Screened Poisson Surface Reconstruction

MICHAEL KAZHDAN
Johns Hopkins University
and
HUGUES HOPPE
Microsoft Research

Poisson surface reconstruction creates watertight surfaces from oriented point sets. In this work we extend the technique to explicitly incorporate the points as interpolation constraints. The extension can be interpreted as a generalization of the underlying mathematical framework to a screened Poisson equation. In contrast to other image and geometry processing techniques, the screening term is defined over a sparse set of points rather than over the full domain. We show that these sparse constraints can nonetheless be integrated efficiently. Because the modified linear system retains the same finite-element discretization, the sparsity structure is unchanged, and the system can still be solved using a multigrid approach. Moreover we present several algorithmic improvements that together reduce the time complexity of the solver to linear in the number of points, thereby enabling faster, higher-quality surface reconstructions.

Categories and Subject Descriptors: I.3.5 [**Computer Graphics**]: Computational Geometry and Object Modeling

Additional Key Words and Phrases: screened Poisson equation, adaptive octree, finite elements, surface fitting

ACM Reference Format:

Kazhdan, M., and Hoppe, H. Screened Poisson surface reconstruction. ACM Trans. Graph. NN, N, Article NN (Month YYYY), PP pages.
DOI = 10.1145/XXXXXXX.YYYYYYY
http://doi.acm.org/10.1145/XXXXXXX.YYYYYYY

1. INTRODUCTION

Poisson surface reconstruction [Kazhdan et al. 2006] is a well known technique for creating watertight surfaces from oriented point samples acquired with 3D range scanners. The technique is resilient to noisy data and misregistration artifacts. However, as noted by several researchers, it suffers from a tendency to over-smooth the data [Alliez et al. 2007; Manson et al. 2008; Calakli and Taubin 2011; Berger et al. 2011; Digne et al. 2011].

In this work, we explore modifying the Poisson reconstruction algorithm to incorporate positional constraints. This modification is inspired by the recent reconstruction technique of Calakli and Taubin [2011]. It also relates to recent work in image and geometry processing [Nehab et al. 2005; Bhat et al. 2008; Chuang and Kazhdan 2011], in which a data fidelity term is used to "screen" the associated Poisson equation. In our surface reconstruction context, this screening term corresponds to a soft constraint that encourages the reconstructed isosurface to pass through the input points.

The approach we propose differs from the traditional screened Poisson formulation in that the position and gradient constraints are defined over different domain types. Whereas gradients are constrained over the full 3D space, positional constraints are introduced only over the input points, which lie near a 2D manifold. We show how these two types of constraints can be efficiently integrated, so that we can leverage the original multigrid structure to solve the linear system without incurring a significant overhead in space or time.

To demonstrate the benefits of screening, Figure 1 compares results of the traditional Poisson surface reconstruction and the screened Poisson formulation on a subset of 11.4M points from the scan of Michelangelo's David [Levoy et al. 2000]. Both reconstructions are computed over a spatial octree of depth 10, corresponding to an effective voxel resolution of 1024^3. Screening generates a model that better captures the input data (as visualized by the surface cross-sections overlaid with the projection of nearby samples), even though both reconstructions have similar complexity (6.8M and 6.9M triangles respectively) and required similar processing time (230 and 272 seconds respectively, without parallelization).[1]

Another contribution of our work is to modify both the octree structure and the multigrid implementation to reduce the time complexity of solving the Poisson system from log-linear to linear in the number of input points. Moreover we show that hierarchical point clustering enables screened Poisson reconstruction to attain this same linear complexity.

2. RELATED WORK

Reconstructing surfaces from scanned points is an important and extensively studied problem in computer graphics. The numerous approaches can be broadly categorized as follows.

Combinatorial Algorithms. Many schemes form a triangulation using a subset of the input points [Cazals and Giesen 2006]. Space is often discretized using a tetrahedralization or a voxel grid, and the resulting elements are partitioned into inside and outside regions using an analysis of cells [Amenta et al. 2001; Boissonnat and Oudot 2005; Podolak and Rusinkiewicz 2005], eigenvector computation [Kolluri et al. 2004], or graph cut [Labatut et al. 2009; Hornung and Kobbelt 2006].

Implicit Functions. In the presence of sampling noise, a common approach is to fit the points using the zero set of an implicit function, such as a sum of radial bases [Carr et al. 2001] or piecewise polynomial functions [Ohtake et al. 2005; Nagai et al. 2009]. Many techniques estimate a signed-distance function [Hoppe et al. 1992;

[1] The performance of the unscreened solver is measured using our implementation with screening weight set to zero. The implementation of the original Poisson reconstruction runs in 412 seconds.

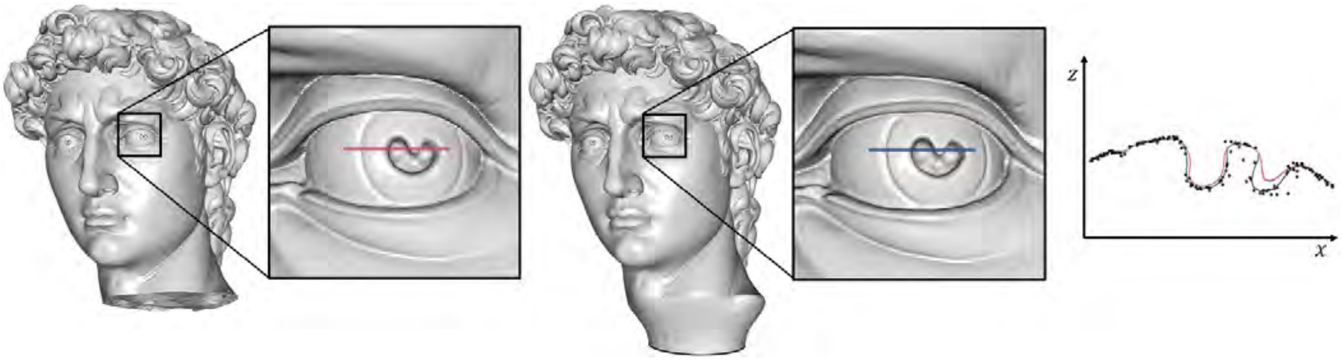


Fig. 1: Reconstruction of the David head‡, comparing traditional Poisson surface reconstruction (left) and screened Poisson surface reconstruction which incorporates point constraints (center). The rightmost diagram plots pixel depth (z) values along the colored segments together with the positions of nearby samples. The introduction of point constraints significantly improves fit accuracy, sharpening the reconstruction without amplifying noise.

Bajaj et al. 1995; Curless and Levoy 1996]. If the input points are unoriented, an important step is to correctly infer the sign of the resulting distance field [Mullen et al. 2010].

Our work extends Poisson surface reconstruction [Kazhdan et al. 2006], in which the implicit function corresponds to the model's indicator function χ. The function χ is often defined to have value 1 inside and value 0 outside the model. To simplify the derivations, in this paper we define χ to be $\frac{1}{2}$ inside and $\frac{-1}{2}$ outside, so that its *zero* isosurface passes near the points. The function χ is solved using a Laplacian system discretized over a multiresolution B-spline basis, as reviewed in Section 3.

Alliez et al. [2007] form a Laplacian system over a tetrahedralization, and constrain the solution's biharmonic energy; the desired function is obtained as the solution to an eigenvector problem. Manson et al. [2008] represent the indicator function χ using a wavelet basis, and efficiently compute the basis coefficients using simple local sums over an adapted octree.

Calakli and Taubin [2011] optimize a signed-distance function to have value zero at the points, have derivatives that agree with the point normals, and minimize a Hessian smoothness norm. The resulting optimization involves a bilaplacian operator, which requires estimating derivatives of higher order than in the Laplacian. The reconstructed surfaces are shown to have good accuracy, strongly suggesting the importance of explicitly fitting the points within the optimization. This motivated us to explore whether a Laplacian system could be extended in this respect, and also be compatible with a multigrid solver.

Screened Poisson Surface Fitting. The method of Nehab et al. [2005], which simultaneously fits position and normal constraints, may also be viewed as the solution of a screened Poisson equation. The fitting algorithm assumes that a 2D parametric domain (i.e., a plane or triangle mesh) is already established. The position and derivative constraints are both defined over this 2D domain.

In contrast, in Poisson surface reconstruction the 2D domain manifold is initially unknown, and therefore the goal is to infer an indicator function χ rather than a parametric function. This leads to a hybrid problem with derivative (Laplacian) constraints defined densely over 3D and position constraints defined sparsely on the set of points sampled near the unknown 2D manifold.

3. REVIEW OF POISSON SURFACE RECONSTRUCTION

The approach of Poisson surface reconstruction is based on the observation that the (inward pointing) normal field of the boundary of a solid can be interpreted as the gradient of the solid's indicator function. Thus, given a set of oriented points sampling the boundary, a watertight mesh can be obtained by (1) transforming the oriented point samples into a continuous vector field in 3D, (2) finding a scalar function whose gradients best match the vector field, and (3) extracting the appropriate isosurface.

Because our work focuses primarily on the second step, we review it here in more detail.

