

Biorthogonal Wavelet Surface Reconstruction Using Partial Integrations

Xiaohua Ren¹, Luan Lyu¹, Xiaowei He², Wei Cao¹, Zhixin Yang^{1,3}, Bin Sheng⁶, Yanci Zhang⁵ and Enhua Wu^{†1,2,4}

¹FST, University of Macau, ²State Key Lab. of CS, ISCAS & Univ. of CAS, ³ State Key Lab. of Internet of Things for Smart City, Univ. of Macau, ⁴Zhuhai-UM S&T Research Institute, ⁵ Sichuan University, ⁶ Shanghai Jiao Tong University

Abstract

We introduce a new biorthogonal wavelet approach to creating a water-tight surface defined by an implicit function, from a finite set of oriented points. Our approach aims at addressing problems with previous wavelet methods which are not resilient to missing or nonuniformly sampled data. To address the problems, our approach has two key elements. First, by applying a three-dimensional partial integration, we derive a new integral formula to compute the wavelet coefficients without requiring the implicit function to be an indicator function. It can be shown that the previously used formula is a special case of our formula when the integrated function is an indicator function. Second, a simple yet general method is proposed to construct smooth wavelets with small support. With our method, a family of wavelets can be constructed with the same support size as previously used wavelets while having one more degree of continuity. Experiments show that our approach can robustly produce results comparable to those produced by the Fourier and Poisson methods, regardless of the input data being noisy, missing or nonuniform. Moreover, our approach does not need to compute global integrals or solve large linear systems.

CCS Concepts

•Computing methodologies → Mesh geometry models; •Mathematics of computing → Computation of transforms;

1. Introduction

Reconstructing water-tight surfaces from oriented point sets (point samples with associated normals) has been widely studied in computer graphics. An ideal reconstruction method should be scalable and easy to implement efficiently. Besides, it should also be robust enough to handle various types of artifacts in point clouds since captured datasets may be noisy, non-uniform or missing. Manson et al. [MPS08] proposed an efficient streaming approach based on wavelets to reconstruct the surface from a large number of oriented point sets. The basic idea of their approach is to compute the wavelet coefficients of the indicator function of an unknown solid model (the indicator function is defined to be 1 inside the model and 0 outside of it). Compared to methods such as the Fourier [Kaz05] or Poisson method [KBH06], the wavelet approach is more efficient due to the local feature of wavelets. However, there are still two problems with the wavelet approach that remain unsolved. The first one is that it is not resilient to missing and nonuniformly sampled data [CT11]. The other is that the reconstructed surface tends to be non-smooth and may exhibit spurious high-frequency artifacts even when the input data is clean [KH13].

The reasons are twofold. The first one is that the key formula of computing wavelet coefficients in their method is derived under an assumption that the implicit function is an indicator function. In practice, this assumption could fail because we often need

to smoothly pre-filter the input data, which can be equivalently seen as implicitly filtering the indicator function (see the discussion in Sec.3.1). So it is inappropriate to apply the formula to the filtered indicator function especially when the data is incomplete or nonuniform. The other reason is that non-smooth wavelets were used for their small compact support. More specifically, Haar (db1) and db2 Daubechies wavelets were used in their work. In general, it is difficult to choose the right wavelets that are both smooth and efficient because smooth wavelets usually require large supports.

In this paper, we propose two techniques to address the aforementioned two problems. For the first problem, we derive a new formula to compute the wavelet coefficients of the implicit function. Our formula is based on three-dimensional partial integrations and does not require the integrated function to be an indicator function. Thus, our formula allows us to smoothly pre-filter the data. It can be proven that the formula used in [MPS08] is a special case of our formula when the integrated function is an indicator function. For the second problem, Daubechies wavelets have poor smoothness due to their construction with emphasis on the property of orthogonality [Dau92]. Our first attempt is to use existing smoother biorthogonal wavelets such as $\text{cdf}(n+1).m$, the B-spline wavelets constructed by Cohen-Daubechies-Feauveau [CDF92]. We have tested $\text{cdf}3.1$ (second-order B-spline wavelets with one vanishing moment) with the same support size as db2, but the quality of results are not good. Although other wavelets such as $\text{cdf}3.3$ and $\text{cdf}3.5$ can produce good results, they have larger support size. This motivates us to construct new smooth wavelets. Based on Lemarié-

[†] ehwu@umac.mo

Rieusset's proposition [LR92] recently introduced in [RLH*17], we propose a simple method to reconstruct smooth biorthogonal wavelets with small support and yet produce good smooth surfaces. Our method explicitly gives the filters of the constructed wavelets, which is convenient for us to perform the fast wavelet transform.

In summary, we present a biorthogonal wavelet approach to creating smooth water-tight surfaces from oriented point sets. Our approach is robust to handle noisy, missing or non-uniform data. The key contributions are summarized as follows:

- A general formula to compute the wavelet coefficients of the implicit function of the unknown solid model. Our formula does not require the implicit function to be an indicator function.
- A simple yet general method to construct smooth wavelets with small support size.
- Specifically, with our method, a family of biorthogonal wavelets is constructed with almost the same support size as Daubechies wavelets while having one more degree of continuity.

2. Related Work

There are numerous works on surface reconstruction from point clouds. Here we review those closely related to ours and refer the reader to [BTS*17] for a brief survey on recent developments in this field or to [BLN*13] for comprehensive evaluations on a number of algorithms.

In summary, surface reconstruction methods can be categorized as follows:

Combinatorial Methods. This family of methods typically produce an interpolating surface where vertices are from all or a subset of the input points. Delaunay/Voronoi based methods [CG06] use a subset of the input points to discretize the space of the underlying surface, and then extract a surface triangulation using specified algorithms. Famous algorithms include Super Cocone [DGH01] and its improvements [TKRL11, DG06], Power Crust [ACK01], Spectral reconstruction [KSO04] and graph cut [LPK09, HK06]. Streaming surface triangulation algorithms [ACA07, BMR*99] have also been developed in the pursuit of processing large data sets. Xiong et al. [XZZ*14] propose a novel method based on dictionary learning. Boltcheva and Lévy [BL17] design a fast, parallelizable surface triangulation algorithm by computing the restricted Voronoi diagram of the input points under the assumption that the input points are located exactly on the underlying surface. These algorithms come with guaranteed reconstruction quality [Dey06]. However, the reconstruction that restricts to have vertices only on the input points is not robust when the data is non-uniform, missing, or noisy.

Implicit Methods. In contrast to combinatorial methods, implicit methods construct an approximate surface using the zero-level set of a best-fit implicit function of the input points. The implicit function could be a sum of radial bases [CBC*01] or piecewise polynomial functions [NOS09, OBA05, OBA*03], or a signed-distance function estimated in [HDD*92, BBX95, CL96].

The Fourier surface reconstruction of Kazhdan [Kaz05] uses an indicator function as the implicit function to represent the underlying solid model of the input oriented points. The indicator function

can be recovered from its Fourier coefficients computed by surface integrals over the oriented points. The reconstructed surface is smooth and the method robustly handles noise in the data. However, both computational and storage costs of computing the coefficients are high since the basis functions are globally supported and a uniform grid is used. This approach was improved in [SBS07] with an adaptive Fourier technique. Manson et al. [MPS08] propose an efficient streaming approach by replacing the Fourier bases with compactly supported wavelets. Recently, their approach has been extended to consolidating polygon soup by Hu et al. [HLL*18].

Fourier surface reconstruction was later modified by Kazhdan et al. [KBH06], which leads to the famous Poisson surface reconstruction. The key idea is to find an implicit function whose gradient best matches the normals of the points. This is equivalent to solving a Poisson equation. Later, streaming [BKBH07], parallel [BKBH09] and GPU [ZGHG11] implementations are available.

Calakli and Taubin [CT11] develop a variational formulation to optimize a smoothed signed-distance function by constraining the position, normal and Hessian. Inspired by their work, Kazhdan and Hoppe [KH13] address the over-smoothing problem of Poisson reconstruction by adding a position constraint to "screen" the associated Poisson equation. Estellers et al. [ESS16] present a robust method by reformulating the position and normal constraints using Huber loss functions. Tang et al. [TF18] optimize a curvature-adaptive signed-distance function on an octree. Since the curvature was estimated directly on point sets, their method may not perform well on incomplete data. Liu et al. [LSYD17] use total variation regularization to deal with the problem of spurious sheets. Pan et al. [PTC16] use multilevel algebraic spline surfaces to represent the implicit function. Schertler et al. [STJ*17] propose an online algorithm for reconstructing surfaces from successively acquired scans. Schroers et al. [SSW14] present a general variational framework for most of these algorithms and also give some improvements.

