Poisson Surface Reconstruction on non-Cartesian Lattices

***Abstract*—In this work, we cast the well-known Poisson surface reconstruction algorithm into a more general setting, we reformulate it in terms of shift invariant spaces (spaces that are spanned by lattice translates of an admissible generating function). The treatment is general, but our specific interest is in reconstructing a surface of a solid model from an oriented point-set within function spaces defined over the body centred cubic lattice. We also propose a general framework for approximating the solution to Poisson?s equation in a hypercube with zero Dirichlet boundary conditions, and a new variational scheme to re-sample the initial oriented point-set onto a regular grid. Once the points have been re-sampled onto a grid, the rest of the pipeline is purely based on digital filtering. We also analyze the error incurred via the digital filtering approximation methodology and propose a Fourier domain error kernel that can be used to design solution filters that fully exploit the approximation capabilities of the target space. Finally, we show that a host of Poisson-like methods fail to take advantage of the full approximation spaces over which they are defined.**

I. Introduction

Reconstructing a solid model from a scattered set of points is a common and well studied problem in computer graphics. Interest in this problem is rooted in the desire to convert real life data-sets – acquired from range and now increasingly from commodity hardware like the Kinect – to 3D polygonal models. The idea also has a applications in model re-meshing and hole filling. [[XXX This needs more meat, it's a pretty weak ending to the intro paragraph.]]

Given an oriented point-set, a common approach is to find an implicit function whose iso-contour corresponds to an approximation of the original surface. In this work we take a "signal processing" approach to this problem – we cast the problem into a general shift-invariant setting. More explicitly, we reconstruct the desired implicit function within a function space spanned by lattice translates of a single generating function. The extension to this class of function spaces allows us to consider anchoring our basis functions at non-Cartesian lattice sites (Section III-B). Our approach is general and applicable to any valid lattice and generator, but our implementation is narrowed to applicable function spaces over the *Body Centred Cubic* (BCC) and *Cartesian* (CC) lattices.

The advantage of moving to the BCC lattice is two fold. First, it's well known that the BCC lattice generates the optimal sampling pattern in $\mathbb{R}^3$. The argument for optimality is simple; the sampling of a function with respect to the BCC lattice is equivalent to a periodization of it's frequency spectrum about the BCC's dual lattice, the *Face Centered Cubic* (FCC) lattice. Optimality follows from the observation that the FCC lattice is the optimal sphere packing lattice in three dimensions, that is, it packs frequency replica as tightly as possible in the Fourier domain, resulting in less *pre-aliasing* from sampling. For isotropically band-limited functions, this tighter packing of frequency content allows for about 30% more information to be captured when compared to samplings generated from a Cartesian lattice. The second advantage to moving to the BCC lattice is that there exist reconstruction filters on the BCC lattice that outperform commonly used filters of equivalent order on the CC lattice in terms of both speed and accuracy [1] (in software implementations) – these filters are also more compact than their typical CC counterparts. In the context of our surface reconstruction scheme, this gives rise to regularization matrices that are more sparse on the BCC lattice and tend to provide faster convergence when optimizing the initial point set (Section **??**.)

Our surface reconstruction methodology follows the general Poisson approach, but with a few twists. First, we construct a smoothed approximation to the gradient field of a model's indicator function. The divergence of the gradient field is estimated and subsequently fed into a Poisson solver that outputs the coefficients needed to approximate the smoothed indicator function in a target shift invariant space. This Poisson solver is tailored to cater to the approximation power provided by each space. However, our approach is novel in two notable ways. Firstly, we employ a variational scheme to obtain a lattice-based approximation of the gradient field. This scheme depends only on the generating kernel of the target space and optimizes a functional that incorporates interpolation and smoothness constraints. Secondly, we utilize the theory behind shift-invariant spaces to seek discretizations of the divergence and Laplacian operators that are tailored to exploit the full approximation capabilities of the target space.

Additionally, we consider the possibility of approximating the smoothed gradient field representation within function spaces that are spanned by shifted versions of a single generating kernel. In the context of gradient estimation, introducing a shift in the direction of the derivative has shown to improve overall gradient approximation fidelity in terms of gradient orientation and magnitude, as well as displaying the ability to capture higher frequency details that are smoothed out in non-shifted schemes [2]. Since the divergence operator is intimately connected to the gradient operator, these results are of interest to us. Using an appropriate discretization of the derivative/divergence operator is an important step in recovering the indicator function of the initial model. [[XXX Need to emphasize the error-kern here]]

To summarize, our contributions are as follows:

- We reformulate the Poisson surface reconstruction approach using general shift-invariant spaces as the target approximation space

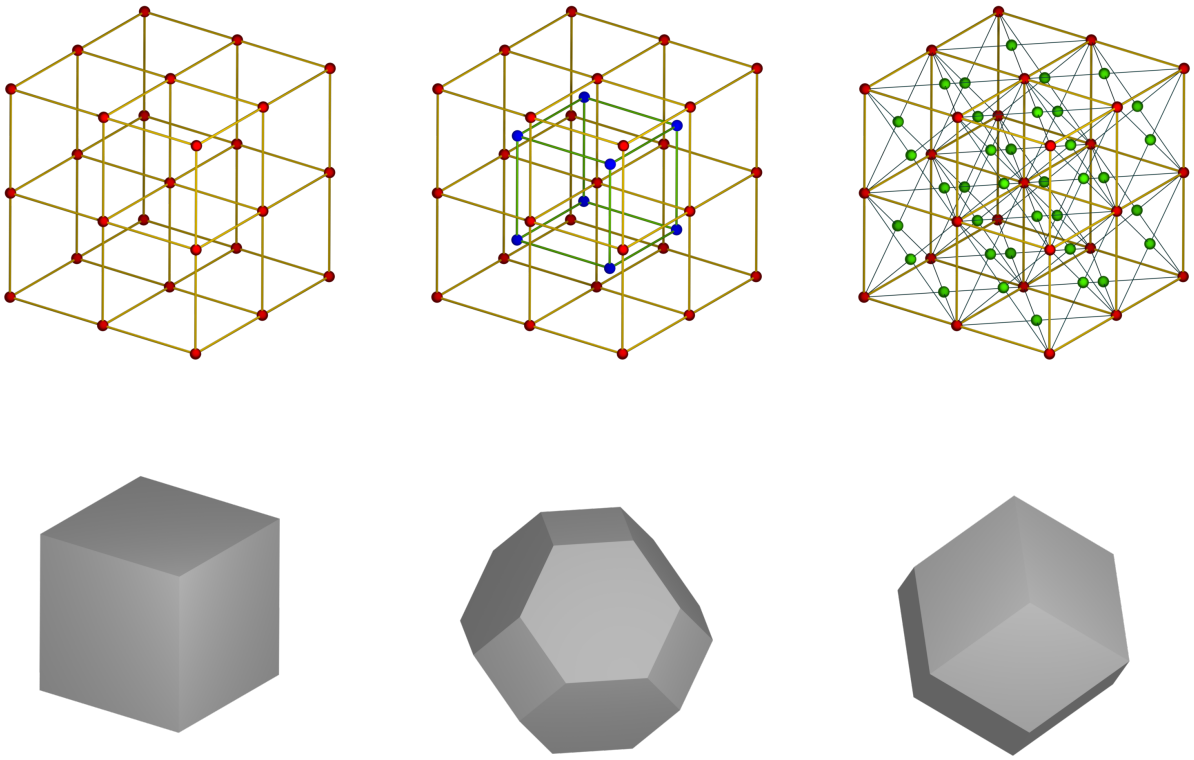

Fig. 1. Sampling lattices (top) with with their corresponding Voronoi regions (bottom). Left to right are the CC, BCC and FCC lattices respectively.