Scalar Function Fitting. Given a vector field $\vec{V}:\mathbb{R}^3\to\mathbb{R}^3$, the goal is to solve for the scalar function $\chi:\mathbb{R}^3\to\mathbb{R}$ minimizing:

$$E(\chi)=\int\|\nabla\chi(p)-\vec{V}(p)\|^2dp. \qquad (1)$$

Using the Euler-Lagrange formulation, the minimum is obtained by solving the Poisson equation:

$$\Delta\chi=\nabla\cdot\vec{V}.$$

System Discretization. The Galerkin formulation is used to transform this into a finite-dimensional system [Fletcher 1984]. First, a basis $\{B_1,\dots,B_N\}:\mathbb{R}^3\to\mathbb{R}$ is chosen, namely a collection of trivariate (usually triquadratic) B-spline functions. With respect to this basis, the discretization becomes:

$$\langle\Delta\chi,B_i\rangle_{[0,1]^3}=\langle\nabla\cdot\vec{V},B_i\rangle_{[0,1]^3}\qquad 1\le i\le N$$

where $\langle\cdot,\cdot\rangle_{[0,1]^3}$ is the standard inner-product on the space of (scalar- and vector-valued) functions defined on the unit cube:

$$\begin{aligned}\langle F,G\rangle_{[0,1]^3} &= \int_{[0,1]^3}F(p)\cdot G(p)\,dp,\\ \langle\vec{U},\vec{V}\rangle_{[0,1]^3} &= \int_{[0,1]^3}\langle\vec{U}(p),\vec{V}(p)\rangle\,dp.\end{aligned}$$

Since the solution is itself expressed in terms of the basis functions:

$$\chi(p)=\sum_{i=1}^{N}x_iB_i(p),$$

finding the coefficients $\{x_i\}$ of the solution reduces to solving the linear system $Ax = b$ where:

$$A_{ij} = \langle \nabla B_i, \nabla B_j \rangle_{[0,1]^3} \qquad \text{and} \qquad b_i = \langle \vec{V}, \nabla B_i \rangle_{[0,1]^3}. \tag{2}$$

The basis functions $\{B_1, \dots, B_N\}$ are chosen to be compactly supported, so most pairs of functions do not have overlapping support, and thus the matrix A is sparse.

Because the solution is expected to be smooth away from the input samples, the linear system is discretized by first adapting an octree to the input samples and then associating an (appropriately scaled and translated) trivariate B-spline function to each octree node. This provides high-resolution detail in the vicinity of the surface while reducing the overall dimensionality of the system.

System Solution. Given the hierarchy defined by an octree of depth D, a multigrid approach is used to solve the linear system. The basis functions are partitioned according to the depths of their associated nodes and, for each depth d, a linear system $A^d x^d = b^d$ is defined using the corresponding B-splines $\{B_1^d, \dots, B_{N_d}^d\}$, such that $\chi(p) = \sum_{d=0}^{D} \sum_i x_i^d B_i^d(p)$.

Because the octree-selected B-spline functions do not form a complete grid at each depth, it is generally not possible to prolong the solution x^d at depth d into the solution x^{d+1} at depth $d+1$. (The B-spline associated with a given node is a sum of B-spline functions associated not only with its own child nodes, but also with child nodes of its neighbors.) Instead, the constraints at depth $d+1$ are adjusted to account for the part of the solution already realized at coarser depths.

Pseudocode for a cascadic solver, where the solution is only relaxed on the up-stroke of the V-cycle, is given in Algorithm 1.

Algorithm 1: Cascadic Poisson Solver

```
For $d \in \{0,\dots,D\}$                        Iterate from coarse to fine
    For $d' \in \{0,\dots,d-1\}$                 Remove the constraints
        $b^d = b^d - A^{dd'} x^{d'}$                 met at coarser depths
    Relax $A^d x^d = b^d$                        Adjust the system at depth d
```

Here, $A^{dd'}$ is the $N_d \times N_{d'}$ matrix used to transform solution coefficients at depth d' into constraints at depth d:

$$A_{ij}^{dd'} = \langle \nabla B_i^d, \nabla B_j^{d'} \rangle_{[0,1]^3}.$$

Note that, by definition, $A^d = A^{dd}$.

Isosurface Extraction. Solving the Poisson equation, one obtains a function χ that approximates the indicator function. Ideally, the function's zero level-set should therefore correspond to the desired surface. In practice however, the function χ can differ from the true indicator function due to several sources of error:

- The point sampling may be noisy, possibly containing outliers.
- The Galerkin discretization is only an approximation of the continuous problem.
- The point sampling density is approximated during octree construction.

To mitigate these errors, in [Kazhdan et al. 2006] the implicit function is adjusted by globally subtracting the average value of the function at the input samples.

4. INCORPORATING POINT CONSTRAINTS

The original Poisson surface reconstruction algorithm adjusts the implicit function using a single global offset such that its average value at all points is zero. However, the presence of errors can cause the implicit function to drift so that no global offset is satisfactory. Instead, we seek to explicitly interpolate the points.

Given the set of input points $\mathscr{P}$ with weights $w : \mathscr{P} \to \mathbb{R}^{\geq 0}$, we add to the energy of Equation 1 a term that penalizes the function's deviation from zero at the samples:

$$E(\chi) = \int \|\vec{V}(p) - \nabla\chi(p)\|^2 dp + \frac{\alpha \cdot \text{Area}(\mathscr{P})}{\sum_{p \in \mathscr{P}} w(p)} \sum_{p \in \mathscr{P}} w(p) \chi^2(p) \tag{3}$$

where α is a weight that trades off the importance of fitting the gradients and fitting the values, and Area($\mathscr{P}$) is the area of the reconstructed surface, estimated by computing the local sampling density as in [Kazhdan et al. 2006]. In our implementation, we set the per-sample weights $w(p) = 1$, although one can also use confidence values if these are available.

The energy can be expressed concisely as

$$E(\chi) = \langle \vec{V} - \nabla\chi, \vec{V} - \nabla\chi \rangle_{[0,1]^3} + \alpha \langle \chi, \chi \rangle_{(w,\mathscr{P})} \tag{4}$$

where $\langle \cdot, \cdot \rangle_{(w,\mathscr{P})}$ is the bilinear, symmetric, positive, semi-definite form on the space of functions in the unit-cube, obtained by taking the weighted sum of function values:

$$\langle F, G \rangle_{(w,\mathscr{P})} = \frac{\text{Area}(\mathscr{P})}{\sum_{p \in \mathscr{P}} w(p)} \sum_{p \in \mathscr{P}} w(p) \cdot F(p) \cdot G(p).$$

4.1 Interpretation as a Screened Poisson Equation

The energy in Equation 4 combines a gradient constraint integrated over the spatial domain with a value constraint summed at discrete points. As shown in the appendix, its minimization can be interpreted as a screened Poisson equation $(\Delta - \alpha\tilde{I})\chi = \nabla \cdot \vec{V}$ with an appropriately defined operator $\tilde{I}$.

4.2 Discretization

We apply a discretization similar to that in Section 3 to the minimization of the energy in Equation 4. The coefficients of the solution χ with respect to the basis $\{B_1, \dots, B_N\}$ are again obtained by solving a linear system of the form $Ax = b$. The right-hand-side b is unchanged because the constrained value at the sample points is zero. Matrix A now includes the point constraints:

$$A_{ij} = \langle \nabla B_i, \nabla B_j \rangle_{[0,1]^3} + \alpha \langle B_i, B_j \rangle_{(w,\mathscr{P})}. \tag{5}$$

Note that incorporating the point constraints does not change the sparsity of matrix A because $B_i(p) \cdot B_j(p)$ is nonzero only if the supports of the two functions overlap, in which case the Poisson equation has already introduced a nonzero entry in the matrix.

As in Section 3, we solve this linear system using a cascadic multigrid algorithm – iterating over the octree depths from coarsest to finest, adjusting the constraints, and relaxing the system. Similar to Equation 5, the matrix used to transform a solution at depth d' to a constraint at depth d is expressed as:

$$A_{ij}^{dd'} = \langle \nabla B_i^d, \nabla B_j^{d'} \rangle_{[0,1]^3} + \alpha \langle B_i^d, B_j^{d'} \rangle_{(w,\mathscr{P})}.$$

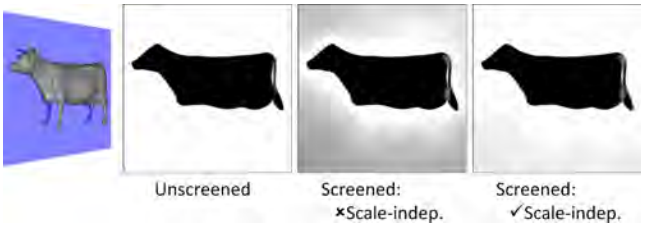


Fig. 2: Visualizations of the reconstructed implicit function along a planar slice through the cow[‡] (shown in blue on the left), for the original Poisson solver, and for the screened Poisson solver without and with scale-independent screening.

This operator adjusts the constraint b^d (line 3 of Algorithm 1) not only by removing the Poisson constraints met at coarser resolutions, but also by modifying the constrained values at points where the coarser solution does not evaluate to zero.

4.3 Scale-Independent Screening

To balance the two energy terms in Equation 3, it is desirable to adjust the screening parameter α such that (1) the reconstructed surface shape is invariant under scaling of the input points with respect to the solver domain, and (2) the prolongation of a solution at a coarse depth is an accurate estimate of the solution at a finer depth in the cascadic multigrid approach. We achieve both these goals by adjusting the relative weighting of position and gradient constraints across the different octree depths. Noting that the magnitude of the gradient constraint scales with resolution, we double the weight of the interpolation constraint with each depth:

$$A_{ij}^{dd'} = \langle \nabla B_i^d, \nabla B_j^{d'} \rangle_{[0,1]^3} + 2^d \alpha \langle B_i^d, B_j^{d'} \rangle_{(w,\mathscr{P})}.$$

The adaptive weight of 2^d is chosen to keep the Laplacian and screening constraints around the surface in balance. To see this, assume that the points are locally planar, and consider the row of the system matrix corresponding to an octree node overlapping the points. The coefficients of the system in that row are the sum of Laplacian and screening terms. If we consider the rows corresponding to the child nodes that overlap the surface, we find that the contribution from the Laplacian constraints scales by a factor of 1/2 while the contribution from the screening term scales by a factor of 1/4.[2] Thus, scaling the screening weights by a factor of two with each resolution keeps the two terms in balance.