Recently, the sample scale in surface reconstruction has received much attention. Fuhrmann and Goesele [FG11] develop a fusion method to estimate the scale information of samples from depth maps. In their later work [FG14], taken as input an oriented point cloud with scale information, the floating scale implicit function was developed for constructing a surface with various scales of geometric details. Ummenhofer and Brox [UB17] propose to globally optimize a signed-distance function on an octree using the same input. Since the scale of points is taken into account, the surfaces reconstructed by these algorithms typically have more geometric details than those constructed only from oriented points.

Our work is closely related to the wavelet surface reconstruction [MPS08]. We focus on developing new wavelets with regard to their properties of smoothness, support size and vanishing moment. We also present a general integral formula to computing wavelet coefficients. Horacsek and Alim [HA17] construct biorthogonal wavelets on the Body Centered Cubic (BCC) lattice and use them to represent volumetric data. Our wavelet construction method is based on Lemarié-Rieusset's proposition [LR92] recently introduced in the work [RLH*17]. They develop an efficient approximate wavelet algorithm for gradient-domain image composition and use existing wavelets as their bases. Note that the new family

of wavelets constructed by our method can be also used for their algorithm.

3. Background

We first review the wavelet [MPS08] and Fourier [Kaz05] surface reconstruction methods under a unified framework and give some discussion. Then, we briefly introduce the wavelet theory.

3.1. Wavelet and Fourier Surface Reconstruction Methods

Let $\{p_i\}$ be a set of points sampled from the surface ∂M of a solid M and $\{\mathbf{n}_i\}$ the associated outward normals. The problem is how to reconstruct the surface ∂M from the oriented point set $\{p_i, \mathbf{n}_i\}$.

Both methods define the surface as a level set of an indicator function. They first construct the indicator function χ_M of the solid M from $\{p_i, \mathbf{n}_i\}$, and then return the surface ∂M by extracting an appropriate level set of χ_M . The key is how to construct χ_M .

Without loss of generality, let M be embedded in the cube $D = [0, 1]^3$. Given an orthogonal basis $B_{\mathbf{k}}$ with $B_{\mathbf{k}}(\mathbf{x}) : D \rightarrow \mathbb{R}$, χ_M can be linearly represented as $\chi_M = \sum_{\mathbf{k}} c_{\mathbf{k}} B_{\mathbf{k}}$. Here $\mathbf{k} = [k_1, k_2, k_3]^T \in \mathbb{Z}^3$ is the set of indices and $\mathbf{x} = [x_1, x_2, x_3]^T$. Kazhdan [Kaz05] and Manson et al. [MPS08] obtain the coefficients $c_{\mathbf{k}}$ by computing surface integrals

$$c_{\mathbf{k}} = \int_D \chi_M(\mathbf{x}) B_{\mathbf{k}}(\mathbf{x}) d\mathbf{x} \quad (1a)$$

$$= \int_M B_{\mathbf{k}}(\mathbf{x}) d\mathbf{x} \quad (1b)$$

$$= \int_M \nabla \cdot \mathbf{F}_{\mathbf{k}}(\mathbf{x}) d\mathbf{x} = \int_{p \in \partial M} \mathbf{F}_{\mathbf{k}}(p) \cdot \mathbf{n}(p) d\sigma, \quad (1c)$$

where $\mathbf{F}_{\mathbf{k}} = [F_{k_1}, F_{k_2}, F_{k_3}]^T$ is a vector-valued function on D satisfying $\nabla \cdot \mathbf{F}_{\mathbf{k}} = B_{\mathbf{k}}$, $\mathbf{n}(p)$ is the outward unit surface normal to ∂M at point p and $d\sigma$ is the differential surface area of ∂M . Eq. (1a) follows from $B_{\mathbf{k}}$ is orthogonal. Eq. (1b) follows from that the indicator function on the domain $D \setminus M$ is equal to zero. Eq. (1c) follows from the Divergence Theorem. The construction of $\mathbf{F}_{\mathbf{k}}$ is not unique. Kazhdan [Kaz05] and Manson et al. [MPS08], respectively, give the form of $\mathbf{F}_{\mathbf{k}}$ using their own basis. Given a particular function $\mathbf{F}_{\mathbf{k}}$, $c_{\mathbf{k}}$ can be estimated by discretizing (1c) over the point samples $\{p_i, \mathbf{n}_i\}$

$$c_{\mathbf{k}} = \sum_i \mathbf{F}_{\mathbf{k}}(p_i) \cdot \mathbf{n}_i d\sigma_i, \quad (2)$$

where $d\sigma_i$ is an estimate of the differential surface area associated with the sample point p_i . Finally, we get $\chi_M = \sum_{\mathbf{k}} c_{\mathbf{k}} B_{\mathbf{k}}$ by an inverse transform.

Discussion 1. When the input points $\{p_i, \mathbf{n}_i\}$ are clean and uniformly sampled from the model surface, the estimation of $c_{\mathbf{k}}$ using Eq. (2) works quite well. However, the data in real scans may be noisy, nonuniform or even incomplete. A pre-smoothing step is often needed. More specifically, we get a smoothed normal field $\mathcal{S} * \mathbf{n}$ by convoluting the normals $\mathbf{n} = \{\mathbf{n}_i\}$ with some smoothing filter \mathcal{S} . Following the lemma in [KBH06], which states that the gradient of the smoothed indicator function $\tilde{\chi}_M = \mathcal{S} * \chi_M$ is equal to the smoothed normal field, i.e. $\nabla \tilde{\chi}_M = \mathcal{S} * \mathbf{n}$, our goal actually becomes the reconstruction of a smoothed indicator function $\tilde{\chi}_M$ from $\mathcal{S} * \mathbf{n}$.

Recalling (1b) and (1c) are derived under the assumption that the integrated function should be an indicator function, it would be inappropriate to still use Eq. (2) to estimate the coefficients of the smoothed indicator function.

Discussion 2. The basis $B_{\mathbf{k}}$ has a significant influence on the efficiency of reconstruction and quality of reconstructed surfaces. For example, the use of the Fourier basis in [Kaz05] can produce smooth surfaces but at high computational and storage costs, and the use of orthogonal wavelets such as db1 and db2 in [MPS08] is efficient but the reconstructed surface is non-smooth. On the other hand, since the functions $\mathbf{F}_{\mathbf{k}}$ satisfy $\nabla \cdot \mathbf{F}_{\mathbf{k}} = B_{\mathbf{k}}$, $B_{\mathbf{k}}$ also influence the choice of $\mathbf{F}_{\mathbf{k}}$. In Sec.4.1, we will discuss how to construct basis $B_{\mathbf{k}}$ and $\mathbf{F}_{\mathbf{k}}$, and its effect on the accuracy of reconstruction.

3.2. Wavelet Theory

Since the discussed orthogonal bases (the Fourier base and orthogonal wavelets) could not get a satisfying balance between efficiency and quality, we seek to use biorthogonal wavelets. First, we give a brief introduction to wavelet theory and refer the reader to the book [Mal08] or [Kai11] for more details.

One important step of wavelet theory is to hierarchically subdivide an interested function space using some compactly supported function called *scaling function*, which satisfies the two-scale relation

$$\varphi(t) = \sum_{k \in \mathbb{Z}} h_k \varphi(2t - k),$$

where the coefficients $\{h_k\}$ are called scaling (or low-pass) filter. This subdivision named multiresolution analysis (MRA) can be formally defined as follows.

Definition 1 A *multiresolution analysis* (MRA) of a function space is a sequence of closed subspaces $\{V_j\}$ satisfying $V_j \subset V_{j+1}$ and some other properties defined in [Mal08], where each $V_j = \text{span}\{\varphi_{j,k} = \varphi(2^j t - k)\}$.

Intuitively speaking, V_{j+1} is bigger or has more details than V_j . Wavelets appear as bases of complementary spaces W_j such that $V_{j+1} = V_j \oplus W_j$, where \oplus indicates the direct sum. One can construct a function ψ , called a *wavelet* such that $W_j = \text{span}\{\psi_{j,k} = \psi(2^j t - k)\}$. Since $W_j \subset V_{j+1}$, we can linearly represent the wavelet using φ

$$\psi(t) = \sum_{k \in \mathbb{Z}} g_k \varphi(2t - k), \quad (3)$$

where $\{g_k\}$ are called wavelet (or high-pass) filter.

So far, we have defined the bases φ_k and ψ_k for subspaces V_j and W_j , respectively. However, they may not be orthogonal. In order to compute the coefficients of a function represented by them, we need to construct their dual bases, denoted by $\tilde{\varphi}_k$ and $\tilde{\psi}_k$, respectively. At first, we give the definition of dual basis.