- Specifically, we investigate the reconstruction of surfaces in both a sub-optimal (Cartesian) and an optimal (BCC) box-spline function space
- We present a new variational resampling scheme that resamples the original scattered oriented points' normals onto any regular lattice. This resampling scheme depends on the generating kernel, but is general enough to be extended to shifted generating kernels as this has been shown to improve gradient estimation in shift invariant spaces.
- We present an extension of the error kernel of Blu and Unser to general linear operators and show how to design filters that respect the full approximation order of the target approximation space.
- While our focus is on comparing the surface reconstruction algorithm on both the CC and BCC lattices, we also provide qualitative and quantitative comparisons between the presented technique (on both the Cartesian and BCC lattices) and similar methods.

[[XXX OVERALL: What is lacking from this intro is the true purpose of this work (from the surface reconstruction aspect.) We want to show that the BCC beats the CC lattice. NOT that our scheme is better than Kazhdan's. We do show this, but we need to emphasize it above!]]

II. Related Work

Surface reconstruction is still a very active area of research in computer graphics and it's beyond the scope of this work to touch on all the material related to the subject. We therefore very briefly review some relevant contributions – those seeking a more in-depth review may look to a recent survey of the field [**?**]. We focus mainly on some of the more related and recent developments.

Many surface reconstruction algorithms are so-called "*implicit*" reconstruction algorithms. They attempt to reconstruct an implicit function, $f : \mathbb{R}^3 \mapsto \mathbb{R}$, whose level-set, $f(\mathbf{x}) = s$ (for some iso-value $s \in \mathbb{R}$), represents an approximation to the original surface. Other non-implicit techniques generally infer meshes through some other combinatorial means, or a set predefined logic or rules.

For example, there exist many Delaunay style algorithms in which a triangulation is constructed using a subset of the original point-set [4]. The Power Crust [5] and Cocone [6] algorithms are two other well-known techniques that do not construct implicit functional representations. These techniques often aim to exactly interpolate the input point set, and in the presence of noise tend to over-fit the surface. When the input data are noisy implicit approaches often perform better than non-implicit algorithms, as implicit algorithms inherently

attempt to fit locally smooth basis functions to the data, averaging out any noise. Reconstructing either a distance function or an indicator function is common to these approaches.

One of the most well known implicit algorithms is the algorithm of by Hoppe et al. [7]. In this work a set of tangent planes is constructed to create an approximation to a surface's distance function. A more recent work, is the Smoothed Signed Distance Field reconstruction technique [8], in which an oriented point set imposes constraints on the reconstructed function, its gradient, and its Hessian. These constraints are equvalent to requiring that the reconstructed function be an approximation to the distance field of the original model. Other techniques attempt to reconstruct a signed distance function within a function space spanned by radial basis functions [9]. Recently, there has also been research that incorporates point normals into the radial basis reconstruction framework [10]. It's also possible to construct an implicit function as a normalize sum of compactly supported basis functions defined over an oct-tree, essentially "smearing" the point-set around space and constructing an indicator "shell" around them [**?**].

Another approach is to find an indicator function that approximates the "inside-outside" function of original model. An early technique [11] transforms the smoothed gradient to the Fourier domain, applies an appropriate operator, then reconstructs it in the spatial domain. This approach often deals well with incomplete, missing, or otherwise noisy data and has the guarantee of creating a water tight surface. However, it involves reconstructing the indicator on a regular grid, which becomes very memory intensive as the grid is refined.

Poisson surface reconstruction techniques make the observation that the Fourier method is equivalently stated as a spatial Poisson problem [12], [13]. Moreover, they seek to rectify the memory limitations of that work, while still maintaining its benefits. However, current techniques based on this idea have been shown to over-smooth the initial point-set, and therefore the resulting surface. [14].

It is also possible to reconstruct the indicator of a point set within a wavelet basis [15]. In this work, an approximate indicator function of the model is projected onto a compact wavelet basis. This leads to a simple and efficient algorithm even when the data sets are massive. However the method's speed comes at a price, as the resulting surfaces are often non-smooth, and display many "jagged" artifacts (although a form of smoothing is applied to the resulting indicator function in an attempt to rectify this.)

Outside of the context of surface reconstruction there has been considerable evidence that shows non-Cartesian approaches to volume rendering [**?**], [1], [20] and fluid flow [**?**] outperform Cartesian approaches in both speed and memory consumption. Additionally, finite difference Laplacian stencils on non-Cartesian grids lead to reduced numerical dispersion compared to their Cartesian counterparts (see e.g. [**?**], [**?**] and references therein). It is therefore natural to question whether these advantages can be exploited within the context of surface reconstruction.

Our approach aims to solve the Poisson surface reconstruction problem in the setting of general shift-invariant spaces. A crucial step in this regard is the accurate estimation of the divergence of the gradient field from the oriented point-set. Radial basis functions (RBFs) are generally used in the context of the reconstruction of scattered data, however, RBF techniques are often computationally expensive, requiring the solution of large unstable systems of equations (for exact interpolation.) A proposed solution to this problem is to "*re-sample*" the input data onto a regular shift-invariant space. This allows one to use efficient compact reconstruction kernels and data processing techniques [16], [17]. Recently, an extension of this idea to box spline spaces [18] proposes a variational reconstruction framework in which the BCC lattice consistently outperforms the Cartesian lattice. Our method takes a similar variational approach and seeks to construct a smoothed lattice-based approximation of the gradient of the indicator function (Section IV-A). Our employed constraints force the gradient to be close to zero away from the surface, while maintaining a certain degree of smoothness in the resulting approximation. This is preferable in comparison to other works [11]–[13] in which each sample is trilinearly distributed about either a grid or an oct-tree respectively, and is known to over-smooth data.

Having obtained a high-quality lattice-based approximation of the gradient field, we invoke the theory behind shift-invariant spaces and accurately estimate the divergence by applying derivative filters that are designed to harness the full approximation capabilities of the target space (Section IV-B).

Once, the smoothed gradient's divergence has been calculated, recovering the indicator function is a simple matter of taking the inverse of the Laplacian over domain (i.e. solving the Poisson system). To do this, we take an approach that is also inspired by filtering methodologies in signal processing. In particular, we're interested in an approximate solution to Poisson?s equation that lies in a chosen shift-invariant space. Our solution methodology is reminiscent of the finite element method [**?**]. However, we do not impose any boundary conditions on the generating kernels. Rather, we satisfy homogeneous Dirichlet boundary conditions implicitly by requiring that the coefficient sequence be odd (Section III). The solution is obtained by applying a digital filter to point samples. We approach the problem from first principles and derive the tools necessary to design solution filters for a chosen shift-invariant space (Section **??**). [[XXX I still need to fix-up section refs/refs here.]]

III. Preliminaries

A. Poisson Surface Reconstruction

We denote the indicator function of a model M as

$$\chi_M(\mathbf{x}) := \begin{cases} 1, & \text{if } \mathbf{x} \text{ is in the interior of the model } M \\ \frac{1}{2}, & \text{if } \mathbf{x} \text{ is on the model's surface} \\ 0, & \text{if } \mathbf{x} \text{ is exterior to } M \end{cases}.$$

Integral to formulating surface reconstruction as a Poisson problem, is the observation that if one obtains some function $\vec{V}(\mathbf{x}) \approx \nabla\chi_M$ that approximates the smoothed gradient of the indicator function, finding an approximation to the indicator function reduces to finding a $\tilde{\chi}$ such that

$$\Delta\tilde{\chi} = \nabla \cdot \vec{V}. \tag{1}$$

We define the set of input surface points to be $P := \{(\mathbf{p}_1, \mathbf{n}_1), (\mathbf{p}_2, \mathbf{n}_2) \dots (\mathbf{p}_M, \mathbf{n}_M) : (\mathbf{p}_i, \mathbf{n}_i) \in \mathbb{R}^3 \times \mathbb{R}^3\}$ where $\mathbf{p}_i$ and $\mathbf{n}_i$ are the position and normal vector of the i^{th} sample respectively. Orientation can be succinctly represented with only two values, however, it is convenient to keep it denoted as the normal at a point. This is because the length of the normal can be used to encode the sampling density of the point-set at a point. For the following discussion, we also need to introduce the notation $\mathcal{P}_\mathbf{s}$, which denotes a patch of the surface model about the sample location $\mathbf{s}$.