Figure 2 shows the benefit of scale-independent screening in reconstructing a cow model. The leftmost image shows a plane passing through the bounding cube of the cow, and the images to the right show the values of the computed indicator function along that plane, for different implementations of the solver. As the figure shows, the unscreened Poisson solver provides a good approximation of the indicator functions, with values inside (resp. outside) the surface approximately 1/2 (resp. -1/2). However, applying the same solver to the screened Poisson equation (second from right) provides a solution that is only correct near the input samples and returns to zero near the faces of the bounding cube, potentially resulting in spurious surface sheets away from the surface. It is only with scale-independent screening (right) that we obtain a high-quality solution to the screened Poisson equation.

Scale-Independence. Using this resolution adaptive weighting, our system has the property that the reconstruction obtained by solving at depth D is identical to the reconstruction that would be obtained by scaling the point set by $1/2$ and solving at depth $D+1$.

To see this, we consider the two energies that guide the reconstruction, $E_{\vec{V}}(\chi)$ measuring the extent to which the gradients of the solution match the prescribed vector field, and $E_{(w,\mathscr{P})}(\chi)$ measuring the extent to which the solution meets the screening constraint:

$$\begin{aligned} E_{\vec{V}}(\chi) &= \int \left\| \vec{V}(p) - \nabla\chi(p) \right\|^2 dp \\ E_{(w,\mathscr{P})}(\chi) &= \frac{\mathrm{Area}(\mathscr{P})}{\sum_{p\in\mathscr{P}} w(p)} \sum_{p\in\mathscr{P}} w(p)\chi^2(p). \end{aligned}$$

Scaling by $1/2$, we obtain a new point set $(\tilde{w}, \tilde{\mathscr{P}})$ with positions scaled by $1/2$, unchanged weights, $\tilde{w}(p) = w(2p)$, and scaled area, $\mathrm{Area}(\tilde{\mathscr{P}}) = \mathrm{Area}(\mathscr{P})/4$; a new scalar field, $\tilde{\chi}(p) = \chi(2p)$; and a new vector field, $\tilde{\vec{V}}(p) = 2\vec{V}(2p)$. Computing the corresponding energies, we get:

$$E_{\tilde{\vec{V}}}(\tilde{\chi}) = \frac{1}{2} E_{\vec{V}}(\chi) \quad \text{and} \quad E_{(\tilde{w},\tilde{\mathscr{P}})}(\tilde{\chi}) = \frac{1}{4} E_{(w,\mathscr{P})}(\chi).$$

Thus, scaling the screening weight by a factor of two with each successive depth ensures that the sum of energies is unchanged (up to multiplication by a constant) so the minimizer remains the same.

4.4 Boundary Conditions

In order to define the linear system, it is necessary to define the behavior of the function space along the boundary of the integration domain. In the original Poisson reconstruction the authors imposed Dirichlet boundary conditions, forcing the implicit function to have a value of $\frac{-1}{2}$ along the boundary. In the present work we extend the implementation to support Neumann boundary conditions as well, forcing the normal derivative to be zero along the boundary.

In principle these two boundary conditions are equivalent for watertight surfaces, since the indicator function has a constant negative value outside the model. However, in the presence of missing data we find Neumann constraints to be less restrictive because they only require that the implicit function have zero derivative across the boundary of the integration domain, a property that is compatible with the gradient constraint since the guiding vector field $\vec{V}$ is set to zero away from the samples. (Note that when the surface does cross the boundary of the domain, the Neumann boundary constraints create a bias to crossing the domain boundary orthogonally.)

Figure 3 shows the practical implications of this choice when reconstructing the Angel model, which was only scanned from the front. The left image shows the original point set and the reconstructions using Dirichlet and Neumann boundary conditions are shown to the right. As the figure shows, imposing Dirichlet constraints creates a water-tight surface that closes off before reaching the boundary while using Neumann constraints allows the surface to extend out to the boundary of the domain.

[2] For the Laplacian term, the Laplacian scales by a factor of 4 with refinement, and volumetric integrals scale by a factor of 1/8. For the screening term, area integrals scale by a factor of 1/4.

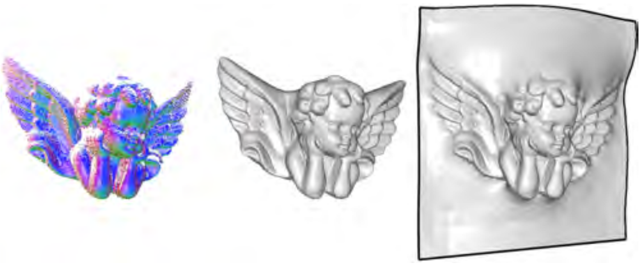


Fig. 3: Reconstructions of the Angel point set[‡] (left) using Dirichlet (center) and Neumann (right) boundary conditions.

Similar results can be seen at the bases of the models in Figures 1 and 4a, with the original Poisson reconstructions obtained using Dirichlet constraints and the screened reconstructions obtained using Neumann constraints.

5. IMPROVED ALGORITHMIC COMPLEXITY

In this section we discuss the efficiency of our reconstruction algorithm. We begin by analyzing the complexity of the algorithm described above. Then, we present two algorithmic improvements. The first describes how hierarchical clustering can be used to reduce the screening overhead at coarser resolutions. The second applies to both the unscreened and screened solver implementations, showing that the asymptotic time complexity in both cases can be reduced to be *linear* in the number of input points.

5.1 Efficiency of basic solver

Let us begin by analyzing the computational complexity of the unscreened and screened solvers. We assume that the points $\mathscr{P}$ are evenly distributed over a surface, so that the depth of the adapted octree is $D = O(\log|\mathscr{P}|)$ and the number of octree nodes at depth d is $O(4^d)$.

We also note that the number of nonzero entries in matrix $A^{dd'}$ is $O(4^d)$, since the matrix has $O(4^d)$ rows and each row has at most 5^3 nonzero entries. (Since we use second-order B-splines, basis functions are supported within their one-ring neighborhoods and the support of two functions will overlap only if one is within the two-ring neighborhood of the other.)

Assuming that the matrices $A^{dd'}$ have already been computed, the computational complexity for the different steps in Algorithm 1 is:

Step 3: $O(4^d)$ – since $A^{dd'}$ has $O(4^d)$ nonzero entries.

Step 4: $O(4^d)$ – since A^d has $O(4^d)$ nonzero entries and the number of relaxation steps performed is constant.

Steps 2-3: $\sum_{d'=0}^{d-1} O(4^d) = O(4^d \cdot d)$.

Steps 2-4: $O(4^d \cdot d + 4^d) = O(4^d \cdot d)$.

Steps 1-4: $\sum_{d=0}^{D} O(4^d \cdot d) = O(4^D \cdot D) = O(|\mathscr{P}| \cdot \log|\mathscr{P}|)$.

There still remains the computation of matrices $A^{dd'}$.

For the unscreened solver, the complexity of computing $A^{dd'}$ is $O(4^d)$, since each entry can be computed in constant time. Thus, the overall time complexity remains $O(|\mathscr{P}| \cdot \log|\mathscr{P}|)$.

For the screened solver, the complexity of computing $A^{dd'}$ is $O(|\mathscr{P}|)$ since defining the coefficients requires accumulating the screening contribution from each of the points, and each point contributes to a constant number of rows. Thus, the overall time complexity is dominated by the cost of evaluating the coefficients of $A^{dd'}$ which is:

$$\sum_{d=0}^{D}\sum_{d'=0}^{d-1} O(|\mathscr{P}|) = O(|\mathscr{P}| \cdot D^2) = O(|\mathscr{P}| \cdot \log^2|\mathscr{P}|).$$

5.2 Hierarchical Clustering of Point Constraints

Our first modification is based on the observation that since the basis functions at coarser resolutions are smooth, it is unnecessary to constrain them at the precise sample locations. Instead, we cluster the weighted points as in [Rusinkiewicz and Levoy 2000]. Specifically, for each depth d, we define $(w^d, \mathscr{P}^d)$ where $p_i \in \mathscr{P}^d$ is the weighted average position of the points falling into octree node i at depth d, and $w^d(p_i)$ is the sum of the associated weights.[3] If all input points have weight $w(p) = 1$, then $w^d(p_i)$ is simply the number of points falling into node i.

This alters the computation of the system matrix coefficients:

$$A_{ij}^{dd'} = \langle \nabla B_i^d, \nabla B_j^{d'} \rangle_{[0,1]^3} + 2^d \alpha \langle B_i^d, B_j^{d'} \rangle_{(w^d, \mathscr{P}^d)}.$$

Note that since $d > d'$, the value $\langle B_i^d, B_j^{d'} \rangle_{(w^d, \mathscr{P}^d)}$ is obtained by summing over points stored with the finer resolution.

In particular, the complexity of computing $A^{dd'}$ for the screened solver becomes $O(|\mathscr{P}^d|) = O(4^d)$, which is the same as that of the unscreened solver, and both implementations now have an overall time complexity of $O(|\mathscr{P}| \cdot \log|\mathscr{P}|)$.

On typical examples, hierarchical clustering reduces execution time by a factor of almost two, and the reconstructed surface is visually indistinguishable.

5.3 Conforming Octrees

To account for the adaptivity of the octree, Algorithm 1 subtracts off the constraints met at all coarser resolutions before relaxing at a given depth (steps 2-3), resulting in an algorithm with log-linear time complexity. We obtain an implementation with linear complexity by forcing the octree to be conforming. Specifically, we define two octree cells to be *mutually visible* if the supports of their associated B-splines overlap, and we require that if a cell at depth d is in the octree, then all visible cells at depth $d-1$ must also be in the tree. Making the tree conforming requires the addition of new nodes at coarser depths, but this still results in $O(4^d)$ nodes at depth d.