Definition 2 Let \tilde{U}_k and U_k be two sets of a Hilbert space equipped with the inner product $\langle \cdot, \cdot \rangle$. \tilde{U}_k is called a *dual set* of U_k if $\langle U_i, \tilde{U}_j \rangle = \delta_{i,j}$ for all $\tilde{U}_i \in \{\tilde{U}_k\}$ and $U_j \in \{U_k\}$. Specially, \tilde{U}_k is called the *dual basis* of U_k if each of them is a basis of the vector space.

Here $\delta_{i,j}$ is the Kronecker delta that is equal to one if $i = j$ and zero otherwise. In the case of U_k being orthogonal, $\tilde{U}_k = U_k$.

Given fixed bases ϕ_k and ψ_k with associated filters $\{h_k, g_k\}_{k \in \mathbb{Z}}$, we can find their dual bases $\tilde{\phi}_k$ and $\tilde{\psi}_k$ using

$$\tilde{\phi}(t) = \sum_{k \in \mathbb{Z}} \tilde{h}_k \tilde{\phi}(2t - k), \quad \tilde{\psi}(t) = \sum_{k \in \mathbb{Z}} \tilde{g}_k \tilde{\psi}(2t - k). \quad (4)$$

Here, the dual filters $\{\tilde{h}_k, \tilde{g}_k\}_{k \in \mathbb{Z}}$ are related to $\{h_k, g_k\}_{k \in \mathbb{Z}}$ by

$$g_k = (-1)^{1-k} \tilde{h}_{1-k}, \quad \tilde{g}_k = (-1)^{1-k} h_{1-k}, \quad \forall k \in \mathbb{Z}. \quad (5)$$

For compactly supported wavelets, these filters have a finite number of non-zero coefficients. We can find the *smallest interval* for such a filter that the coefficients outside of it are zeros. We denote it by $[b, n]$ with b being the first index and n the length.

4. Our approach

Recalling our discussion in Sec. 3.1, we have two challenges with the current formula (1c). The first one is that it is inappropriate to use the formula to compute the smoothed indicator function $\tilde{\chi}_M$. The other is that it is difficult to choose appropriate B_k and \tilde{F}_k . In this section, we first derive a general formula for computing the coefficients of $\tilde{\chi}_M$, and then discuss the construction of \tilde{F}_k and B_k . Finally, we describe our wavelet construction method.

4.1. A General Formula for Computing Coefficients

The following derivation is under the same framework as used in Sec. 3.1, except we use more general biorthogonal bases.

Our Formula. Let $\tilde{\chi}_M$ be a smoothed indicator function of the solid model M . Given a biorthogonal basis B_k associated with its dual basis \tilde{B}_k , $\tilde{\chi}_M$ can be linearly represented as $\tilde{\chi}_M = \sum_k c_k B_k$. We obtain the coefficients c_k by computing volume integrals

$$c_k = \int_D \tilde{\chi}_M(\mathbf{x}) \tilde{B}_k(\mathbf{x}) d\mathbf{x} \quad (6a)$$

$$= \int_D \tilde{\chi}_M(\mathbf{x}) \nabla \cdot \tilde{F}_k(\mathbf{x}) d\mathbf{x} \quad (6b)$$

$$= \int_{p \in \partial D} \tilde{F}_k(p) \cdot \mathbf{n}(p) d\sigma - \int_D \nabla \tilde{\chi}_M \cdot \tilde{F}_k d\mathbf{x} \quad (6c)$$

$$= - \int_D \nabla \tilde{\chi}_M \cdot \tilde{F}_k d\mathbf{x}, \quad (6d)$$

where $\tilde{F}_k = [\tilde{F}_{k_1}, \tilde{F}_{k_2}, \tilde{F}_{k_3}]^T$ is a vector-valued function on D satisfying $\nabla \cdot \tilde{F}_k = \tilde{B}_k$. Eq. (6a) follows from the definition of dual basis. Eq. (6c) follows from a *Partial Integration*, where $\mathbf{n}(p)$ is the outward unit surface normal to ∂D at point p and $d\sigma$ is the differential surface area of ∂D . The first term of (6c) would be equal to 0 if \tilde{F}_k was constructed to equal $\mathbf{0} = [0, 0, 0]^T$ on the boundary ∂D .

Different from the formula (1c) being a surface integral and only valid for indicator functions, our formula (6d) is a volume integral and valid for any function in $\text{span}\{B_k\}$. Specially, (1c) is one case of our formula when $\tilde{\chi}_M$ degenerates to χ_M in our formula. The middle and right columns of Fig. 1 show 1D smoothed indicator functions reconstructed using (6d) and (1c) from the two smoothed gradient fields shown on the left. In this test, we use second-order

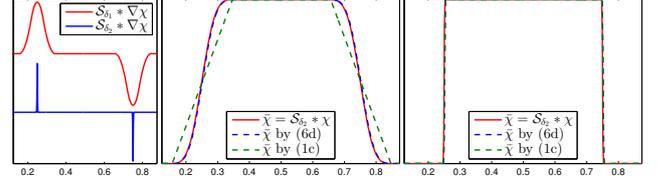


Figure 1: Reconstruction of 1D smoothed indicator functions from smoothed gradient fields. Left: Two input smoothed gradient fields obtained by convoluting $\nabla \chi$ with smoothing filters S_{δ_1} and S_{δ_2} , respectively. Middle: Smoothed indicator functions reconstructed using Eq. (6d) (dashed blue) and Eq. (1c) (dashed green) from the smoothed gradient field red-colored on the left, and the ground truth (red). Right: Results reconstructed from the smoothed gradient field blue-colored on the left, and the ground truth (red).

B-splines as our smoothing filters S_{δ_1} and S_{δ_2} with δ_1 and δ_2 representing their filter sizes, and choose db2 as the basis for both (6d) and (1c). For the computation of (1c), we follow the method [MP-S08] adapted to 1D case. We can see that (6d) can accurately recover smoothed indicator functions for large filter size δ and the original indicator function as δ tending towards 0. Next, we will discuss how to construct \tilde{F}_k and its effect on B_k .

Construction of \tilde{F}_k . Ignoring the condition $\nabla \cdot \tilde{F}_k = \tilde{B}_k$, we have the following lemma.

Lemma 1 Eq. (6d) holds for any function in $\text{span}\{B_k\}$ if and only if $-\tilde{F}_k$ is a dual set of ∇B_k .

Proof: We leave the proof to Appendix A.

Note that the lemma also holds for 1D and 2D. Since the number of dual sets of ∇B_k is infinite, we can not determine a particular one by the lemma.

In the derivation of Eq. (6d), we have a sufficient condition that if \tilde{F}_k satisfy $\nabla \cdot \tilde{F}_k = \tilde{B}_k$, then Eq. (6d) holds. However, given an arbitrary B_k , there are still many choices of \tilde{F}_k satisfying the condition. For 1D case, we can find B_k such that the choice is unique.

For simplicity, we rewrite $-\tilde{F}_k(\mathbf{x})$ by $\tilde{B}_k^0(t) : [0, 1] \rightarrow \mathbb{R}$ and $B_k(\mathbf{x})$ by $B_k^1(t) : [0, 1] \rightarrow \mathbb{R}$ with its dual basis $\tilde{B}_k^1(t)$. Here $k \in \mathbb{Z}$ is the set of indices. We now have the following relations

$$\nabla B_k^1(t) = (B_k^1)'(t) = B_k^0(t), \quad (7a)$$

$$\nabla \cdot \tilde{B}_k^0(t) = (\tilde{B}_k^0)'(t) = -\tilde{B}_k^1(t). \quad (7b)$$

By the lemma for 1D case, \tilde{B}_k^0 computed from the integral Eq. (7b) is a dual set of B_k^0 . Recalling that while there are many dual sets of B_k^0 , there is only one called dual basis. Thus, our attempt is to find (B_k^1, \tilde{B}_k^1) such that (1) B_k^0 by Eq. (7a) forms a basis of $\text{span}\{B_k^1\}$; (2) \tilde{B}_k^0 by Eq. (7b) forms a basis of $\text{span}\{\tilde{B}_k^1\}$. In this case, by Definition 2, we can infer that \tilde{B}_k^0 is just the dual basis of B_k^0 . If such (B_k^1, \tilde{B}_k^1) exists, we would get two conclusions.