Inspired by the approach of Kazhdan et al. [12] the approximation $\vec{V}(\mathbf{x})$ is defined to be

$$\begin{aligned} \vec{V}(\mathbf{x}) &:= \sum_{(\mathbf{s},\mathbf{n})\in P} |\mathcal{P}_s|\, F(\mathbf{x}-\mathbf{s})\cdot\mathbf{n} \\ &\approx \sum_{(\mathbf{s},\mathbf{n})\in P} \int_{\mathcal{P}_s} F(\mathbf{x}-\mathbf{p})\nabla\chi(\mathbf{p})\mathrm{d}\mathbf{p}. \end{aligned} \tag{2}$$

Thus, $\vec{V}(\mathbf{x})$ approximates the gradient of the indicator smoothed by some filter F, broken up over surface patches of the model. We assume the point sampling to be uniform, thus the area of each patch $|\mathcal{P}_s|$ should be a constant, and our resulting approximation will be scaled by an unknown factor. We return to this in Section **??**. The choice of F places computational constraints on the resulting algorithm.

F is often explicitly chosen to be a compactly supported function. This simplifies and reduces the computational cost of the method when used in conjunction with an oct-tree based representation of $\vec{V}(\mathbf{x})$ [12]. However, there is a disadvantage in that F may over smooth data in areas of high curvature [14], much in the same way that a representation of a sampled signal may fail to take advantage of the full approximation power of the function space in which it is represented if it is not properly prefiltered. Theoretically, one could reconstruct the surface at higher resolutions, allowing the basis functions to become more "fine", but increasing the total amount of memory used. However, this is not realistic in praxis. Another approach is to impose a type of "interpolation" constraint on the original data [13].

Ideally, the filter F should correspond to some globally supported RBF. However, as the support of F grows, so does the cost of evaluating it at lattice sites. Moreover, the added benefit of a globally supported RBF is partially lost as the subsequent steps are entirely lattice-based and do not take the RBF into consideration. Our approach circumvents the choice of F entirely by posing the gradient approximation problem as a variational problem that, in addition to imposing the interpolation constraint, also imposes constraints on the "compactness" and "smoothness" of the gradient approximation. The only kernel we need to consider is the one that generates our target shift-invariant space, i.e. the space used to recover the smoothed indicator function. This type of variational reconstruction can be seen as an approximation to radial basis approaches [16], and is well-suited to our Poisson formulation in shift-invariant spaces.

Before presenting our approach (Section IV), we review the necessary background behind shift-invariant spaces and its connection to irregular sampling.

B. Sampling Lattices

In the univariate case, there is only one way to uniformly sample a signal; take equally spaced samples along the independent axis. However, when sampling multivariate signals, the situation is not as straightforward. A common approach is to inherit the ideas from the univariate case, and sample the function at regular intervals in each axis (known as a Cartesian sampling), while another might be to place samples at the centres of spheres arranged in the densest possible sphere packing (otherwise known as an FCC sampling). If $\mathbf{L}$ is a unimodular matrix, the set $\{\mathbf{L}\cdot\mathbf{n} : \mathbf{n} \in \mathbb{Z}^3\}$ (linear combinations of the vectors of $\mathbb{L}$ with integer coefficients), represents all possible sampling lattices. $\mathbf{L}$ is often called the *generating matrix* for a lattice. To control the sampling rate of a function, a uniform scaling parameter h is introduced. Typically, for the problem of surface reconstruction, our input data are not uniformly distributed in space (or even about the surface of the model). However, the concept of a sampling lattice is fundamental in forming the definition of our target approximation spaces (Section III-C).

The Cartesian lattice is given by the generating matrix $\mathbf{L}_C := \mathbf{I}_3$, where $\mathbf{I}_3$ is the 3×3 identity matrix. There are many factors that contribute to the ubiquity of the Cartesian lattice in practice. Possibly one of its most attractive features is its separability. Owing to this, the implementation of generating functions that span function spaces on the Cartesian lattice may be realized by a tensor product extension of existing univariate techniques. Filtering techniques are also easy to implement, as the Fast Fourier Transform is also separable, and may be applied independently to each axis of the data. However, in terms of sampling efficiency, the BCC lattice is the optimal lattice in $\mathbb{R}^3$, while the Cartesian is sub-optimal.

The Body-Centered Cubic (BCC) lattice is generated by the integer matrix

$$\mathbf{L}_B := \begin{bmatrix} -1 & 1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \end{bmatrix}.$$

However, the BCC lattice is non-separable, and requires special care in data retrieval and indexing. Figure 1 shows these lattices and their Voronoi regions, including the dual of the BCC lattice, the FCC lattice (the Cartesian lattice is dual to itself.)

C. Shift Invariant Spaces

As our target reconstruction spaces, we consider the shift-invariant spaces defined by

$$\mathbb{V}(\varphi, h\mathbf{L}) := \left\{ g(\mathbf{x}) := \sum_{\mathbf{m}\in\mathbb{Z}^3} c[\mathbf{m}]\varphi(\frac{\mathbf{x}}{h} - \mathbf{L}\mathbf{m}) : c \in \ell_2 \right\},$$

where φ is the generating function for the space, h is a scaling parameter that controls the sampling rate or granularity, and $\mathbf{L}$ is the generating matrix for a given sampling lattice. Representing a function f in a shift invariant space boils down to finding an appropriate coefficient $c[\mathbf{m}]$ vector that brings g as close as possible to the true underlying function f.

If the generating function that spans the space satisfies the *Strang-Fix* conditions of order k, and f belongs to the

Sobolev space of at least order k, then it can be shown that the minimum approximation error (in the least-squares sense) decays asymptotically as $O(h^k)$, and the space is said to have approximation order k. We often only have access to f through a discrete amount of samples, $\mathbf{f} := [f_1\ f_2\ \ldots\ f_M]^T$ located at sample points $S := \{\mathbf{x}_1, \mathbf{x}_2 \ldots \mathbf{x}_M\}$ respectively. When this is the case, the minimum-error approximation is unattainable. However, interpolative and quasi-interpolative methods [23], [24], which can be seen as oblique projections onto $\mathbb{V}(\varphi, h\mathbf{L})$, maintain the same order of accuracy.

For sufficiently smooth functions f that are uniformly sampled and properly prefiltered, the trilinear and linear reconstruction filters (Cartesian and BCC respectively) assure second order convergence, while the tricubic and Quintic assure fourth order. Another nicety of these spaces is that data processing techniques — derivative approximation for example — can often be represented by a convolution of the coefficient vector with either an *Infinite Impulse Response* (IIR) or *Finite Impulse Response* (FIR) filter.

Typically, as memory is a limited quantity, we are only interested in a finite number of points on a lattice. We denote the set of such points as $\{\mathbf{s}_1, \mathbf{s}_2,\ \ldots\ \mathbf{s}_N\} \subset \mathbf{L}\mathbb{Z}^3$. Corresponding to each lattice site is a coefficient c_k; we collect all such coefficients in the column vector $\mathbf{c} := [c_1\ c_2\ \ldots\ c_N]^T$. We also use the short hand notation $\varphi_k(\mathbf{x}) := \varphi(\frac{\mathbf{x}}{h} - \mathbf{s}_k)$; from this perspective, our function space is now

$$\left\{ \sum_{k=0}^{N} c_k \varphi_k(\mathbf{x}) \right\} \subset \mathbb{V}(\varphi, h\mathbf{L}). \tag{3}$$

Further, we denote $\mathbf{P}$ to be the $M \times N$ matrix whose elements are given by sampling the basis functions anchored at each lattice site, or $\mathbf{P}_{i,j} := \varphi_j(\mathbf{x}_i)$. Clearly, if our data are sampled at the lattice sites ($\mathbf{x}_i = \mathbf{s}_i$), then the problem of finding an interpolative scheme reduces to finding $\mathbf{c}$ such that $\mathbf{Pc} = \mathbf{f}$, which only involves inverting the square matrix $\mathbf{P}$. However, our sample locations are almost certain to not coincide with lattice sites, and we must consider alternative schemes for finding $\mathbf{c}$.