While the conforming octree does not satisfy the condition that a coarser solution can be prolonged into a finer one, it has the property that the solution obtained at depths $\{0,\ldots,d-1\}$ that is visible to a node at depth d can be expressed entirely in terms of the coefficients at depth $d-1$. Using an accumulation vector to store the visible part of the solution, we obtain the linear-time implementation in Algorithm 2.

[3] Note that the weight $w^d(p)$ is unrelated to the screening weight 2^d introduced in Section 4.3 for scale-independent screening.

Here, P_{d-1}^{d} is the B-spline prolongation operator, expressing a solution at depth $d-1$ in terms of coefficients at depth d. The number of nonzero entries in P_{d-1}^{d} is $O(4^d)$, since each column has at most 4^3 nonzero entries, so steps 2-5 of Algorithm 2 all have complexity $O(4^d)$. Thus, the overall complexity of both the unscreened and screened solvers becomes $O(|\mathscr{P}|)$.

Algorithm 2: Conforming Cascadic Poisson Solver

```
1  For $d \in \{0,\dots,D\}$                                  Iterate from coarse to fine.
2      $\hat{x}^{d-1} = P_{d-2}^{d-1}\hat{x}^{d-2}$             Upsample coarser accumulation vector.
3      $\hat{x}^{d-1} = \hat{x}^{d-1} + x^{d-1}$                Add in coarser solution.
4      $b^d = b^d - A^{d\,d-1}\hat{x}^{d-1}$                    Remove constraints met at coarser depths.
5      Relax $A^d x^d = b^d$                                    Adjust the system at depth d.
```

5.4 Implementation Details

The algorithm is implemented in C++, using OpenMP for multi-threaded parallelization. We use a conjugate-gradient solver to relax the system at each multigrid level. With the exception of the octree construction, most of the operations involved in the Poisson reconstruction can be categorized as operations that either "accumulate" or "distribute" information [Bolitho et al. 2007, 2009]. The former do not introduce write-on-write conflicts and are trivial to parallelize. The latter only involve linear operations, and are parallelized using a standard map-reduce approach: in the map phase we create a duplicate copy of the data for each thread to distribute values into, and in the reduce phase we merge the copies by taking their sum.

6. RESULTS

We evaluate the algorithm (*Screened*) by comparing its accuracy and computational efficiency with several prior methods: the original Poisson reconstruction of Kazhdan et al. [2006] (*Poisson*), the Wavelet reconstruction of Manson et al. [2008] (*Wavelet*), and the Smooth Signed Distance reconstruction of Calakli and Taubin [2011] (*SSD*).

For the new algorithm, we set the screening weight to $\alpha = 4$ and use Neumann boundary conditions in all experiments. (Numerical results obtained using Dirichlet boundaries were indistinguishable.) For the prior methods, we set algorithmic parameters to values recommended by the authors, using Haar Wavelets in the *Wavelet* reconstruction and setting the value/normal/Hessian weights to $1/1/0.25$ in the *SSD* reconstruction. For *Poisson*, *SSD*, and *Screened* we set the "samples-per-node" parameter to 1 and the "bounding-box-scale" parameter to 1.1. (For *Wavelet* the bounding box scale is hard-coded at 1 and there is no parameter to adjust the sampling density.)

6.1 Accuracy

We run three different types of experiments.

Real Scanner Data. To evaluate the accuracy of the different reconstruction algorithms on real-world data, we gathered several scanned datasets: the Awakening (10M points), the Stanford Bunny (0.2M points), the David (11M points), the Lucy (1.0M points), and the Neptune (2.4M points). For each dataset, we randomly partitioned the points into two equal-sized subsets: input points for the reconstruction algorithms, and validation points to measure point-to-reconstruction distances.

Figure 4a shows reconstructions results for the Neptune and David models at depth 10. It also shows surface cross-sections overlaid with the validation points in their vicinity. These images reveal that the Poisson reconstruction (far left), and to a lesser extent the SSD reconstruction (center left), over-smooth the data, while the Wavelet reconstruction (center left) has apparent derivative discontinuities. In contrast, our screened Poisson approach (far right) provides a reconstruction that faithfully fits the samples without introducing noise.

Figure 4b shows quantitative results across all datasets, in the form of RMS errors, measured using the distances from the validation points to the reconstructed surface. (We also computed the maximum error, but found that its sensitivity to individual outlier points made it an unreliable and unindicative statistic.) As the figure indicates, the Screened Poisson reconstruction (blue) is always more accurate than both the original Poisson reconstruction algorithm (red) and the Wavelet reconstruction (purple), and generates reconstruction whose RMS errors are comparable to or smaller than those of the SSD reconstruction (green).

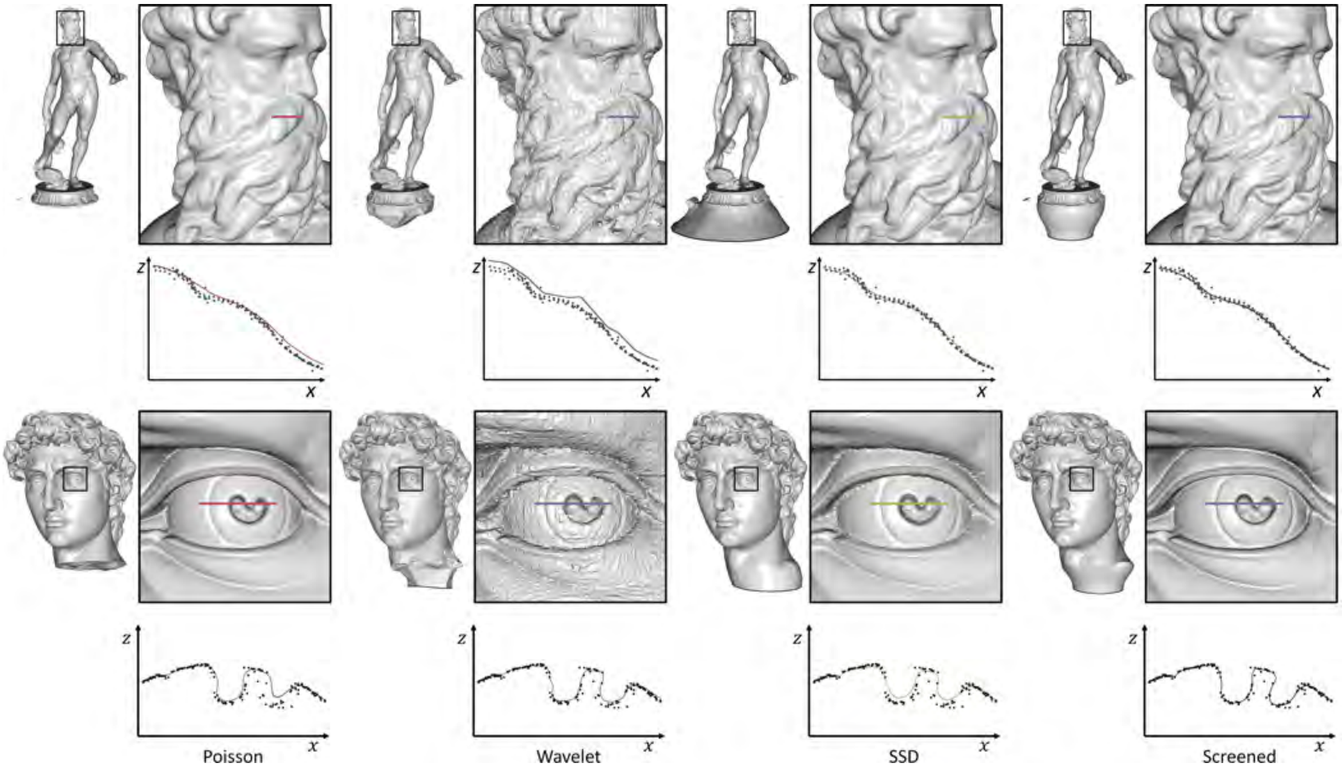
Clean Uniformly Sampled Data. To evaluate reconstruction accuracy on clean data, we used the approach of Osada et al. [2001] to generate oriented point sets by uniformly sampling the surfaces of the Fandisk, Armadillo Man, Dragon, and Raptor models. For each model, we generated datasets of 100K and 1M points and reconstructed surfaces from each point set using the four different reconstruction algorithms.

As an example, Figure 5a shows the reconstructions of the fandisk and raptor models using 1M point samples at depth 10. Despite the lack of noise in the input data, the Wavelet reconstruction has spurious high-frequency detail. Focusing on the sharp edges in the model, we also observe that the screened Poisson reconstruction introduces less smoothing, providing a reconstruction that is truer to the original data than either the original Poisson or the SSD reconstructions.

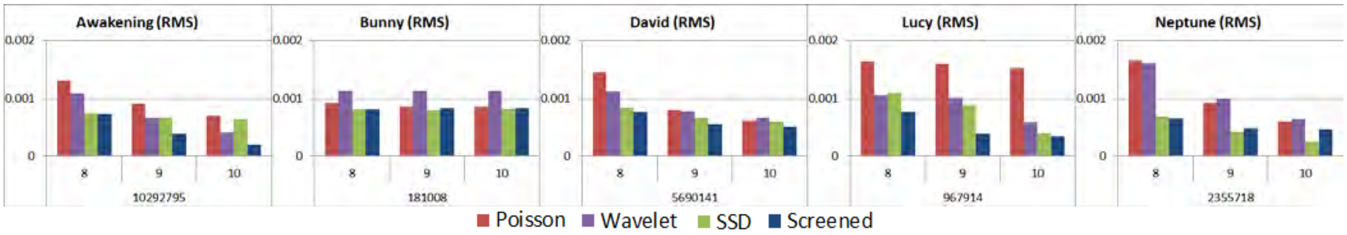
Figure 5b plots RMS errors across all models, measured bidirectionally between the original surface and the reconstructed surface using the Metro tool [Cignoni and Scopigno 1998]. As in the case of real scanner data, screened Poisson reconstruction always outperforms the original Poisson and Wavelet reconstructions, and is comparable to or better than the SSD reconstruction.