- **C 1.** Eqs. (7a, 7b) provide us a way to construct new basis. More specifically, given such (B_k^1, \tilde{B}_k^1) , (B_k^0, \tilde{B}_k^0) can be constructed by computing the derivative of B_k^1 and integral of \tilde{B}_k^1 , respectively.
- **C 2.** More importantly, observing that Eqs. (7a, 7b) are antisymmetric, they still hold for the pair of $-(\tilde{B}_k^0, B_k^0)$ and (\tilde{B}_k^1, B_k^1) (here the former is constructed using the latter in C1). This means we

can use the constructed \tilde{B}_k^0 instead of B_k^1 as the basis for $\tilde{\chi}_M$. The benefit is that \tilde{B}_k^0 has one more degree of continuity than B_k^1 as it is the integral of B_k^1 .

Generally, it is hard to find such basis. Fortunately, wavelets satisfy the conditions according to Lemarié-Rieusset's proposition [LR92] (see Sec.4.2).

So far, we have discussed the construction of \tilde{F}_k for 1D case. By tensor-products of $B_k^1(t)$, we define $B_k^1(\mathbf{x}) = B_{k_1}^1(x)B_{k_2}^1(y)B_{k_3}^1(z)$ and its dual basis $\tilde{B}_k^1(\mathbf{x}) = \tilde{B}_{k_1}^1(x)\tilde{B}_{k_2}^1(y)\tilde{B}_{k_3}^1(z)$, and then construct

$$\tilde{F}_k = -[\tilde{B}_{k_1}^0\tilde{B}_{k_2}^1\tilde{B}_{k_3}^1, \tilde{B}_{k_1}^1\tilde{B}_{k_2}^0\tilde{B}_{k_3}^1, \tilde{B}_{k_1}^1\tilde{B}_{k_2}^1\tilde{B}_{k_3}^0]^T. \quad (8)$$

They satisfy $\nabla \cdot \tilde{F}_k = \tilde{B}_k^1$ using Eq. (7b). It can be checked that the so defined $-\tilde{F}_k$ is truly a dual set of ∇B_k^1 (Lemma 1).

4.2. New Wavelets

In the above section, we have obtained a general formula for computing coefficients using an arbitrary biorthogonal basis. We also give a construction of \tilde{F}_k , which is closely related to the choice of 1D basis (B_k^1, \tilde{B}_k^1) . We find that it would be useful if (B_k^0, \tilde{B}_k^0) constructed from (B_k^1, \tilde{B}_k^1) by Eqs. (7a, 7b) satisfying two conditions: (1) B_k^0 forms a basis of $\text{span}\{B_k^1\}$; (2) \tilde{B}_k^0 forms a basis of $\text{span}\{\tilde{B}_k^1\}$. Lemarié-Rieusset's proposition [LR92] shows that wavelets satisfy the conditions, and gives the specific form of Eqs. (7a, 7b) as well.

In this section, we will present a practical algorithm to construct (B_k^0, \tilde{B}_k^0) based on the proposition. With the algorithm, we construct a family of wavelets, which is smoother than Daubechies wavelets and has better reconstruction accuracy than B-Spline wavelets. First, let's introduce the proposition. Please refer to [D-P09, RLH*17] for more details.

Proposition 1 (Lemarié-Rieusset's Proposition)

Let $\{V_j^1\}$ be a MRA of $L^2(\mathbb{R})$, with associated scaling and wavelet functions (φ^1, ψ^1) and their dual functions $(\tilde{\varphi}^1, \tilde{\psi}^1)$. Then, there exists a MRA $\{V_j^0\}$ of the space, with associated scaling and wavelet functions (φ^0, ψ^0) and their dual functions $(\tilde{\varphi}^0, \tilde{\psi}^0)$, satisfying:

$$(\varphi^1)'(t) = \varphi^0(t) - \varphi^0(t-1), \quad (9a)$$

$$(\tilde{\varphi}^0)'(x) = \tilde{\varphi}^1(x+1) - \tilde{\varphi}^1(x), \quad (9b)$$

and

$$(\psi^1)'(t) = 4\psi^0(t), \quad (10a)$$

$$\left(\frac{1}{4}\tilde{\psi}^0\right)'(t) = -\tilde{\psi}^1(t). \quad (10b)$$

We can see Eqs. (10a, 10b) is a specific form of Eqs. (7a, 7b). Well-known pairs of wavelets are $\text{cdf}(n+1).m$ as (φ^1, ψ^1) and $\text{cdf}n.(m+1)$ as (φ^0, ψ^0) . For example, the pair of $\text{cdf}3.5$ and $\text{cdf}2.6$ has been used in [RLH*17]. However, B-spline wavelets are not suitable for reconstruction as discussed later. So, we explore to construct new wavelets using Eqs. (10a, 10b).

Since ψ^1 or $\tilde{\psi}^1$ often has no analytical representation, it is difficult to construct ψ^0 and $\tilde{\psi}^0$ by directly taking the derivative of ψ^1 and integral of $\tilde{\psi}^1$ as described in C1. Another way is to explicitly

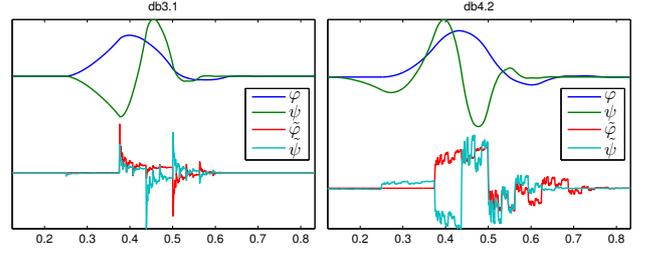


Figure 2: Plot of $\text{db}(m+1).(m-1)$. We plot the primal and dual scaling and wavelet functions of $\text{db}3.1$ (left) and $\text{db}4.2$ (right).

compute their function values using numerical differential and integral methods. However, the constructed wavelets are unfriendly for taking the fast wavelet transform.

Here, we introduce an algorithm to computing the filters of new wavelets with the help of Eqs. (9a, 9b). The algorithm can be summarized in the following corollary from Lemarié-Rieusset's Proposition.

Corollary 1 Let $\{g_k^1\}$ and $\{\tilde{g}_k^1\}$ be the filters of compactly supported wavelets ψ^1 and $\tilde{\psi}^1$, respectively. Their smallest intervals are $[b_1, n_1]$ and $[\tilde{b}_1, \tilde{n}_1]$. The filters $\{g_k^0\}$ and $\{\tilde{g}_k^0\}$ of new wavelets ψ^0 and $\tilde{\psi}^0$ can be respectively computed by

$$\mathbf{g}^0 = \frac{1}{2}\mathbf{D} \cdot \mathbf{g}^1, \quad (11a)$$

$$\tilde{\mathbf{g}}^0 = -2\tilde{\mathbf{D}}^{-1} \cdot \tilde{\mathbf{g}}^1, \quad (11b)$$

where \mathbf{g}^1 denotes a vector that stores $\{g_k^1\}$ for $k \in [b_1 - 1, n_1 + 2]$, and similarly for $\tilde{\mathbf{g}}^1$, \mathbf{g}^0 and $\tilde{\mathbf{g}}^0$. \mathbf{D} is a difference matrix defined by

$$\mathbf{D} = \begin{bmatrix} 1 & & & & \\ -1 & 1 & & & \\ & \ddots & \ddots & \ddots & \\ & & & -1 & 1 \\ & & & & -1 \end{bmatrix}_{N_1 \times N_2} \quad (12)$$

with $N_1 = N_2 + 1$ and $N_2 = n_1 + 2$. $\tilde{\mathbf{D}}$ is also a different matrix with $N_2 = \tilde{n}_1 + 1$ and $\tilde{\mathbf{D}}^{-1}$ is a pseudo-inverse of $\tilde{\mathbf{D}}$.

Proof: We leave the proof to Appendix B.

In practice, we are often given the filters $\{h_k^1, g_k^1\}$ of scaling and wavelet functions (φ^1, ψ^1) and need to know $\{h_k^0, g_k^0\}$ of new ones. In this case, we use Eq. (5) to get their dual filters and vice versa.

Notation of wavelets. By convention, we refer a wavelet type by its wavelet name with two numbers, such as $\text{cdf}2.6$, $\text{cdf}3.5$. The two numbers are the count of vanishing moments (VMs) of the dual wavelet $\tilde{\psi}$ and primal one ψ , respectively. In case of the numbers being equal, one number is used, such as $\text{db}1$ ($\text{db}1.1$), $\text{db}2$ ($\text{db}2.2$).

A new family of wavelets. Using the corollary, we construct a new family of wavelets named $\text{db}(m-1).(m+1)$ from $\text{db}m$, such as $\text{db}0.2$ from $\text{db}1$, $\text{db}1.3$ from $\text{db}2$, and $\text{db}2.4$ from $\text{db}3$. The VM numbers $(m-1).(m+1)$ follow from that ψ^0 computed by Eq. (10a) should have one more VM than ψ^1 , and $\tilde{\psi}^0$ by Eq. (10b) has one less VM than $\tilde{\psi}^1$. Based on C2, $\text{db}m$ can be seen as new wavelets constructed from $\text{db}(m+1).(m-1)$.