D. Box Splines

Box splines are compact piece-wise polynomial functions $M_{\mathbf{X}} : \mathbb{R}^s \to \mathbb{R}$. The matrix $\mathbf{X}$ denotes a collection of direction vectors, $\mathbf{X} := [\xi_1\ \xi_2 \ldots \xi_n]$. When $n = s$, the function $M_{\mathbf{X}}(\mathbf{x})$ is defined to be $\frac{1}{det|\mathbf{X}|}$ inside the polytope formed by the Minkowski sum of the direction vectors of $\mathbf{X}$, and zero otherwise (that is, $\frac{1}{det|\mathbf{X}|}$ when $\mathbf{x} \in \{\xi_1 t_1 + \ldots + \xi_s t_s : t_i \in [0, 1)\}$). When $n > s$, the box spline is recursively defined as

$$M_{\mathbf{X}}(\mathbf{x}) := \int_0^1 M_{\mathbf{X}/\xi_n}(\mathbf{x} - t\xi_n)\mathrm{d}t, \tag{4}$$

where $M_{\mathbf{X}/\xi_n}$ is the matrix obtained by removing the column vector ξ_n from $\mathbf{X}$. Box splines are a natural fit for our purposes, and are chosen to span our function spaces. One motivating factor that encourages the use of box splines on the BCC lattice over the Cartesian lattice is that the support of the box splines on the BCC lattice contains less lattice sites as opposed to their Cartesian counterparts. At the same time, they maintain the same order of accuracy and smoothness. This implies that one should expect roughly the same quality reconstruction, with less memory accesses per reconstructed value.

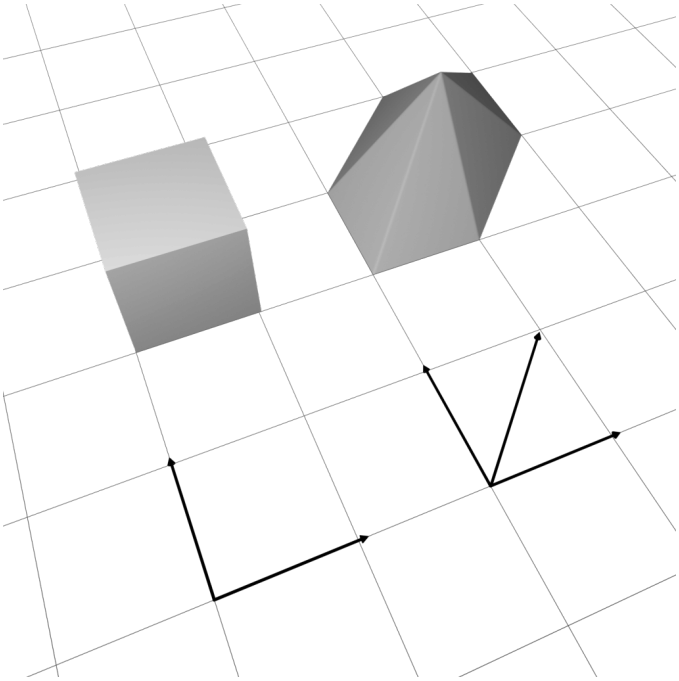

Fig. 2. An example of box splines in 2D. On the left is the box spline comprised of the direction vectors ξ_1, ξ_2, (the case n=s) which is the indicator function of the Minkowski sum of the vectors. The box spline on the right, includes a third vector ξ_3, which can be thought of as "smearing" the first spline along the direction ξ_3.

The evaluation of a box spline, in general, can be achieved by evaluating a simple recursive expression [19]. However, for many purposes, this form of evaluation is too slow. There has been a line of research [20] aimed towards obtaining fast piecewise polynomial representations of the box splines on the BCC lattice, and showing their advantage in speed and accuracy over splines on the Cartesian lattice when the data to be interpolated are sufficiently smooth.

E. Irregular Sampling

In the work by Kazdhan et al. [11] the range data are splatted via a linear filter onto a regular Cartesian grid. This tends to over-smooth the resulting reconstruction, as it allows the surface to deviate from the original sample locations. Ideally, we would like to find a function that not only interpolates the input data, but provides the representation with the smallest possible "shell" about the initial point set (i.e., the fewest non zero coefficients) while still interpolating the data and eliciting a unique solution.

Thus, we recast the problem of finding a representation of g within our function space as an optimization problem. There are similar works that impose variational constraints on random point samples for images [16], as well as multi-resolution techniques for volumetric data [17] and an extension

to box spline spaces [18]. In the same vein, we seek a function $g \in \mathbb{V}(\varphi, h\mathbf{L})$ that attempts to interpolate the input data, and minimize a functional E. Specifically, we seek a g that minimizes

$$\sum_{i=1}^{M}(g(\mathbf{x}_i) - f_i)^2 + E(g). \tag{5}$$

It is important to introduce a bit of notation before proceeding, $\mathbf{D_v}f := \mathbf{v} \cdot \nabla f$ defines the directional derivative of f in the direction $\mathbf{v}$. The Beppo-Levi inner product of order n is defined as $\langle f, g\rangle_{BL^n} := \sum_{|\mathbf{p}|=n}(n!/\mathbf{p}!)\langle \mathbf{D^p}f, \mathbf{D^p}g\rangle$, where $\mathbf{p}$ is an integer vector such that $|\mathbf{p}| := p_1+p_2+p_3$. We also use the short hand $\mathbf{D^p}f := \partial^{|\mathbf{p}|}f/(\partial^{p_1}x_1\partial^{p_2}x_2\partial^{p_3}x_3)$. The second order Beppo-Levi norm of f is defined as $||f||^2_{BL^2} := \langle f, f\rangle_{BL^2}$.

The second order Beppo-Levi norm can be thought to measure the "smoothness" of a function. However, constraining the smoothness of our approximation alone is not sufficient to elicit the desired solution in our case. We propose that the smoothed normal field should also minimize the inner product of the function with itself. This has the physical interpretation of forcing the solution to be close to zero everywhere except near the input points. Our proposed cost functional is then $E(g) := \lambda_1 ||g|| + \lambda_2 ||g||_{BL^2}$. Here, λ_1 controls the amount to which the function is penalized for large coefficient values. When combined with the interpolation constraint, this imposes the behavior of forcing the function to be zero away from the surface. The parameter λ_2 controls the smoothness of the resulting normal field.

Finding g that minimizes equation (5) involves rewriting the minimization problem in terms of $\mathbf{c}$ [18]. Following some straightforward algebraic manipulations (see [18] for details), the coefficient vector can be obtained by taking $\mathbf{c} = (\mathbf{P}^T\mathbf{P} + \lambda_1\mathbf{G} + \lambda_2\mathbf{S})^{-1}\mathbf{P}^T\mathbf{f}$, where $G_{i,j} := \langle\varphi_i, \varphi_j\rangle$ and $S_{i,j} := \langle\varphi_i, \varphi_j\rangle_{BL^2}$. Since φ is known in closed form, the entries of both $\mathbf{G}$ and $\mathbf{S}$ can be evaluated analytically. The above vector equation can be efficiently solved using the Conjugate Gradient method.

