafellar. *Convex analysis*. Number 28. Princeton university press, 1997.
- [37] M. A. Scott, D. C. Thomas, and E. J. Evans. Isogeometric spline forests. *Computer Methods in Applied Mechanics and Engineering*, 269:222–264, 2014.
- [38] Scott Schaefer and Joe Warren. Dual marching cubes: Primal contouring of dual grids. *Computer Graphics Forum*, 24:195–201, 2005.
- [39] Chen Shen, James F. O’Brien, and Jonathan R. Shewchuk. Interpolating and approximating implicit surfaces from polygon soup. *ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH 2004*, 23(3):896–904, 2004.
- [40] Jonathan Richard Shewchuk and James F O Brien. Spectral Surface Reconstruction from Noisy Point Clouds. *ACM SIGGRAPH symposium on Geometry processing*, 14:11–21, 2004.
- [41] Gabriele Steidl, Stephan Didas, and Julia Neumann. Splines in Higher Order TV Regularization. *International Journal of Computer Vision*, 70(3):241–255, December 2006.
- [42] Tolga Tasdizen, Ross Whitaker, Paul Burchard, and Stanley Osher. Geometric surface smoothing via anisotropic diffusion of normals. *Proceedings of IEEE Visualization (VIS2002)*, pages 125–132, 2002.
- [43] Michael Unser. Splines a Perfect Fit for Signal and Image Processing. *IEEE Signal Processing Magazine*, 16(6):22–38, 1999.
- [44] A.-V. Vuong, C. Giannelli, B. Jüttler, and B. Simeon. A hierarchical approach to adaptive local refinement in isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 200(49-52):3554–3567, 2011.