In the work [MPS08], $\text{db}m$ was used for reconstruction. We find that $\text{db}(m+1).(m-1)$ is better than $\text{db}m$ since it has one more degree of continuity than $\text{db}m$ while preserving almost the same support size. This is because, by the integral Eq. (10b), the dual wavelet

of $\text{db}(m-1).(m+1)$ has one more degree of continuity than that of $\text{db}m$, and the dual and primal wavelets of $\text{db}m$ are the same. We will give detail comparisons of these two kinds of wavelets in Sec. 6.

Alternately, B-spline wavelets $\text{cdf}(n+1).m$ (when $n \geq 2$) can also be used for reconstructions since they are smooth and have been successfully used for image compositing in [RLH*17] and other applications in computer graphics. We find that $\text{db}(m+1).(m-1)$ also produce better reconstructed surfaces than $\text{cdf}(n+1).m$ when both of them have the same support size. As an example, Fig. 6(c) and (e) show the results for the horse dataset reconstructed using $\text{cdf}3.1$ and $\text{db}3.1$, respectively.

Fig. 2 shows the plots of $\text{db}3.1$ and $\text{db}4.2$. We can see the primal scaling and wavelet functions of $\text{db}3.1$ are smooth. As an example, we list the values of the primal and dual filters of $\text{db}3.1$ in Table 1.

5. Implementation

In this section, we describe our complete algorithm for constructing surface from input point set $\{p_i, \mathbf{n}_i\}$, which includes three steps: (1) pre-processing the point set, i.e. computing a smoothed normal field \mathbf{u} from $\{p_i, \mathbf{n}_i\}$. (2) computing the coefficients of $\tilde{\chi}_M$ by the formula (6d) using a particular wavelet base; (3) returning the reconstructed surface by extracting an appropriate level set of $\tilde{\chi}_M$.

For the first and third steps, we follow the method in [KBH06] to estimate a normal field \mathbf{u} from $\{p_i, \mathbf{n}_i\}$ and use their weight average formula to select an iso-value. We use the dual marching cube method [SW04] to get the final triangle mesh. We will introduce the computation of coefficients and its implementation on octrees.

Computation of coefficients. First, we write the wavelet representation of $\tilde{\chi}_M$ using 3D wavelet bases

$$\tilde{\chi}_M(\mathbf{x}) = \sum_{\mathbf{e}} \sum_{\mathbf{j}_{\mathbf{e}}} \sum_{\mathbf{k}} c_{\mathbf{j}_{\mathbf{e}}, \mathbf{k}} \Psi_{\mathbf{j}_{\mathbf{e}}, \mathbf{k}}^1(\mathbf{x}), \quad (13)$$

where $\Psi_{\mathbf{j}_{\mathbf{e}}, \mathbf{k}}^1$ with $\mathbf{e} \in \{0, 1\}^3$ are the eight types of wavelets generated by 3D tensor-products of 1D basis φ^1 and ψ^1 , and $c_{\mathbf{j}_{\mathbf{e}}, \mathbf{k}}$ are the corresponding coefficients we need to compute. Then, we compute the wavelet representation of the smoothed normal field \mathbf{u} .

$$\mathbf{u} = \sum_{\mathbf{e}} \sum_{\mathbf{j}_{\mathbf{e}}} \sum_{\mathbf{k}} \tilde{\mathbf{u}}_{\mathbf{j}_{\mathbf{e}}, \mathbf{k}} \Psi_{\mathbf{j}_{\mathbf{e}}, \mathbf{k}}(\mathbf{x}), \quad (14)$$

where $\Psi_{\mathbf{j}_{\mathbf{e}}, \mathbf{k}}$ with $\mathbf{e} \in \{0, 1\}^3$ are the eight types of vector-valued wavelets derived from the gradient of $\Psi_{\mathbf{j}_{\mathbf{e}}, \mathbf{k}}^1$, and $\tilde{\mathbf{u}}_{\mathbf{j}_{\mathbf{e}}, \mathbf{k}}$ are the wavelet coefficients computed by taking the wavelet transform of \mathbf{u} . Finally, we compute $c_{\mathbf{j}_{\mathbf{e}}, \mathbf{k}}$ from $\tilde{\mathbf{u}}$ by Eqs. (6d, 8)

$$c_{\mathbf{j}_{\mathbf{e}}, \mathbf{k}} = \frac{\langle \mathbf{w}_{\mathbf{j}_{\mathbf{e}}}, \tilde{\mathbf{u}}_{\mathbf{j}_{\mathbf{e}}, \mathbf{k}} \rangle}{\|\mathbf{w}_{\mathbf{j}_{\mathbf{e}}}\|^2}, \quad (15)$$

where $\mathbf{w}_{\mathbf{j}_{\mathbf{e}}}$ with $\mathbf{e} \in \{0, 1\}^3 \setminus [0, 0, 0]$ are the coefficients of $\nabla \Psi_{\mathbf{j}_{\mathbf{e}}, \mathbf{k}}^1$.

We leave the definitions of $\Psi_{\mathbf{j}_{\mathbf{e}}, \mathbf{k}}^1$, $\Psi_{\mathbf{j}_{\mathbf{e}}, \mathbf{k}}$ and $\mathbf{w}_{\mathbf{j}_{\mathbf{e}}}$, and the derivations of Equations (13–15) to Appendix C. Our final Eq. (15) for computing wavelet coefficients is a 3D extension of the 2D one in [RLH*17], where they used it for gradient-domain image compositing.

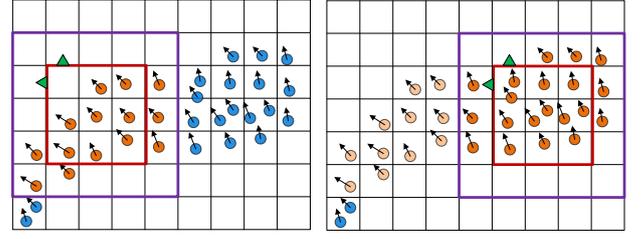


Figure 3: The wavelet transform of the normal field on small grids. Left: After reading a subset of oriented points (blue+orange oriented circles), we cluster the points in the purple grid and splat each normal (orange circles) into the orange grid to get a normal field (green arrows) stored on the facets of the grid. Then, we continue the same process for another grid (Right). Finally, we perform the wavelet transform on each orange grid.

Octree-Based Implementation. Our octree-based implementation of computing coefficients is similar to the one in [MPS08]. However, different from their method which computes point-wise sums of normals \mathbf{n}_i by the formula (2) on each octree node, we perform the wavelet transform of the smoothed normal field \mathbf{u} on each node. If we also perform point-wise wavelet transforms of each normal \mathbf{n}_i , the cost would be high since every \mathbf{n}_i would produce multiple normals due to the convolution process in Step 1. We present a strategy to reduce the cost considerably.

Our strategy is to read a subset of oriented points each time, and then cluster the points and splat their normals into non-overlapped grids of small size. Finally, we perform the wavelet transform of the normal field on each grid and write the wavelet coefficients extracted by Eq. (15) to the corresponding octree nodes. Fig. 3 illustrates this process in 2D. The grid width is the maximum of the size of wavelet filters used and the width of the smoothing filter in Step 1.

6. Experimental Results

We first discuss the reconstruction quality of our method using various wavelets. Then, we compare our method with the Streaming Wavelet reconstruction of Manson et al. [MPS08] (*SWavelet*), the Poisson reconstruction of Kazhdan et al. [KBH06] (*Poisson*), the Smoothed Signed Distance reconstruction of Calakli and Taubin [CT11] (*SSD*) and the Screened Poisson reconstruction of Kazhdan Hoppe [KH13] (*SPoisson*). For *SPoisson* and *SSD*, we use their latest implementations of Kazhdan and Hoppe [KH18] and their default settings. For *Poisson*, we use *SPoisson* by setting the value of the point weight to 0 and leave other settings unchanged. For *SWavelet*, we use the implementation by the authors and turn on the blurring setting. We use three different types of datasets including the benchmark of Berger et al. [BLN*13], real scanner datasets and highly nonuniformly sampled synthetic datasets. The benchmark has five models: Anchor, Dancing Children, Daratech, Gargoyle, and Quasimodo. We present the reconstruction results in terms of speed, memory, efficiency, and accuracy.