IV. Approach

A high level description of our approach is outlined in Figure 3. From the oriented surface samples, we construct the approximation $\vec{V}(\mathbf{x})$ using our variational formulation. We then apply FIR filters to the coefficients of the approximation to construct a scalar field representing the approximate divergence of the smoothed normal field (Figure 3c.) We then use an appropriate discretization of the inverse Laplacian operator to solve the Poisson equation (1), and obtain a smooth function that approximates the original indicator function (Figure 3c). Our code will is also available for download

A. Obtaining $\vec{V}(\mathbf{x})$

We define the vector field as $\vec{V}(\mathbf{x}) := \mathbf{L}\left[\upsilon_1(\mathbf{x})\ \upsilon_2(\mathbf{x})\ \upsilon_3(\mathbf{x})\right]^T$ where $\mathbf{L} := [\mathbf{b_1}\ \mathbf{b_2}\ \mathbf{b_3}]$ is a basis for $\mathbb{R}^3$ that contains the principal lattice directions of the chosen lattice. Each υ_i is a scalar function representing the projection of the smoothed normal field onto each respective principal lattice direction $\mathbf{b}_i$. This may seem unintuitive at first, however, using a non-canonical basis to represent $\vec{V}$ will be convenient in two ways. Firstly, it will allow us to approximate the divergence of $\vec{V}$ by taking directional derivatives of $\vec{V}$ in the principal lattice directions. This is motivated by the fact that it is more natural to approximate derivatives in the principal lattice directions on non-Cartesian lattices. Secondly, it will allow us to construct divergence of the smoothed normal field from more than three principal lattice directions. We also allow each function υ_i to potentially be represented in different function spaces, i.e.

$$\upsilon_i(\mathbf{x}) = \sum_{k=0}^{N}\mathrm{v}_k^i\varphi_k^i(\mathbf{x}) \in \mathbb{V}(\varphi^i, h\mathbf{L}). \tag{6}$$

The coefficient vector of υ_i is denoted as $\mathbf{v}_i := \begin{bmatrix}\mathrm{v}_1^i & \cdots & \mathrm{v}_N^i\end{bmatrix}$. We consider each component of $\vec{V}(\mathbf{x})$ separately, thus, it suffices to only look at one υ_i. We simply find the υ_i that minimizes

$$\sum_{\forall(\mathbf{x},\mathbf{n})\in P}^{M}(\upsilon_i(\mathbf{x}) - (\mathbf{L}^{-1})_i \cdot \mathbf{n})^2 + E(\upsilon_i). \tag{7}$$

Where $(\mathbf{L}^{-1})_i$ is the i^{th} column of $(\mathbf{L}^{-1})$. As before, this provides us with a parametrized way of controlling the smoothness of the resulting approximation, without the need to explicitly choose a smoothing kernel.

B. Divergence of $\vec{V}(\mathbf{x})$

A simple way to obtain the divergence of $\vec{V}(\mathbf{x})$ is to consider its analytical partial derivatives which, due to the linearity of the expression, requires one to only take the analytic partial derivatives of the underlying kernel functions. However, this reduces the smoothness of the solution [**?**]. There is also no conclusive evidence that suggests the analytic derivative of the kernel provides a better approximation to true underlying derivative. With this in mind, we choose to take a discrete filtering approach and approximate the divergence of $\vec{V}$ with a set of discrete FIR filters. While these filters do not strictly facilitate the best approximation scheme for derivatives, they provide a good compromise between compactness versus accuracy, even more so when the reconstruction spaces are shifted [2].

1) Spaces Spanned by a Single Generating Function: When each space is identical (when $\upsilon_i \in \mathbb{V}(\varphi, L)$) the divergence can be found by taking directional derivatives of $\vec{V}$ along the vectors in $\mathbf{L}$ as follows:

$$\begin{aligned}\nabla\cdot\vec{V}(\mathbf{x}) &= \nabla\cdot\mathbf{L}\left[\upsilon_1(\mathbf{x})\ \upsilon_2(\mathbf{x})\ \upsilon_3(\mathbf{x})\right]^T \\ &= \left[(\nabla\cdot\mathbf{b_1})\ (\nabla\cdot\mathbf{b_2})\ (\nabla\cdot\mathbf{b_3})\right]\cdot\left[\upsilon_1\ \upsilon_2\ \upsilon_3\right]^T \\ &= \mathbf{D_{b_1}}\upsilon_1 + \mathbf{D_{b_2}}\upsilon_2 + \mathbf{D_{b_3}}\upsilon_3.\end{aligned} \tag{8}$$

This allows us to approximate the divergence of $\vec{V}$ by simply convolving the coefficient vector corresponding to υ_i with an appropriate 1D derivative filter $d[n]$, then sum the coefficients. The expression for the resulting coefficient vector of the divergence is then

$$\mathbf{f} := \sum_{i=1}^{3}\mathbf{v}_i * \mathbf{d}_i \tag{9}$$

where $\mathbf{d}_i$ is a directional derivative filter obtained by orienting the 1D filter d along the direction $\mathbf{b}_i$.

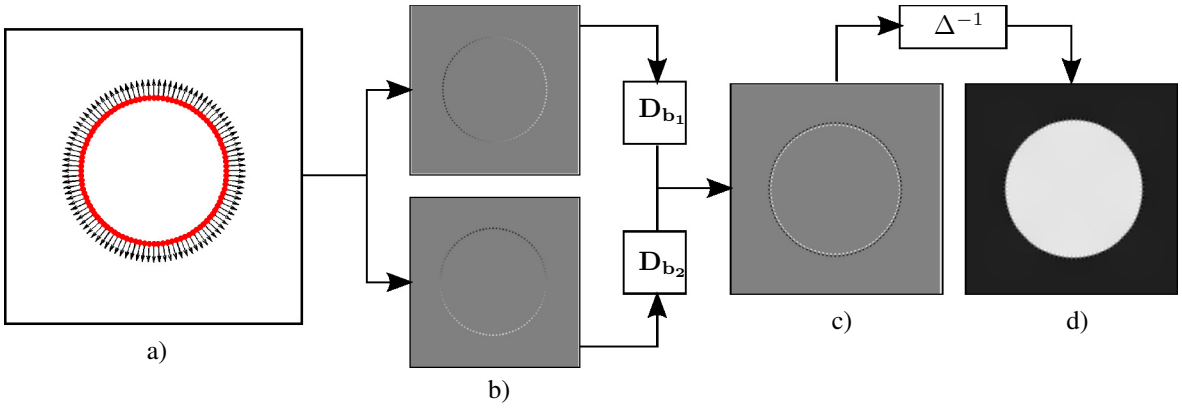

Fig. 3. A high level overview of our surface reconstruction pipeline in 2D. On the far left, a) shows the initial point-set. b) is the result of the smoothed variational reconstruction of the gradient of the indicator, the top image is the component in basis, the bottom is the y component. The approximate divergence of the smoothed indicator function is c), and d) is the resulting indicator function. [[XXX UA: Please fix this.]]

2) Shifted Reconstruction Spaces: To reconstruct within a shifted space, the input point-set is shifted by $\frac{1}{2}\mathbf{b_i}$ then reconstructed within $\mathbb{V}(\varphi, L)$. This is clearly equivalent to reconstructing in a shifted basis space, however, care must be taken to ensure that the data are brought back to a centered representation. Our choice of a shifted derivative filter $s[n]$ accounts for this and brings the resulting approximation back to the unshifted space. Both the shifted and unshifted filters we utilize are found in Table I and are chosen so that they provide a fourth-order discretization of the 1D derivative operator [2].

3) Four Direction Divergence on the BCC Lattice: It is also possible to consider a basis $\mathbf{L}$ in which we choose more than three principal directions to reconstruct the divergence. Thus, we may choose $\mathbf{L} = \mathbf{X_B}$ as the basis for our function. To accommodate for this, we redefine $\vec{V}(\mathbf{x}) := \mathbf{L}\,[v_1\ v_2\ v_3\ v_4]^T$ Since this matrix is not square, it is not possible to take its inverse in (7). Instead, the least norm inverse $\mathbf{L^T}(\mathbf{L}\mathbf{L}^T)^{-1}$ is used. It is easy to see that, by the same reasoning as in equation (8), $\nabla \cdot \vec{V} = \mathbf{D_{b_1}}v_1 + \mathbf{D_{b_2}}v_2 + \mathbf{D_{b_3}}v_3 + \mathbf{D_{b_4}}v_4$. Thus, we may use the same idea as in the three direction case, approximating each direction separately, and summing the coefficients to obtain an approximation to the divergence (the shifted case is also similar.)