Reconstruction Benchmark. We use the benchmark of Berger et al. [2011] to evaluate the accuracy of the algorithms under different simulations of scanner error, including nonuniform sampling, noise, and misalignment. The dataset consists of multiple virtual scans of implicit surfaces representing the Anchor, Dancing Children, Daratech, Gargoyle, and Quasimodo models.

As an example, Figure 6a visualizes the error in the reconstructions of the anchor model from a virtual scan consisting of 210K points (demarked with a dashed rectangle in Figure 6b) at depth 9. The error is visualized using a red-green-blue scale, with red signifying



(a) Depth 10 reconstructions of the Neptune and David models[‡]. The 2D visualizations in the bottom row show cross-section curves from the reconstructions of the mustache and the iris, overlaid with adjacent validation points.



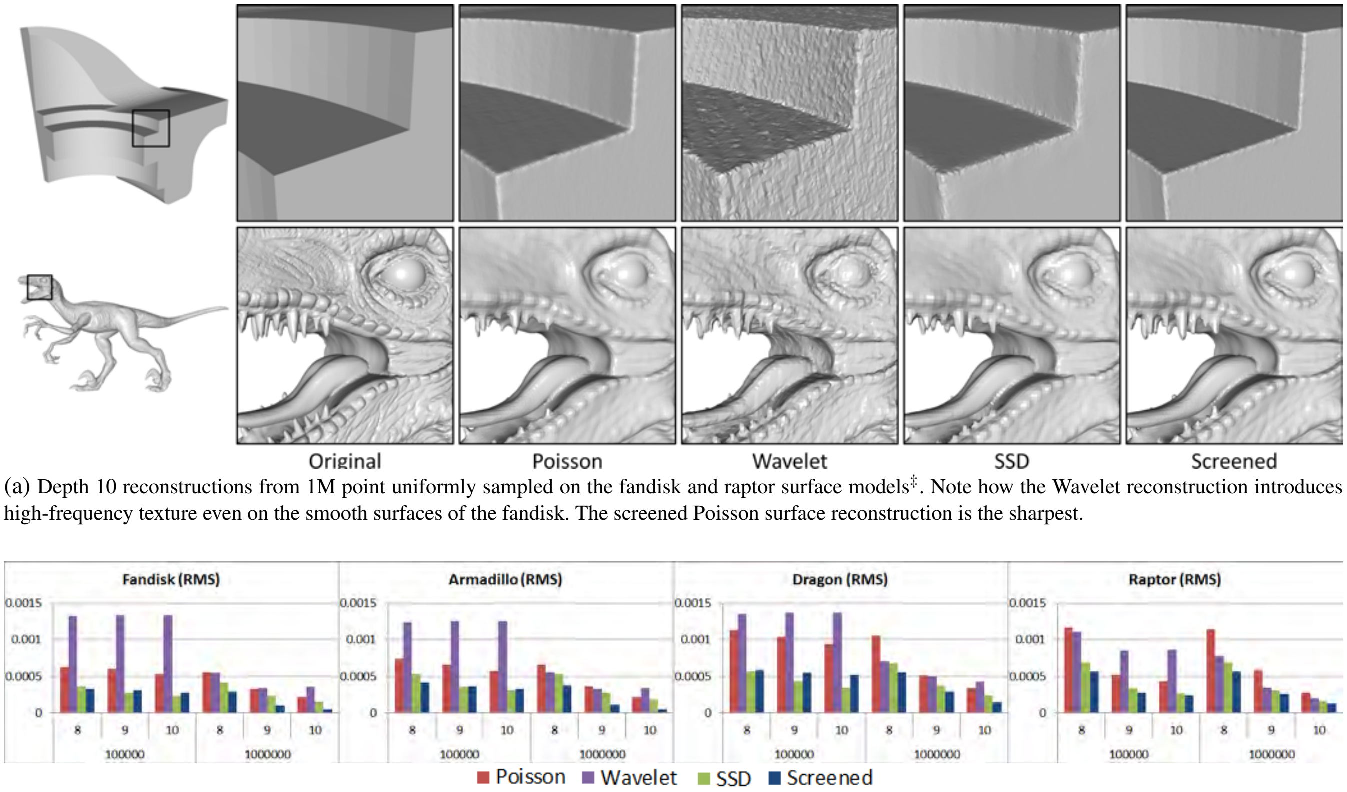
(b) For all models, plots of one-sided RMS errors, measured from the evaluation points to the reconstructed surface, as a function of the resolution depth (8, 9, and 10). In these experiments half the points were provided as input to the reconstruction algorithms, and the other half were used for evaluation.

Fig. 4: Experiments of reconstruction from real scanner data

low error and blue signifying high error. The top row shows the distance from the reconstructed surfaces to the ground-truth, and the bottom row shows the distance from the input samples to the reconstructed surfaces. Examining the top row, we note that only the SSD reconstruction produces a low error surface inside the cylindrical hole. Looking at the point sets in the bottom row, we observe that this region does not contain input samples, and therefore the success of the SSD reconstruction reflects its ability to better extrapolate the reconstruction into regions of missing data. However, if we only consider the fit of the reconstructions to the input point set (bottom row of Figure 6a) we find again that our screened Poisson reconstruction outperforms both the original Poisson reconstruction and the Wavelet reconstructions, and produces a surface whose fit to the input samples is comparable to that of the SSD reconstruction.

More generally, Figure 6b shows the results of these experiments with reconstructions computed at depth 9, giving the average distance from the reconstructions to the original implicit surface (top) and the average angle between the normals of points on the reconstruction and the normals of the corresponding points on the implicit surface (bottom), as a function of the number of point samples in the virtual scan. The results are visualized as scatter plots giving the ratio of the errors for the Wavelet, SSD, and screened Poisson reconstructions to the errors of the original Poisson reconstruction. Thus, values less than one indicate improved accuracy over Poisson reconstruction.

The results in Figure 6b therefore indicate that SSD reconstructs a surface that is closest to the original model. Visual analysis as in Figure 6a reveals that this improved accuracy is primarily due to the quality of the extrapolated surface in regions of missing data.



(a) Depth 10 reconstructions from 1M point uniformly sampled on the fandisk and raptor surface models‡. Note how the Wavelet reconstruction introduces high-frequency texture even on the smooth surfaces of the fandisk. The screened Poisson surface reconstruction is the sharpest.

(b) Accuracy of reconstruction techniques on "clean" point sets obtained by uniformly sampling 100K and 1M oriented points from various meshes. Reconstructions are obtained at depths 8, 9, and 10. Plots show RMS errors, measured in both directions between each reconstruction and original surface.

Fig. 5: Experiments of reconstruction from clean, uniformly sampled data

Thus, measuring distances with respect to the ground-truth model leads to relatively large errors in occluded surface regions, which naturally dominate the overall error statistics. This is corroborated by the angular errors plotted in the bottom row. Since these errors are formulated in terms of angles between normals, they emphasize the quality of the high-frequency component of the reconstruction. As such, they give lower weight in regions of missing data (where the reconstructions tend to be smooth) – so the comparable quality of the SSD and screened Poisson reconstructions indicate that they are of similar quality away from the missing data.

6.2 Computational Efficiency

Table I compares the (wall-clock) running time and memory usage of all four algorithms in reconstructing surfaces at depths 8, 9, 10, and 11 from the Neptune and David datasets. All experiments were run on a laptop with a quad-core Intel Core i7 and 8GB of RAM.

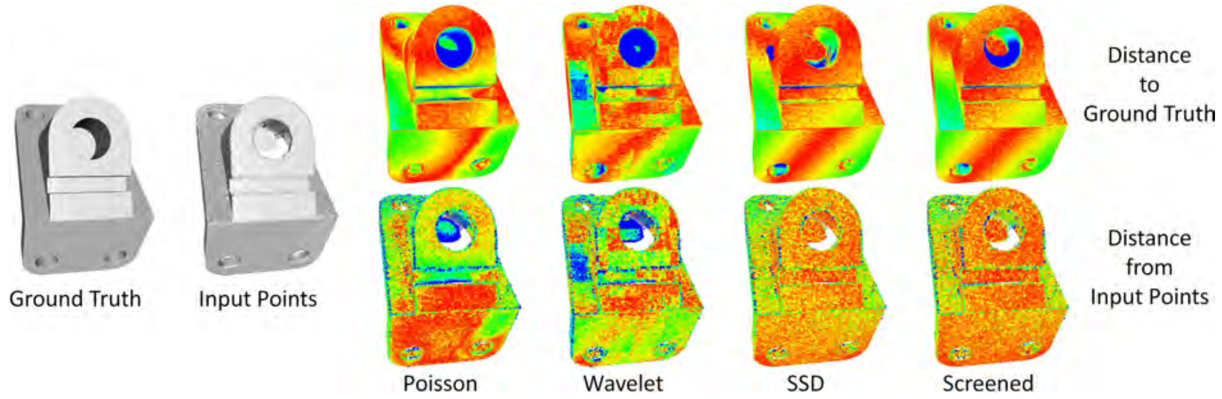
Running Time. The Wavelet reconstruction is fast; its use of compactly supported, orthogonal basis functions lets the reconstruction algorithm compute the implicit function coefficients through integration – never requiring the explicit solution of a linear system.