Comparisons of various wavelets. We compare the reconstruction quality of our method with various wavelets. The comparison is two-fold. First, we compare constructed $\text{db}(m+1).(m-1)$ to $\text{db}m$. Our results show that $\text{db}(m+1).(m-1)$ produce better and

smoother results than dbm . Second, using the same wavelets dbm , we compare our method to SWavelet. Thanks to our general formula (6d), our method can produce cleaner results than SWavelet.

As an example, Fig. 4 shows the reconstruction results of the first test of the Dancing Children model. Fig. (f) and (d) (resp. (e) and (b)) show that the surface of $db3.1$ (resp. $db2.0$) is smoother than that of $db2.2$ (resp. $db1.1$). Fig. (a) and (b) for $db1.1$, and Fig. (c) and (d) for $db2.2$ show that our method produces cleaner surfaces than SWavelet, and are more robust to handle the incomplete data.

Comparing Fig. 4(e) and (f), $db3.1$ produces smoother surfaces than $db2.0$, and has a better ability to repair the surfaces where samples are missing (see the abdomen and legs of the second child). On the other hand, since $db2.0$ has only half support size of $db3.1$, it is very suitable when reconstruction speed is more important than quality. Overall, $db(m+1).(m-1)$ provides us a good tradeoff between quality and speed by selecting the appropriate wavelet basis.

Comparisons of various methods. We compare our method to SWavelet, Poisson, SSD and SPoisson on the three types of datasets. Fig. 5 shows the reconstructions of the real scanned Eagle dataset using the five methods at depth 10. The data is noisy and incomplete. We can see that the results of SWavelet and our method are sharper than that by Poisson. Our method has a better ability to predict the surface under the neck than SWavelet. SPoisson and SSD produce sharper results than ours and SWavelet. Fig. 8 shows the results of the last test of the Daratech model at depth 9. This dataset is highly noisy. SPoisson produces the sharpest result. However, it over-fits the points leading to noise. Poisson tends to over-smooth the data. Both SSD and our method produce smooth surfaces and yet preserve the features. Fig. 6 shows the results of the horse dataset at depth 9. The dataset is obtained by sampling a virtual horse model with a sampling density proportional to curvature, and thus highly non-uniform. For this challenging dataset, SWavelet fails to produce the surface accurately. Poisson, SPoisson, SSD and our method can accurately reconstruct the surface.

To evaluate the numerical accuracy of the five methods, we use the benchmark of Berger et al. [BLN^{*}13], which simulates scanner errors such as noise, nonuniform sampling, and misalignment. As an example, Fig. 7 visualizes the errors in the reconstructions of the first test of the Dancing Children model as shown in Fig. 4 and the last test of the Daratech model as shown in Fig. 8. More generally, Fig. 9 gives the mean distance and angle error distributions of all the tests of the models of the benchmark for the five methods. Our method significantly improves the accuracy of SWavelet and outperforms Poisson. SPoisson and SSD are the two best ones among the five methods because both methods incorporate position constraints, and SSD additionally incorporates Hessian constraints. This suggests the importance of adding these constraints to the wavelet framework.

Performance. In Table 2, we list the time and memory of the five methods. SWavelet is the fastest and uses the least memory among the five methods. Compared to SWavelet, our method uses 2 times of memory and 4 times of time because we need more octree nodes to store the splatted normals. We can see that the implementation of SSD using the Adaptive Multigrid by Kazhdan and Hoppe [KH18] is more efficient than the original one in [CT11]. Compared to Poisson, SPoisson and SSD, our method has similar memory and

time cost. However, different from those methods which can be seen as variational implicit methods [SSW14], our method can be seen as an analytic implicit method since it explicitly gives analytic solutions for the problem of surface reconstruction. While our method improves upon SWavelet, to the best of our knowledge, it is the first analytic implicit method that can produce robust results comparable to variational implicit methods like Poisson in terms of both accuracy and efficiency. On the other side, it inherits all the features of SWavelet such as friendly streaming implementation, and thus can be implemented in a friendly streaming way to process very large datasets.

7. Conclusion and Future Work

In this paper, we have introduced a new biorthogonal wavelet approach for the problem of reconstructing a water-tight surface from a finite set of oriented points regardless of the input data being noisy, missing or nonuniform. Our method fundamentally addresses the problems with previous wavelet methods and improves the accuracy significantly. The key ingredients are a general formula to computing wavelet coefficients and a simple method for constructing smoother wavelets. With our method, we construct a family of smooth biorthogonal wavelets called $db(m+1).(m-1)$. Our wavelets have almost the same support size as dbm while producing smooth reconstructed surfaces. Since the wide application of wavelet theory, we believe the method can be used in other fields.

While our method improves the accuracy of SWavelet considerably, one limitation is that it needs extra tree nodes to store the splatted normals in the pre-processing step. This leads to an increase in memory and computational time. In general, the total number of nodes is 2 times more than that of SWavelet. This number would be higher if the input sample was extremely non-uniform. Considering that the final extracted surface is embedded in a narrow band of nodes, however, we believe that a large number of nodes could be trimmed without sacrificing surface quality.

There are two ways to extend our work. First, although we have given the conditions for the unique construction of \tilde{F}_k for the 1D case, i.e. the dual basis of the derivative of B_k^1 , we have not yet found similar conditions for the 3D case. However, we know that the best \tilde{F}_k should be the dual basis of ∇B_k^1 . Thus, our goal is to find a better B_k such that \tilde{F}_k constructed by Eq. (8) is as close to the dual basis of ∇B_k^1 as possible while preserving relatively small support size. Second, the high accuracy of the Screened Poisson and SSD reconstruction methods strongly suggest that the incorporations of position and Hessian constraints would improve the reconstruction accuracy. Thus, our second extension is to incorporate these constraints to the wavelet framework.

Acknowledgements

The authors thank the anonymous reviewers for their constructive feedback. This work is supported by the National Key Research and Development Program of China (2017YFB1002701), Macao Science and Technology Development Fund (FDCT: 068/2015/A2, 136/2014/A3, 015/2015/AMJ, 121/2016/A3), NSFC (61632003, 61672502, 6187070657, 61472261, 61572316), Univ. of Macau

db3.1	-2	-1	0	1	2	3
h	0.241481456572267	0.659739608441171	0.530330085889911	0.047367172745377	-0.064704761275630	0
g	0	0	-0.258819045102521	-0.707106781186548	0.965925826289068	0
\tilde{h}	0	0.965925826289068	0.707106781186548	-0.258819045102521	0	0
\tilde{g}	0	0.064704761275630	0.047367172745377	-0.530330085889911	0.659739608441171	-0.241481456572267

Table 1: The values of the primal and dual filters of db3.1.

Model	Depth	Time (s)				Memory (MB)				Vertices $\times 10^6$			
		SWavelet	Ours	SPoisson	SSD	SWavelet	Ours	SPoisson	SSD	SWavelet	Ours	SPoisson	SSD
bunny	9	12	46	54	40	199	400	461	460	0.7	0.7	0.7	0.7
eagle	10	33	140	134	115	548	1190	811	1010	1.6	2.0	1.7	1.7
Daratech	9	5	23	23	20	97	177	482	516	0.3	0.3	0.3	0.3
Dancing Chidren	9	8	32	28	26	156	236	245	357	0.5	0.5	0.2	0.2

Table 2: The time and memory of SWavelet (db2), SPoisson, SSD and our method (db3.1). Poisson has almost the same time and memory as SPoisson.

Research Fund (MYRG2016-00160-FST, MYRG2018-00248-FST), and Natural Science Foundation of Guangdong Province (2016A030310342).