C. Experiments and Results

1) Cartesian Lattice: Choosing $\mathbf{X} = \mathbf{X_{C2}} := [\mathbf{I_3}\ \mathbf{I_3}]$ (the 3×6 matrix created by the union of two identity matrices) generates a triliniear piecewise polynomial. It is easy to see, by definition, when $\mathbf{X_{C2}}$ is used in conjunction with Equation 4 on page 5, the resulting function $M_{\mathbf{X_{C2}}}(\mathbf{x})$ corresponds to the non-centered tensor product extension of the linear B-spline in three dimensions. Likewise, the non-centered cubic tensor product box spline is given by $M_{\mathbf{X_{C4}}}(\mathbf{x})$, where $\mathbf{X_{C4}} := [\mathbf{I_3}\ \mathbf{I_3}\ \mathbf{I_3}\ \mathbf{I_3}]$. A box spline is easily centered by taking $M_{\mathbf{X}}(\mathbf{x} + \mathbf{c_X})$, with $\mathbf{c_X} := \sum_{\xi \in \mathbf{X}} \frac{1}{2}\xi$. We will assume, unless otherwise stated, that $M_{\mathbf{X}}$ defines the centered box spline generated by $\mathbf{X}$.

Filter	Notation	Filter wieghts
Derivative	$d[n]$	$[\frac{-1}{12}\ \frac{2}{3}\ 0\ \frac{-2}{3}\ \frac{1}{12}]$
Shifted Derivative	$s[n]$	$[\frac{-1}{24}\ \frac{9}{8}\ \frac{-9}{8}\ \frac{1}{24}\ 0]$
Second Derivative	$l[n]$	$[\frac{-1}{12}\ \frac{4}{3}\ \frac{-5}{2}\ \frac{4}{3}\ \frac{-1}{12}]$

TABLE I. FILTERS USED IN OUR EXPERIMENTS. THESE FILTERS ARE APPLIED ALONG THE PRINCIPAL LATTICE DIRECTIONS.

2) BCC Lattice: On the BCC lattice, a choice of

$$\mathbf{X_B} := \begin{bmatrix} -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \end{bmatrix}$$

generates a piecewise linear polynomial function [1]. The support of this box spline is a rhombic dodecahedron, which incidentally contains the set of first nearest neighbors to the Voronoi region of the lattice at the origin, and is a natural fit for the lattice. The quintic box spline is given by the matrix $\mathbf{X_{B2}} := [\mathbf{X_B}\ \mathbf{X_B}]$, or may also be seen as the convolution of the linear box spline with itself. However, there exist both an efficient piecewise polynomial representation [1], and an efficient implementation [21] of these piecewise quintic polynomial functions.

3) Shifted Basis Functions: We also consider shifted box splines $M^{\xi_i}_{\mathbf{X}}(\mathbf{x}) := M_{\mathbf{X}}(\mathbf{x} - \frac{1}{2}\xi_i)$, where ξ_i is any direction vector in $\mathbf{X}$. In the univariate case with linear B-splines, it has been shown that introducing a shift can offer an improvement when representing scalar data [22]. More generally and relevant to our purposes, they have also been shown to reduce error when approximating the gradient of scalar fields [2].

V. Results and Discussion

For our tests, we restrict ourselves to the function spaces spanned by the Cubic tensor product basis on the Cartesian lattice, and the Quintic basis function on the BCC lattice. To provide a baseline reconstruction with which to compare our results, we consider FFT, Poisson and Screened Poisson Methods. We utilize the test models and point-sets of the Anchor,

Daratech, Dancing Children, Gargoyle, and Quasimoto models provided by the Surface Reconstruction Benchmark [14].

Quantitatively, we evaluate whether function spaces on the BCC lattice provide better approximation spaces than those on the Cartesian lattice. In order to reduce any artifacts that may arise from any noise or the non-uniformity of the data, we utilize the ideal reference point-sets of our test models. We measure both the Hausdorff distance, mean distance, and angle deviation from the original shape with the tools provided by Berger et al. [14]. We evaluate the FFT, Poisson, and Screened Poisson surface reconstruction algorithms, as their theory is close to ours. We choose λ_1 and λ_2 so that the reconstructions are comparable to the Screened Poisson results.

Since the Gargoyle model has many interesting high frequency features, we consider reconstructing it at increasing grid resolutions. For the Poisson and Screened Poisson methods, we restrict the depth of the tree so that the finest resolution provided by the oct-tree is equivalent to that of the regular Cartesian, Figure 4. We also seem to be able to improve upon the Screened Poisson method, however, the parameter space needs to be explored further to conclusively assert this. Regardless, it is still a good baseline with which to compare our results.

We fix the resolution of the grid at 128^3, and reconstruct the remaining models (Figure 5.) We choose this resolution as it seems to be a point of contention between the cases with the Gargoyle model. However, there seems to be a consistent advantage in using the BCC lattice over the Cartesian lattice. Moreover, in both cases, we noticed that introducing a shift tends to reduce all error metrics. Unfortunately, reconstructing the divergence of the smoothed vector field from more than three lattice directions produced little difference. Figure 6 shows a visual comparison of the techniques. We manage to outperform the FFT and Poisson techniques by a large margin, and outperform the screened Poisson case.

A. Robustness

When the input data contain noise, our method allows us to choose an appropriate smoothing parameter λ_2 to average out said error. However, our method fails to be as robust as the Screened Poisson algorithm when the data are non-uniform. We speculate that this is due to the fact that we employ no heuristics to account for the deviation from the assumption of uniformity (Figure 7.)

B. Complexity

In terms of re-sampling speed, representation memory consumption, and reconstruction time, the BCC approach tends to outperform the Cartesian. However, when compared to similar methods whose function spaces are sparse, our method is limited. Specifically, our memory requirement grows as $O(n^3)$ where n is the grid size, and the matrix solve is quite slow as n is large. While the regularization matrices on the BCC lattice are more sparse, and tend to converge faster, their size is still prohibitive for big n. These limitations prevent us from going to the same resolutions that similar methods allow.