In contrast, the other three techniques use non-orthogonal basis functions, thus requiring a global system solution. For the original and screened Poisson reconstructions, the multigrid solver performs a constant number of conjugate-gradient iterations at each level, giving linear complexity in the number N of octree nodes. Thus, increasing the depth by one roughly quadruples the computation time.[4] In contrast, the SSD reconstruction uses conjugate-gradients to solve for all the coefficients simultaneously, which has a complexity of $O(N^{1.5})$, resulting in significantly slower performance at higher resolutions.

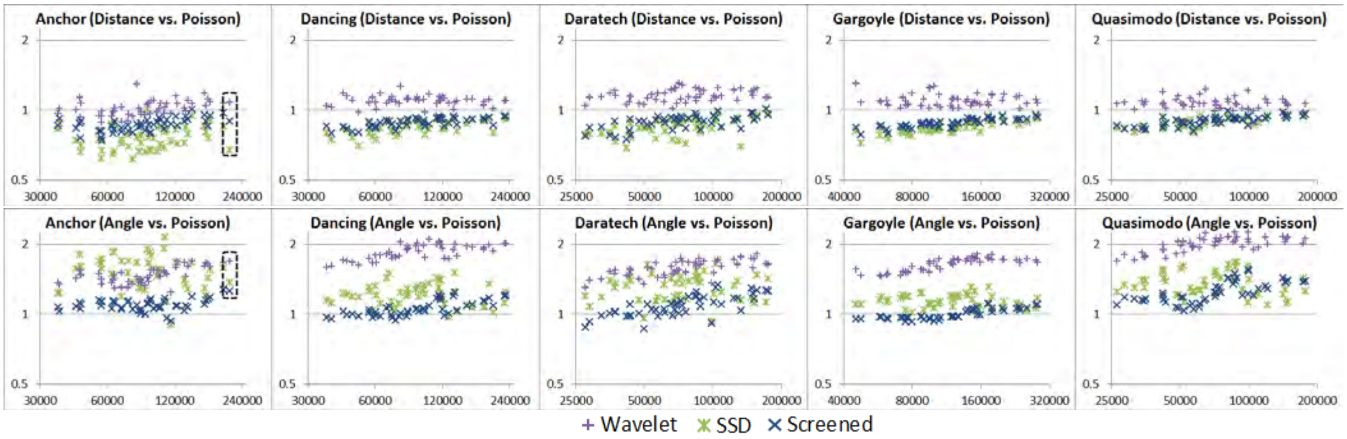
Interestingly, it might be possible to implement the SSD reconstruction technique using second-order B-splines as basis functions, just as in the Poisson reconstruction algorithms, and also benefit from a similar multigrid framework. However, we believe that this may not completely alleviate the performance bottleneck. In particular, the technique of Calakli and Taubin incorporates a Hessian smoothness term in the linear system. For the same discretization complexity, the condition number of the resulting system matrix is square that of the Laplacian matrix used in the Poisson reconstructions. Consequently, we expect the solver to require more iterations on large problem sizes.

Table I also highlights the speedup due to our algorithmic complexity improvements. When setting $\alpha = 0$, the new algorithm (shown in brackets) is 2–3 times faster than the original Poisson reconstruction algorithm. A small portion of this improvement (×1.1 to ×1.6) is due to multithreaded parallelization.

[4] Although the octree construction phase is log-linear, the constant tends to be small so this step does not dominate the running time in practice.



(a) Visualization of errors for reconstructions of the Anchor model‡. The top rows show the distance of the reconstructed surfaces from the ground-truth original surface, while the bottom row shows the distance from the input samples to the reconstructions. Errors are visualized using a red-green-blue colormap, with red corresponding to small error and blue to large.



(b) Reconstruction accuracy as measured by the benchmarks tests of Berger et al. [2011]. For each of the 5 datasets (columns), the two plots show the ratios of the mean distance (top) and mean normal (bottom) errors of the SSD, Wavelet, and Screened Poisson reconstructions, relative to the original Poisson reconstruction algorithm. Each symbol is one benchmark test, and the horizontal axis indicates the number of points in that test.

Fig. 6: Results using the reconstruction benchmark of Berger et al. [2011]

		Time in seconds				Memory in MB				Vertices $\times 10^6$			
Model	Depth	Poisson	Wavelet	SSD	Screened	Poisson	Wavelet	SSD	Screened	Poisson	Wavelet	SSD	Screened
Neptune	8	10 [13]	3	275	14	113	4	238	133	0.1	0.1	0.1	0.1
	9	25 [17]	4	547	20	149	11	455	269	0.2	0.2	0.2	0.2
	10	89 [36]	6	3302	44	422	35	1247	604	0.9	0.7	0.9	0.9
	11	320 [105]	9	15441	126	1387	118	3495	1622	3.1	1.5	2.9	3.2
David	8	41 [45]	9	492	48	427	11	863	454	0.2	0.2	0.2	0.2
	9	108 [66]	12	2355	73	510	38	1724	932	0.8	0.7	0.8	0.9
	10	412 [157]	20	19158	182	1498	151	4895	2194	3.4	2.8	3.3	3.5
	11	1710 [522]	43	†119119	609	5318	545	>8192	6188	12.8	7.3	11.6	13.3

Table I. : Runtime performance of the different reconstruction techniques on the David and Neptune datasets at depths 8, 9, 10, and 11. The numbers in brackets are timing results obtained using our method without screening ($\alpha = 0$) i.e., after the algorithmic improvements of Section 5.
†Since the memory usage of SSD exceeds the available RAM, we report the CPU user time for this experiment rather than the wall-clock time.

Even with the additional overhead of screening (which requires explicit evaluation of the piecewise polynomial basis functions at sub-voxel locations and cannot leverage the advantages of homogeneity used to accelerate the discretization of the Laplacian), our running times are faster than the original Poisson reconstruction.

Memory Usage. The memory usage of the different reconstruction algorithms also highlights the cost of formulating the reconstruction problem in terms of the solution to a linear system. Since the Poisson and SSD reconstructions define a linear system using the two-ring neighbors, the system matrix can have as many as 125 entries per row, resulting in a significant overhead for just storing the matrix. In contrast, the Wavelet reconstruction algorithm does not need to solve a linear system and avoids the associated memory overhead.

Reconstruction Resolution. The resolutions of the different reconstructions are shown in the last four columns of Table I. In practice, the resolution is determined by the depth of the octree, the fraction of the unit-cube occupied by the bounding box of the point set, the isosurfacing algorithm, and (for the Poisson, SSD, and screened Poisson reconstructions) the sample density threshold used in adaptive octree refinement. As the table shows, the similarly set parameters result in reconstructions of similar resolution for the original Poisson, SSD, and screened Poisson reconstructions.

6.3 Discussion and Limitations

Screening Parameter α. Empirically we have found that a value of $\alpha = 4$ works well for many types of datasets. We use this value for all examples in this paper. One nice property of the screened Poisson equation (3) is that with sufficient discretization, both its energy terms (gradient and screening) approach zero for the case of a uniform, noise-free sampling. Intuitively, the two terms are compatible for the case of clean data. In contrast, for the SSD reconstruction approach the smoothness energy only reaches zero for a planar surface. Because curved surfaces are penalized, the relative weighting of the different energy terms is important, and the smoothness parameter may need adjustment depending on the model geometry and the amount of noise. (See discussion below.)

Over-Fitting. Although the screening of the Poisson equation helps to more closely fit the input samples, there are circumstances under which this may lead to undesirable reconstructions. Specifically, when the input data contains significant amounts of misalignment or noise, the screened Poisson reconstruction will generate a surface that interpolates these artifacts. Figure 7 shows an example of this situation for the reconstruction of the Lucy model, with the results of the original Poisson reconstruction shown on the left and the results of the screened reconstruction shown on the right. Because the input scans from the right side of the face are misaligned (the points in the cross-section at the bottom cluster along two distinct curves), the screened reconstruction generates a pock-marked surface that undulates somewhat between the two scans.

This example also highlights a limitation of using the point-to-surface distance. As shown in Figure 4b, although the screened Poisson reconstruction has lower visual quality, it still provides a tighter fit to the data, resulting in RMS errors that are noticeably smaller than those of the unscreened reconstruction.

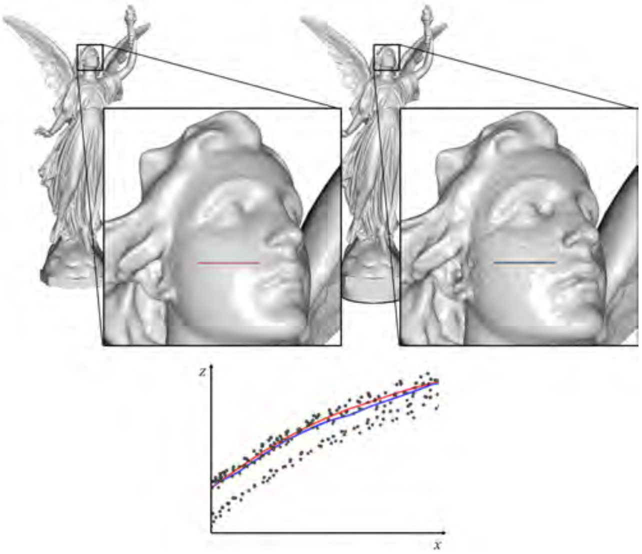


Fig. 7: Reconstruction of the Lucy model‡. When the input data has misaligned scans, the screened Poisson reconstruction (right) more tightly fits the noise, resulting in a qualitatively lower-quality reconstruction than the one returned by the original Poisson reconstruction (left).