References

- [ACA07] ALLÈGRE R., CHAINE R., AKKOUCHE S.: A streaming algorithm for surface reconstruction. In *Proceedings of the Fifth Eurographics Symposium on Geometry Processing* (2007), SGP '07, Eurographics Association, pp. 79–88. 2
- [ACK01] AMENTA N., CHOI S., KOLLURI R. K.: The power crust. In *Proceedings of the Sixth ACM Symposium on Solid Modeling and Applications* (2001), SMA '01, ACM, pp. 249–266. 2
- [BBX95] BAJAJ C. L., BERNARDINI F., XU G.: Automatic reconstruction of surfaces and scalar fields from 3d scans. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques* (1995), SIGGRAPH '95, ACM, pp. 109–118. 2
- [BKBH07] BOLITHO M., KAZHDAN M., BURNS R., HOPPE H.: Multilevel streaming for out-of-core surface reconstruction. In *Proceedings of the Fifth Eurographics Symposium on Geometry Processing* (2007), SGP '07, Eurographics Association, pp. 69–78. 2
- [BKBH09] BOLITHO M., KAZHDAN M., BURNS R., HOPPE H.: Parallel poisson surface reconstruction. In *Proceedings of the 5th International Symposium on Advances in Visual Computing: Part I* (2009), ISVC '09, Springer-Verlag, pp. 678–689. 2
- [BL17] BOLTCHEVA D., LÉVY B.: Surface reconstruction by computing restricted voronoi cells in parallel. *Computer-Aided Design* 90 (2017), 123–134. SI:SPM2017. 2
- [BLN*13] BERGER M., LEVINE J. A., NONATO L. G., TAUBIN G., SILVA C. T.: A benchmark for surface reconstruction. *ACM Trans. Graph.* 32, 2 (Apr. 2013), 20:1–20:17. 2, 6, 7, 11
- [BMR*99] BERNARDINI F., MITTLEMAN J., RUSHMEIER H., SILVA C., TAUBIN G.: The ball-pivoting algorithm for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics* 5, 4 (Oct. 1999), 349–359. 2
- [BTS*17] BERGER M., TAGLIASACCHI A., SEVERSKY L. M., ALLIEZ P., GUENNEBAUD G., LEVINE J. A., SHARF A., SILVA C. T.: A survey of surface reconstruction from point clouds. *Comput. Graph. Forum* 36, 1 (Jan. 2017), 301–329. 2
- [CBC*01] CARR J. C., BEATSON R. K., CHERRIE J. B., MITCHELL T. J., FRIGHT W. R., MCCALLUM B. C., EVANS T. R.: Reconstruction and representation of 3d objects with radial basis functions. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques* (2001), SIGGRAPH '01, ACM, pp. 67–76. 2
- [CDF92] COHEN A., DAUBECHIES I., FEAUVEAU J.-C.: Biorthogonal bases of compactly supported wavelets. *Communications in Pure and Applied Mathematics* 45, 5 (1992), 485–560. 1
- [CG06] CAZALS F., GIESEN J.: Delaunay triangulation based surface reconstruction: Ideas and algorithms. In *EFFECTIVE COMPUTATIONAL GEOMETRY FOR CURVES AND SURFACES* (2006), Springer, p. p. 231–273. 2
- [CL96] CURLESS B., LEVOY M.: A volumetric method for building complex models from range images. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques* (1996), SIGGRAPH '96, ACM, pp. 303–312. 2
- [CT11] CALAKLI F., TAUBIN G.: Ssd: Smooth signed distance surface reconstruction. *Computer Graphics Forum* 30, 7 (2011), 1993–2002. 1, 2, 6, 7
- [Dau92] DAUBECHIES I.: *Ten Lectures on Wavelets*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1992. 1
- [Dey06] DEY T. K.: *Curve and Surface Reconstruction: Algorithms with Mathematical Analysis (Cambridge Monographs on Applied and Computational Mathematics)*. Cambridge University Press, New York, NY, USA, 2006. 2
- [DG06] DEY T. K., GOSWAMI S.: Provable surface reconstruction from noisy samples. *Comput. Geom. Theory Appl.* 35, 1 (Aug. 2006), 124–141. 2
- [DGH01] DEY T. K., GIESEN J., HUDSON J.: Delaunay based shape reconstruction from large data. In *Proceedings of the IEEE 2001 Symposium on Parallel and Large-data Visualization and Graphics* (Piscataway, NJ, USA, 2001), PVG '01, IEEE Press, pp. 19–27. 2
- [DP09] DERIAZ E., PERRIER V.: Orthogonal helmholtz decomposition in arbitrary dimension using divergence-free and curl-free wavelets. *Applied and Computational Harmonic Analysis* 26, 2 (Mar. 2009), 249–269. 5
- [ESS16] ESTELLERS V., SCOTT M. A., SOATTO S.: Robust surface reconstruction. *SIAM Journal on Imaging Sciences* 9, 4 (2016), 2073–2098. 2
- [FG11] FUHRMANN S., GOESELE M.: Fusion of depth maps with multiple scales. *ACM Trans. Graph.* 30, 6 (Dec. 2011), 148:1–148:8. 2
- [FG14] FUHRMANN S., GOESELE M.: Floating scale surface reconstruction. *ACM Trans. Graph.* 33, 4 (July 2014), 46:1–46:11. 2
- [HA17] HORACEK J., ALIM U.: Compactly supported biorthogonal wavelet bases on the body centered cubic lattice. *Comput. Graph. Forum* 36, 3 (June 2017), 35–45. 2
- [HDD*92] HOPPE H., DE ROSE T., DUCHAMP T., McDONALD J., STUETZLE W.: Surface reconstruction from unorganized points. In *Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques* (1992), SIGGRAPH '92, ACM, pp. 71–78. 2
- [HK06] HORNING A., KOBELT L.: Robust reconstruction of watertight 3d models from non-uniformly sampled point clouds without normal information. In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing* (2006), SGP '06, Eurographics Association, p. p. 41–50. 2

- [HLL*18] HU L., LI Q., LIU S., LIU X., WANG Z.: Wavelet-based polygon soup consolidation. *Computers & Graphics* 70 (2018), 39–50. CAD/Graphics 2017. 2
- [Kai11] KAISER G.: *A friendly guide to wavelets*. Birkhäuser, Basel, 2011. 3
- [Kaz05] KAZHDAN M.: Reconstruction of solid models from oriented point sets. In *Proceedings of the Third Eurographics Symposium on Geometry Processing* (2005), SGP '05. 1, 2, 3
- [KBH06] KAZHDAN M., BOLITHO M., HOPPE H.: Poisson surface reconstruction. In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing* (2006), SGP '06, pp. 61–70. 1, 2, 3, 6
- [KH13] KAZHDAN M., HOPPE H.: Screened poisson surface reconstruction. *ACM Trans. Graph.* 32, 3 (July 2013), 29:1–29:13. 1, 2, 6
- [KH18] KAZHDAN M., HOPPE H.: An adaptive multigrid solver for applications in computer graphics. *Computer Graphics Forum* 0, 0 (2018). 6, 7
- [KSO04] KOLLURI R., SHEWCHUK J. R., O'BRIEN J. F.: Spectral surface reconstruction from noisy point clouds. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing* (2004), SGP '04, ACM, pp. 11–21. 2
- [LPK09] LABATUT P., PONS. J., KERIVEN R.: Robust and efficient surface reconstruction from range data. *Computer Graphics Forum* 28 (2009). 2
- [LR92] LEMARIÉ-RIEUSSET P. G.: Analyses multi-résolutions non orthogonales, commutation entre projecteurs et dérivation et ondelettes vecteurs à divergence nulle. *Rev. Mat. Iberoamericana* 8, 2 (Mar. 1992), 221–237. 2, 5
- [LSYD17] LIU Y., SONG Y., YANG Z., DENG J.: Implicit surface reconstruction with total variation regularization. *Computer Aided Geometric Design* 52-53 (2017), 135–153. Geometric Modeling and Processing 2017. 2
- [Mal08] MALLAT S.: *A Wavelet Tour of Signal Processing, Third Edition: The Sparse Way*, 3rd ed. Academic Press, 2008. 3
- [MPS08] MANSON J., PETROVA G., SCHAEFER S.: Streaming surface reconstruction using wavelets. In *Proceedings of the Symposium on Geometry Processing* (2008), SGP '08, Eurographics Association, p. 1411–1420. 1, 2, 3, 4, 5, 6
- [NOS09] NAGAI Y., OHTAKE Y., SUZUKI H.: Smoothing of partition of unity implicit surfaces for noise robust surface reconstruction. In *Proceedings of the Symposium on Geometry Processing* (2009), SGP '09, Eurographics Association, pp. 1339–1348. 2
- [OBA*03] OHTAKE Y., BELYAEV A., ALEXA M., TURK G., SEIDEL H.-P.: Multi-level partition of unity implicits. *ACM Trans. Graph.* 22, 3 (July 2003), 463–470. 2
- [OBA05] OHTAKE Y., BELYAEV A., ALEXA M.: Sparse low-degree implicit surfaces with applications to high quality rendering, feature extraction, and smoothing. In *Proceedings of the Third Eurographics Symposium on Geometry Processing* (2005), SGP '05, Eurographics Association. 2
- [PTC16] PAN M., TONG W., CHEN F.: Compact implicit surface reconstruction via low-rank tensor approximation. *Comput. Aided Des.* 78, C (Sept. 2016), 158–167. 2
- [RLH*17] REN X., LYU L., HE X., ZHANG Y., WU E.: Efficient gradient-domain compositing using an approximate curl-free wavelet projection. *Computer Graphics Forum* 36, 7 (2017), 207–215. 2, 5, 6
- [SBS07] SCHALL O., BELYAEV A., SEIDEL H.-P.: Error-guided adaptive fourier-based surface reconstruction. *Computer-Aided Design* 39, 5 (2007), 421–426. Geometric Modeling and Processing 2006. 2
- [SSW14] SCHROERS C., SETZER S., WEICKERT J.: A variational taxonomy for surface reconstruction from oriented points. In *Proceedings of the Symposium on Geometry Processing* (2014), SGP '14, Eurographics Association, pp. 195–204. 2, 7
- [STJ*17] SCHERTLER N., TARINI M., JAKOB W., KAZHDAN M., GUMHOLD S., PANOZZO D.: Field-aligned online surface reconstruction. *ACM Trans. Graph.* 36, 4 (July 2017), 77:1–77:13. 2
- [SW04] SCHAEFER S., WARREN J.: Dual marching cubes: Primal contouring of dual grids. In *Proceedings of the Computer Graphics and Applications, 12th Pacific Conference* (Washington, DC, USA, 2004), PG '04, IEEE Computer Society, pp. 70–76. 6
- [TF18] TANG Y., FENG J.: Multi-scale surface reconstruction based on a curvature-adaptive signed distance field. *Computers & Graphics* 70 (2018), 28–38. CAD/Graphics 2017. 2
- [TKRL11] TAMAL K. D., RAMSAY D., LEI W.: Localized cocone surface reconstruction. *Computers & Graphics* 35, 3 (2011), 483–491. Shape Modeling International (SMI) Conference 2011. 2
- [UB17] UMMENHOFER B., BROX T.: Global, dense multiscale reconstruction for a billion points. *Int. J. Comput. Vision* 125, 1-3 (Dec. 2017), 82–94. 2
- [XZZ*14] XIONG S., ZHANG J., ZHENG J., CAI J., LIU L.: Robust surface reconstruction via dictionary learning. *ACM Trans. Graph.* 33, 6 (Nov. 2014), 201:1–201:12. 2
- [ZGHG11] ZHOU K., GONG M., HUANG X., GUO B.: Data-parallel octrees for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics* 17, 5 (May 2011), 669–681. 2