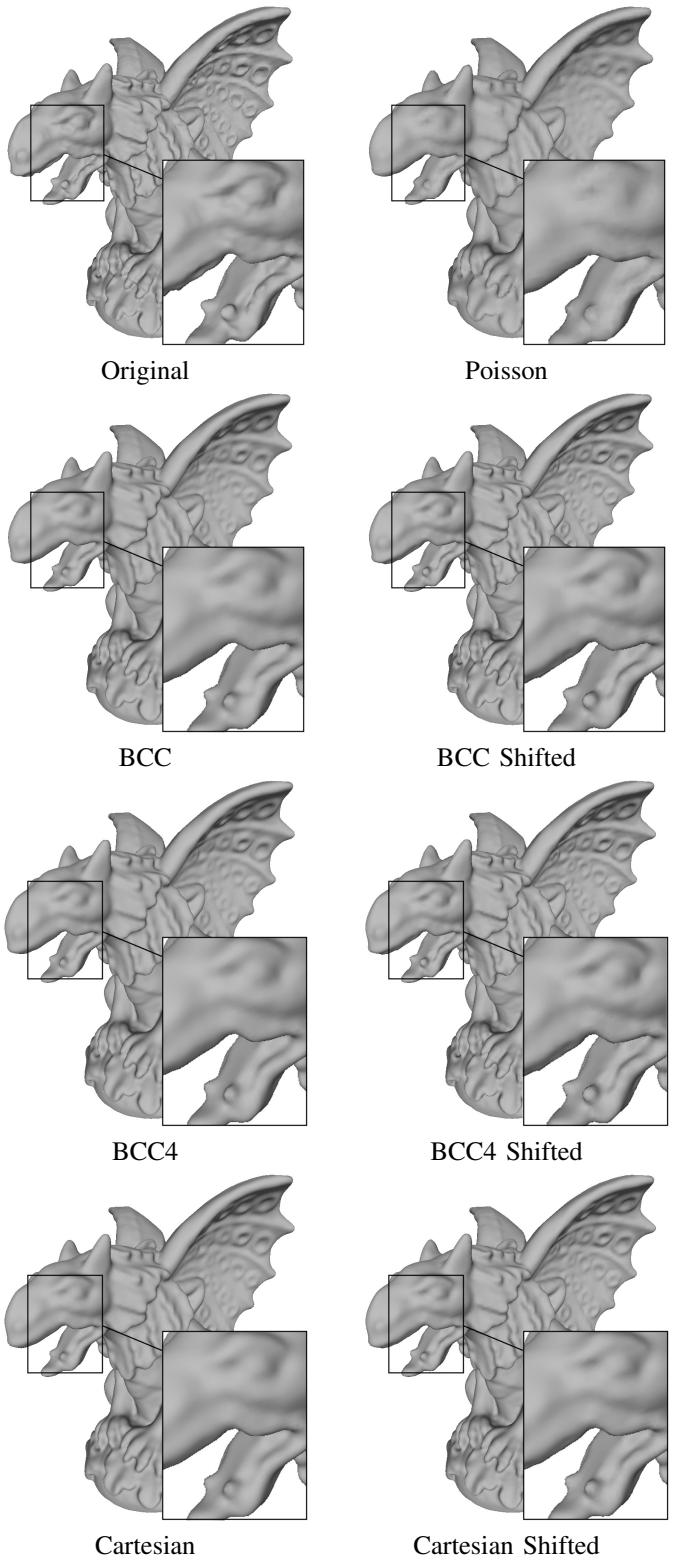

Fig. 6. A set of reconstructions on a 128^3 Cartesian grid, $50 \times 50 \times 100$ BCC grid or an oct-tree depth of 7. Parameters were chosen as $\lambda_1 = 1 \times 10^2$ and $\lambda_2 = 1 \times 10^{-4}$. Comparing between the Cartesian and BCC reconstruction spaces, the BCC lattice seems to consistently outperform the Cartesian lattice beyond a resolution of 128^3, moreover, the advantage of shifting the reconstruction space is apparent, specifically around the teeth and lips of the model.

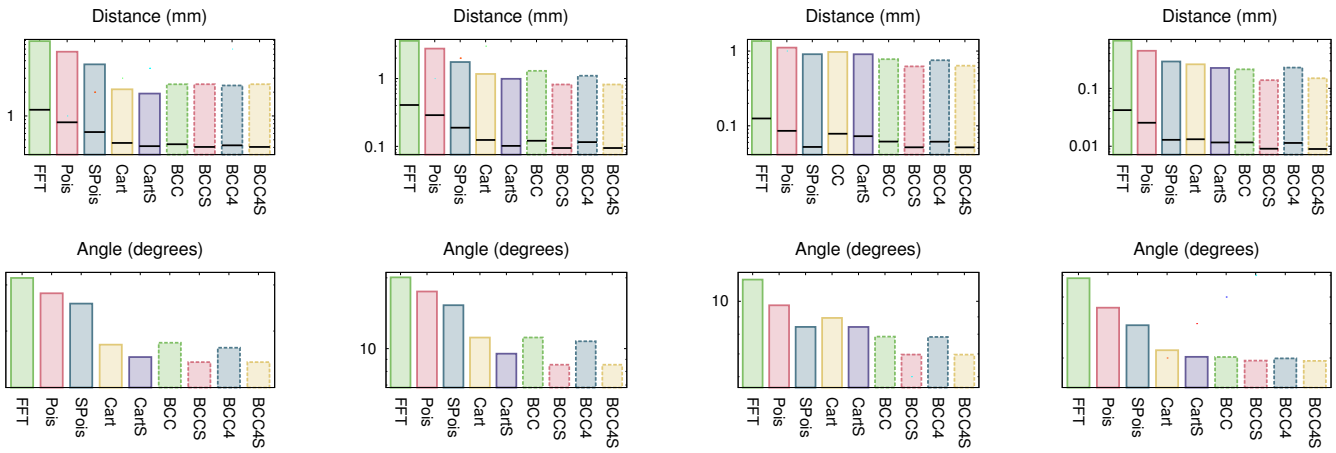

Fig. 4. An error comparison for the Gargoyle model as the grid resolution increases from 32^3 to 256^3 in steps of powers of two. The top row represents the Hausdorff (bar height) and mean distance (black line) to the original baseline model, while the bottom row represents the max angle deviation. $\lambda_1 = 100$, and λ_2, was chosen to be 1×10^{-3}, 1×10^{-4}, 1×10^{-5} and 1×10^{-6} from left to right respectively. Notice that the introduced shift greatly reduces angular error.

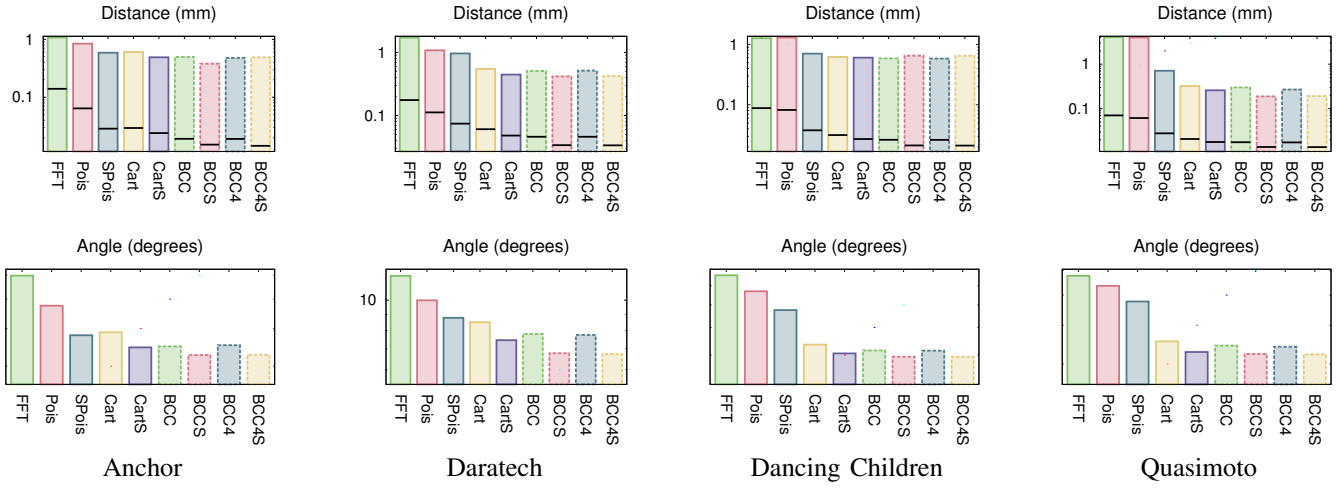

Fig. 5. A similar comparison for the case in which the grid is fixed at a resolution of 128^3 for all models. Here, $\lambda_1 = 1 \times 10^2$, and $\lambda_2 = 5 \times 10^{-5}$. Again, we see the same trend of the BCC lattice outperforming the comparable Cartesian lattices, and the Screened Poisson approach.

One possible remedy to this, is to construct our solution within a wavelet or multiresolution space defined over the BCC lattice. Since the function we wish to reconstruct only needs a detailed representation near the surface or the model, one can expect much better memory consumption. However, this is still an open problem, and needs further research. Another possibility, would be to break the initial point-set up into smaller sets, reconstruct the surface over those sets, then join the resulting meshes.

C. Parameter Selection

In our experiments, choosing λ_2 to be small reduced smoothing as expected. However, we observed that when λ_2 passed below a threshold, the surface began to show ringing or aliasing artifacts. This is somewhat to be expected, as choosing a very thin smoothing will cause the approximation (2) to break down. Additionally, the choice of λ_1, the compactness constraint, didn't seem to effect the solution much, it's only effect was on the choice of the parameter λ_2. That is, when λ_1 was large, λ_2 had to be large to obtain similar smooth solutions.

The parameterized nature of our solution allows us to control the degree to which the initial data are smoothed. However, this also makes it difficult to conclusively compare methods. More investigation and exploration of the parameter space is required. Moreover, there also appears to be a point at which choosing λ_2 to be too small introduces aliasing artifacts.