Figure 8 shows an example with significantly noisy points, obtained using a multiview stereo algorithm.In this case, the screening term accentuates the noise in the reconstructed surface. Note however that the surface is stable, even over a wide range of α values. The figure also shows the effect of increasing the samples/node parameter from its default value of 1. The resulting coarsening of the octree structure helps to reduce reconstruction noise, but the simultaneous reduction in mesh resolution results in loss of detail. For this noisy example, the best strategy is to reduce the screening parameter α while keeping the samples/node parameter at 1. At an extreme setting $\alpha = 0$ we obtain an unscreened Poisson reconstruction as in [Kazhdan et al. 2006]. (The slight differences between our result with $\alpha = 0$ and the original Poisson reconstruction, particularly at the base of the Eagle's neck, derive from our use of a conforming octree. Because we introduce additional leaf nodes near regions of sparse sampling, we obtain a correspondingly refined triangulation at those locations.)

By comparison, when applying the SSD approach on this noisy data (Figure 9), it is difficult to find a smoothing parameter that simultaneously preserves detail and avoids the formation of spurious geometry (e.g., the ballooning at the tip of the beak).

Comparison with data interpolation. The recent scale-space meshing of Digne et al. [2011] is more akin to computational geometry approaches in that it directly interpolates a filtered subset of the original points.

Figure 10 compares our approach to such an interpolating reconstruction for a 1.6M point set obtained from a noise-free scan of the Tanagra statue [Digne et al. 2011]. As the figure shows, introducing screening improves the quality of the reconstruction, producing surfaces that more closely resemble the result of scale-space meshing (right). While interpolation is able to better reproduce sharp detail in the seal (bottom), it also results in a mesh with more high-frequency oscillation of the surface normal in the head (top).

7. CONCLUSION

Adding a dualized screening term to the Poisson surface reconstruction framework significantly improves its geometric fidelity, while still allowing an efficient multigrid solver. The reconstruction

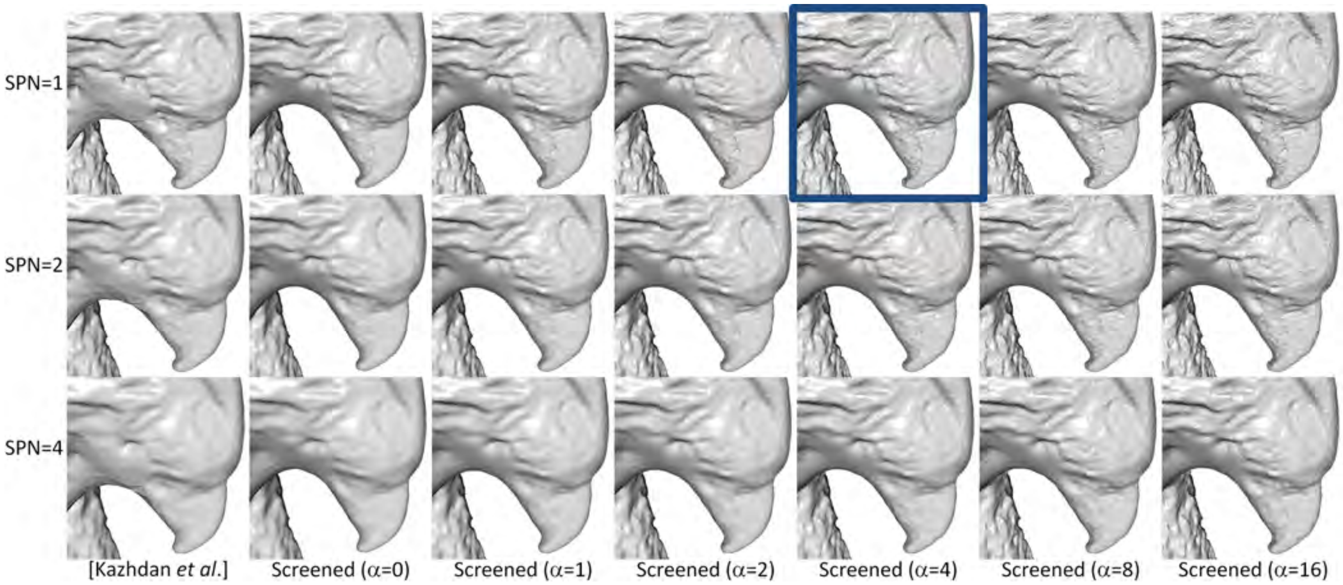


Fig. 8: Reconstruction of the noisy Eagle dataset‡ using the original scheme [Kazhdan et al. 2006] and our new algorithm, with various settings of the screening parameter (α) and samples per node (SPN). (The highlighted results are those with the default parameters used for evaluation in Section 6.1.)

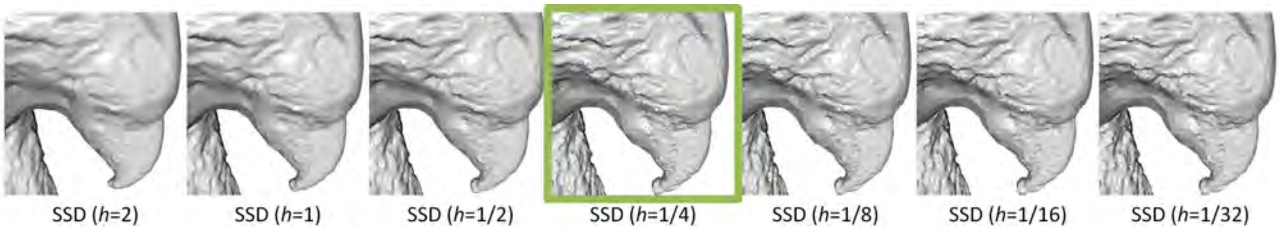


Fig. 9: Reconstruction of the noisy Eagle dataset‡ using the Smoothed Signed Distance reconstruction [Calakli and Taubin 2011] using various settings of the Hessian weighting parameter (h). (The highlighted results are those with the default parameters used for evaluation in Section 6.1.)

accuracy is comparable to or exceeds that of the Smooth Signed Distance reconstruction algorithm, yet the processing times are significantly faster. Hierarchical clustering of the points and a conforming octree structure enables a multigrid algorithm with linear complexity on the number of input points.

Experiments show that the SSD algorithm is able to better extrapolate the surface into regions of missing data. An interesting area for future work would be to develop a scheme that incorporates both the Laplacian and bilaplacian operators, such that the bilaplacian (which has higher cost in terms of convergence rate) is only activated in the challenging, under-constrained regions of the domain.

Some other avenues for future work include:

—Generalization of the multigrid solver to support full V- and W-cycles.

—Parallelization of the algorithm for GPU computation.

—Reconstruction of surfaces with boundaries.

—Extension of the algorithm to incorporate additional knowledge of the scanned surface, such as range maps and silhouettes.

—Adaptive refinement of the octree based on residuals measured at coarser levels, to allow the output mesh complexity to adapt not only to sampling density but also to solution quality.

APPENDIX

A. DUALIZED SCREENING

Following the Euler-Lagrange formulation, the function χ minimizes Equation 4 if, for any function $D : [0,1]^3 \to \mathbb{R}$,

$$\begin{aligned} 0 &= \lim_{\varepsilon \to 0} \frac{E(\chi + \varepsilon D) - E(\chi)}{\varepsilon} \\ \Leftrightarrow \quad 0 &= \langle \nabla \cdot \vec{V} - \Delta\chi, D \rangle_{[0,1]^3} + \alpha \langle \chi, D \rangle_{(w,\mathscr{P})}. \end{aligned}$$

Thus, setting $\tilde{I}$ to be the operator defined by

$$\langle \tilde{I}(F), G \rangle_{[0,1]^3} = \langle F, G \rangle_{(w,\mathscr{P})} \quad \forall G,$$

it follows that χ is a minimizer of the energy if and only if

$$\langle (\Delta - \alpha\tilde{I})\chi, D \rangle_{[0,1]^3} = \langle \nabla \cdot \vec{V}, D \rangle_{[0,1]^3}, \quad \forall D,$$

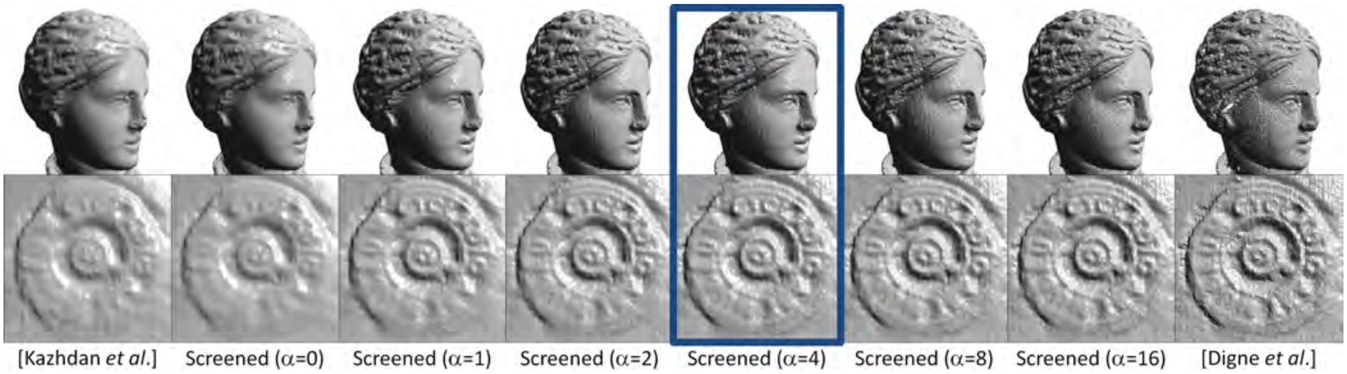


Fig. 10: Comparison of Poisson reconstruction with the interpolation technique of Digne et al. [2011] for the head (top) and seal (bottom) of the Tanagra model‡. (The highlighted results are those with the default parameters used for evaluation in Section 6.1.)

i.e., iff χ satisfies the equation

$$(\Delta - \alpha\tilde{I})\chi = \nabla\cdot\vec{V}. \qquad (6)$$

One can interpret Equation 6 as a (dually) screened Poisson equation. Specifically, the inner-products $\langle\cdot,\cdot\rangle_{[0,1]^3}$ and $\langle\cdot,\cdot\rangle_{(w,\mathscr{P})}$ define maps $\mathscr{I}_{[0,1]^3}$ and $\mathscr{I}_{(w,\mathscr{P})}$ from the space of functions into its dual, with:

$$\left(\mathscr{I}_{[0,1]^3}(f)\right)(g) = \langle f,g\rangle_{[0,1]^3} \qquad \left(\mathscr{I}_{(w,\mathscr{P})}(f)\right)(g) = \langle f,g\rangle_{(w,\mathscr{P})}$$

for all functions f and g.