VI. Conclusion

The results regarding reconstruction methods are not surprising, they show that the Poisson methods in question are not using the full representation power of their approximation

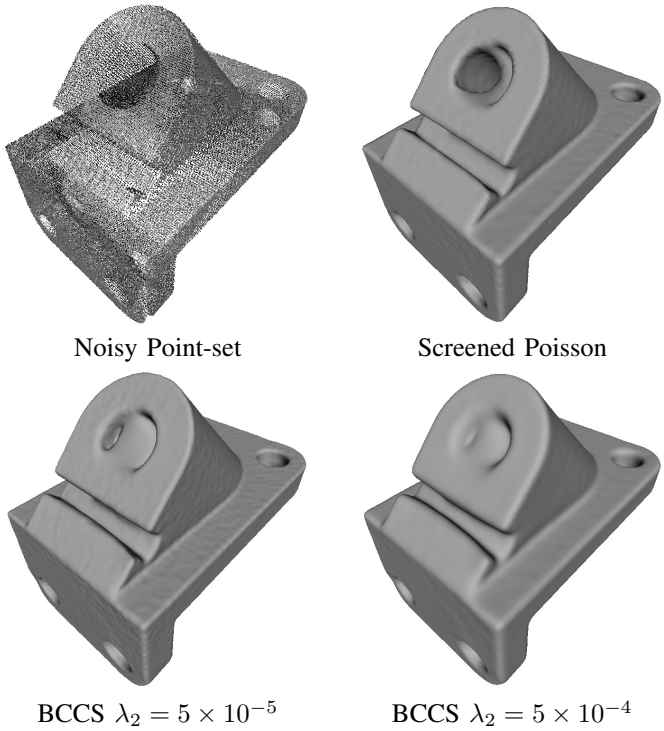

Fig. 7. Varying the smoothing parameter λ_2 allows us to smooth our noise from practical data sets ($\lambda_1 = 1 \times 10^2$). The Screened Poisson method is able to more faithfully reconstruct the center hole of the Anchor model. Our method, however, is unable to infer the importance of the samples on the interior of the Anchor.

spaces they use. Further, when the we focus on solely on the method presented in this paper, we see an improvement in reconstruction quality. Reconstruction space can play a significant role in the fidelity of surface reconstruction. From the optimality argument, it is not surprising to see that function spaces defined over the BCC lattice tend to reconstruct more details at equivalent resolutions when compared to those defined over the Cartesian lattice. Furthermore, reconstructing within shifted spaces seems to better reconstruct higher frequency details.

The main hindrance for this method its cubic memory requirement, which prevents us from utilizing high density grids, and prevents the method from being used in practical settings verbatim. However, since no multiresolution techniques are currently available on non-Cartesian lattices in 3D, such topics are subject of future work. Additionally, a more complete analysis is required for parameter selection between the two lattices, and research into a faster (perhaps multiresolution) variational algorithm is needed.

ACKNOWLEDGMENT

The authors would like to thank...

REFERENCES

[1] A. Entezari, D. Van De Ville, and T. Möller, "Practical box splines for reconstruction on the body centered cubic lattice," *IEEE Trans. Vis. and Comp. Graph.*, vol. 14, no. 2, pp. 313–328, 2008.

[2] U. R. Alim, T. Möller, and L. Condat, "Gradient estimation revitalized," *IEEE Trans. on Vis. and Comp. Graph.*, vol. 16, no. 6, pp. 1494–1503, November-December 2010.

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

[4] F. Cazals and J. Giesen, "Delaunay triangulation based surface reconstruction: Ideas and algorithms," in *Effective computational geometry for curves and surfaces*. Springer, 2006, pp. 231–273.

[5] N. Amenta, S. Choi, and R. K. Kolluri, "The power crust," in *Proc. ACM Symp. Solid Modeling and Applications*, ser. SMA '01. New York, NY, USA: ACM, 2001, pp. 249–266.

[6] N. Amenta, S. Choi, T. K. Dey, and N. Leekha, "A simple algorithm for homeomorphic surface reconstruction," in *Proc. SGP*, ser. SCG '00. New York, NY, USA: ACM, 2000, pp. 213–222.

[7] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Surface reconstruction from unorganized points," in *Proc. ACM SIGRAPH*, ser. SIGGRAPH '92. New York, NY, USA: ACM, 1992, pp. 71–78.

[8] F. Calakli and G. Taubin, "SSD: Smooth signed distance surface reconstruction," *Computer Graphics Forum*, vol. 30, no. 7, pp. 1993–2002, 2011.

[9] J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, and T. R. Evans, "Reconstruction and representation of 3D objects with radial basis functions," in *Proc. ACM SIGRAPH*, ser. SIGGRAPH '01. New York, NY, USA: ACM, 2001, pp. 67–76.

[10] I. Macedo, J. P. Gois, and L. Velho, "Hermite radial basis functions implicits," *CG Forum*, vol. 30, no. 1, pp. 27–42, 2011.

[11] M. Kazhdan, "Reconstruction of solid models from oriented point sets," in *Proc. SGP*, ser. SGP '05. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, 2005.

[12] M. Kazhdan, M. Bolitho, and H. Hoppe, "Poisson surface reconstruction," in *Proc. SGP*, ser. SGP '06. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, 2006, pp. 61–70.

[13] M. Kazhdan and H. Hoppe, "Screened Poisson surface reconstruction," *ACM Trans. Graph.*, vol. 32, no. 3, pp. 29:1–29:13, Jul. 2013.

[14] M. Berger, J. A. Levine, L. G. Nonato, G. Taubin, and C. T. Silva, "A benchmark for surface reconstruction," *ACM Trans. Graph.*, vol. 32, no. 2, pp. 20:1–20:17, Apr. 2013.

[15] J. Manson, G. Petrova, and S. Schaefer, "Streaming surface reconstruction using wavelets," *Proc. SGP*, vol. 27, no. 5, pp. 1411–1420, 2008.

[16] M. Arigovindan, M. Suhling, P. Hunziker, and M. Unser, "Variational image reconstruction from arbitrarily spaced samples: a fast multiresolution spline solution," *IEEE Trans. on Img. Proc.*, vol. 14, no. 4, pp. 450–460, 2005.

[17] E. Vuçini, T. Möller, and E. Gröller, "On visualization and reconstruction from non-uniform point sets using B-splines," *Computer Graphics Forum (Proc. of Eurographics 2009)*, vol. 28, no. 3, pp. 1007–1014, 2009.

[18] X. Xu, A. Alvarado, and A. Entezari, "Reconstruction of irregularly-sampled volumetric data in efficient box spline spaces," *IEEE Trans. Med. Img.*, vol. 31, no. 7, pp. 1472–1480, 2012.

[19] C. de Boor, K. Höllig, and S. Riemenschneider, *Box Splines (Applied Mathematical Sciences)*, softcover reprint of hardcover 1st ed. 1993 ed. Springer, 2 2010.

[20] A. Entezari, R. Dyer, and T. Möller, "Linear and cubic box splines for the body centered cubic lattice," in *Visualization, 2004. IEEE*, 2004, pp. 11–18.

[21] B. Finkbeiner, A. Entezari, D. Van De Ville, and T. Möller, "Efficient volume rendering on the body centered cubic lattice using box splines," *Comput. Graph.*, vol. 34, no. 4, pp. 409–423, Aug. 2010.

[22] T. Blu, P. Thevenaz, and M. Unser, "Linear interpolation revitalized," *IEEE Trans. Img. Proc.*, vol. 13, no. 5, pp. 710–719, May 2004.

[23] A. Entezari, M. Mirzargar, and L. Kalantari, "Quasi-interpolation on the body centered cubic lattice," in *Proc. Eurographics / IEEE - VGTC Conf. on Vis.*, ser. EuroVis'09. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, 2009, pp. 1015–1022.

[24] P. Thevenaz, T. Blu, and M. Unser, "Interpolation revisited," *IEEE Trans. Med. Img.*, vol. 19, no. 7, pp. 739–758, 2000.