Using this notation, the operator $\tilde{I}$ is the composition $\tilde{I} = \mathscr{I}^{-1}_{[0,1]^3}\circ\mathscr{I}_{(w,\mathscr{P})}$. In traditional filtering applications the two inner-products are the same, so $\tilde{I}$ is the identity and we get a screened Poisson equation. In our case the gradient and interpolation constraints are defined over different domains and we require the operator $\tilde{I}$ to transition between the two.

For infinite-dimensional spaces, it is not always the case that $\tilde{I}$ is well defined. However, for finite-dimensional spaces, one can always define such an operator by choosing a basis and representing $\tilde{I}$ by $M^{-1}_{[0,1]^3}\circ M_{(w,\mathscr{P})}$, where $M_{[0,1]^3}$ and $M_{(w,\mathscr{P})}$ are the mass matrices for the chosen basis, defined with respect to the two inner-products.[5] (Note that the above expression for $\tilde{I}$ only requires that the mass-matrix $M_{[0,1]^3}$ be invertible, so the operator is well-defined even when there are basis functions whose support does not intersect $\mathscr{P}$.)

‡Acknowledgements

We thank the EPFL Computer Graphics and Geometry Laboratory for the Eagle model, the Stanford 3D Scanning Repository for the Lucy and David models, Aim@Shape for the Neptune, Raptor, fandisk, and cow models, Calakli and Taubin for the Angel model, Berger et al. for the Anchor model, and Digne et al. for the Tanagra model. We also thank Fatih Calakli and Gabriel Taubin for publishing their SSD implementation, Julie Digne and colleagues for sharing their reconstruction results, and Berger et al. for making available their surface reconstruction benchmark. This work was funded in part by the NSF Career Grant (#6801727).

REFERENCES

ALLIEZ, P., COHEN-STEINER, D., TONG, Y., AND DESBRUN, M. 2007. Voronoi-based variational reconstruction of unoriented point sets. *Symposium on Geometry processing*, 39–48.

AMENTA, N., CHOI, S., AND KOLLURI, R. 2001. Power crust. *ACM Symposium on Solid Modeling and Applications*, 249–260.

BAJAJ, C., BERNARDINI, F., AND XU, G. 1995. Automatic reconstruction of surfaces and scalar fields from 3D scans. *ACM SIGGRAPH*, 109–18.

BERGER, M., LEVINE, J., NONATO, L., TAUBIN, G., AND SILVA, C. 2011. An end-to-end framework for evaluating surface reconstruction. Technical Report UUSCI-2011-001, SCI Institute, University of Utah.

BHAT, P., CURLESS, B., COHEN, M., AND ZITNICK, L. 2008. Fourier analysis of the 2D screened Poisson equation for gradient domain problems. *European Conference on Computer Vision*, 114–128.

BOISSONNAT, J. AND OUDOT, S. 2005. Provably good sampling and meshing of surfaces. *Graphical Models 67,* 5, 405–451.

BOLITHO, M., KAZHDAN, M., BURNS, R., AND HOPPE, H. 2007. Multilevel streaming for out-of-core surface reconstruction. *Symposium on Geometry Processing*, 69–78.

BOLITHO, M., KAZHDAN, M., BURNS, R., AND HOPPE, H. 2009. Parallel Poisson surface reconstruction. *International Symposium on Visual Computing*.

CALAKLI, F. AND TAUBIN, G. 2011. SSD: Smooth signed distance surface reconstruction. *Computer Graphics Forum 30,* 7.

[5] To see that this is the case, we observe that if $\{B^*_1,\ldots,B^*_N\}$ is the basis for the dual space, with $B^*_i(B_j) = \delta_{ij}$ then the mass matrix $M_{[0,1]^3}$ (respectively $M_{(w,\mathscr{P})}$) is the matrix expression for the map $\mathscr{I}_{[0,1]^3}$ (respectively $\mathscr{I}_{(w,\mathscr{P})}$) with respect to the bases $\{B_1,\ldots,B_N\}$ and $\{B^*_1,\ldots,B^*_N\}$.

CARR, J., BEATSON, R., CHERRIE, H., MITCHELL, T., FRIGHT, W., MCCALLUM, B., AND EVANS, T. 2001. Reconstruction and representation of 3D objects with radial basis functions. *ACM SIGGRAPH*, 67–76.

CAZALS, F. AND GIESEN, J. 2006. Delaunay triangulation based surface reconstruction: Ideas and algorithms. In *Effective Computational Geometry for Curves and Surfaces*. Springer, 231–273.

CHUANG, M. AND KAZHDAN, M. 2011. Interactive and anisotropic geometry processing using the screened Poisson equation. *ACM Trans. Graphics 30,* 4.

CIGNONI, P. AND SCOPIGNO, R. 1998. Metro: Measuring error on simplified surfaces. *Computer Graphics Forum 17,* 2.

CURLESS, B. AND LEVOY, M. 1996. A volumetric method for building complex models from range images. *ACM SIGGRAPH*, 303–312.

DIGNE, J., AUDFRAY, N., LARTIGUE, C., MEHDI-SOUZANI, C., AND MOREL, J.-M. 2011. Farman Institute 3D Point Sets - High Precision 3D Data Sets. *Image Processing On Line*.

DIGNE, J., MOREL, J.-M., SOUZANI, C.-M., AND LARTIGUE, C. 2011. Scale space meshing of raw data point sets. *Computer Graphics Forum 30,* 6.

FLETCHER, C. 1984. *Computational Galerkin Methods*. Springer.

HOPPE, H., DEROSE, T., DUCHAMP, T., MCDONALD, J., AND STUETZLE, W. 1992. Surface reconstruction from unorganized points. *ACM SIGGRAPH*, 71–78.

HORNUNG, A. AND KOBBELT, L. 2006. Robust reconstruction of watertight 3D models from non-uniformly sampled point clouds without normal information. *Symposium on Geometry Processing*, 41–50.

KAZHDAN, M., BOLITHO, M., AND HOPPE, H. 2006. Poisson surface reconstruction. *Symposium on Geometry processing*, 61–70.

KOLLURI, R., SHEWCHUK, J., AND O'BRIEN, J. 2004. Spectral surface reconstruction from noise point clouds. *Symposium on Geometry Processing*, 11–21.

LABATUT, P., PONS, J.-P., AND KERIVEN, R. 2009. Robust and efficient surface reconstruction from range data. *Computer Graphics Forum 28,* 8, 2275–2290.

LEVOY, M., PULLI, K., CURLESS, B., RUSINKIEWICZ, S., KOLLER, D., PEREIRA, L., GINZTON, M., ANDERSON, S., DAVIS, J., GINSBERG, J., SHADE, J., AND FULK, D. 2000. The Digital Michelangelo Project: 3D scanning of large statues. *ACM SIGGRAPH*, 131–144.

MANSON, J., PETROVA, G., AND SCHAEFER, S. 2008. Streaming surface reconstruction using wavelets. *Symposium on Geometry Processing*, 1411–1420.

MULLEN, P., DE GOES, F., DESBRUN, M., COHEN-STEINER, D., AND ALLIEZ, P. 2010. Signing the unsigned: robust surface reconstruction from raw pointsets. *Symposium on Geometry Processing*, 1733–1741.

NAGAI, Y., OHTAKE, Y., AND SUZUKI, H. 2009. Smoothing of partition of unity implicit surfaces for noise robust surface reconstruction. *Computer Graphics Forum 28,* 5, 1339–1348.

NEHAB, D., RUSINKIEWICZ, S., DAVIS, J., AND RAMAMOORTHI, R. 2005. Efficiently combining positions and normals for precise 3D geometry. *ACM Trans. Graphics 24,* 3, 536–543.

OHTAKE, Y., BELYAEV, A., AND ALEXA, M. 2005. Sparse low-degree implicits with applications to high quality rendering, feature extraction, and smoothing. *Symposium on Geometry Processing*, 149–158.

OSADA, R., FUNKHOUSER, T., CHAZELLE, B., AND DOBKIN, D. 2001. Matching 3d models with shape distributions. In *Proceedings of the International Conference on Shape Modeling & Applications*. SMI '01. 154–166.

PODOLAK, J. AND RUSINKIEWICZ, S. 2005. Atomic volumes for mesh completion. *Symposium on Geometry Processing*.

RUSINKIEWICZ, S. AND LEVOY, M. 2000. Qsplat: a multiresolution point rendering system for large meshes. In *ACM SIGGRAPH*. 343–352.

Received May 2012; accepted December 2012