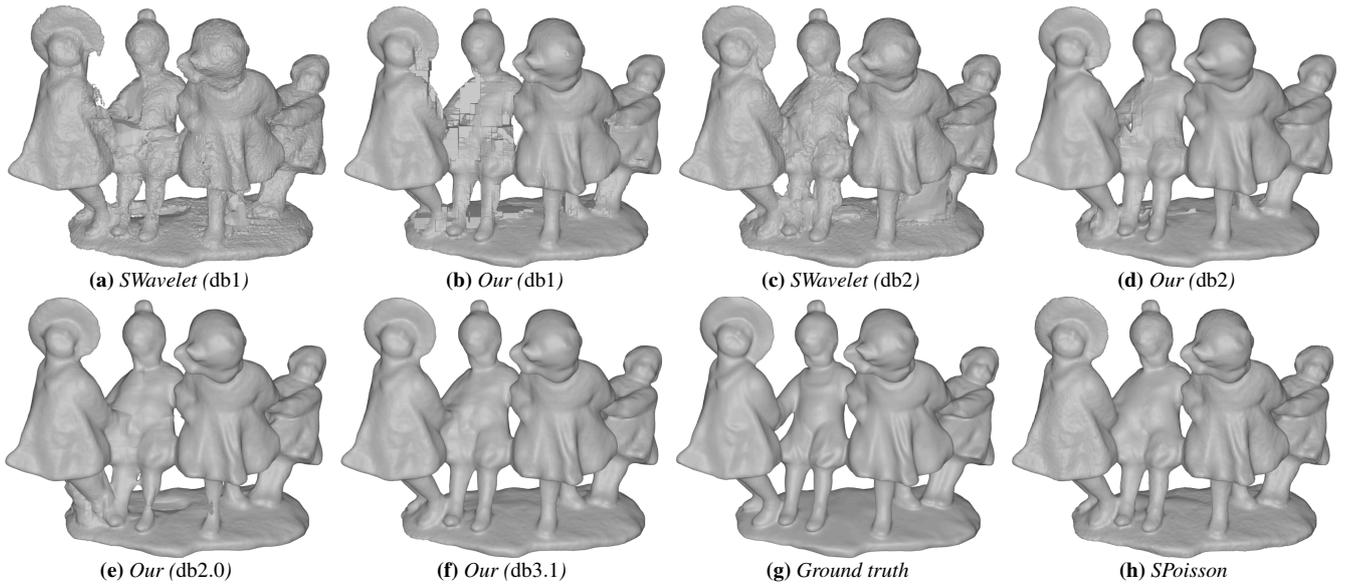


Figure 4: Comparisons of reconstructions of the first test of the Dancing Children model using SWavelet and our method with different types of wavelets. The result by Screened Poisson is for reference.

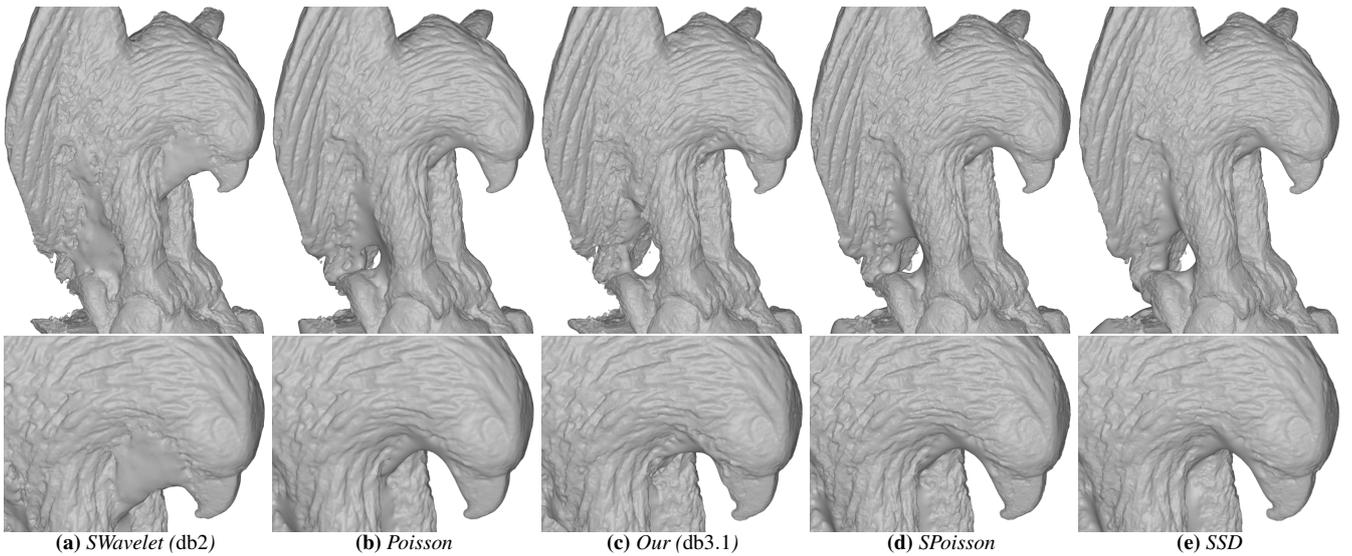


Figure 5: Comparisons of reconstructions of the noisy and incomplete Eagle dataset using SWavelet, Poisson, Screened Poisson, SSD and our method.

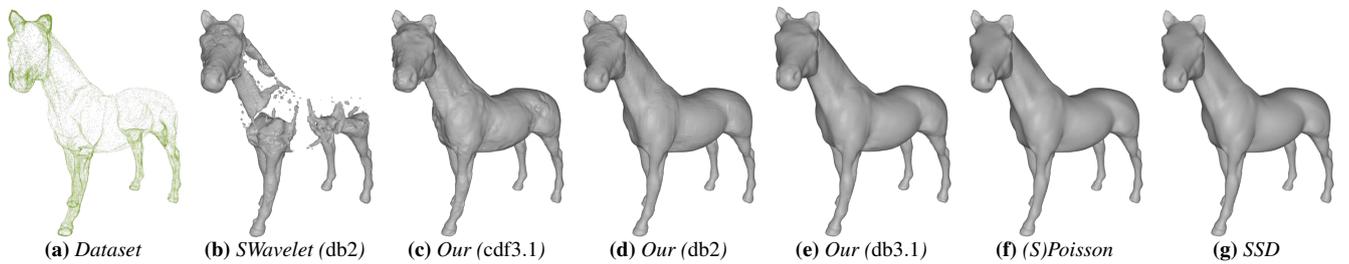


Figure 6: Comparisons of reconstructions of the highly non-uniformly sampled horse dataset using SWavelet, Poisson, Screened Poisson, SSD and our method.

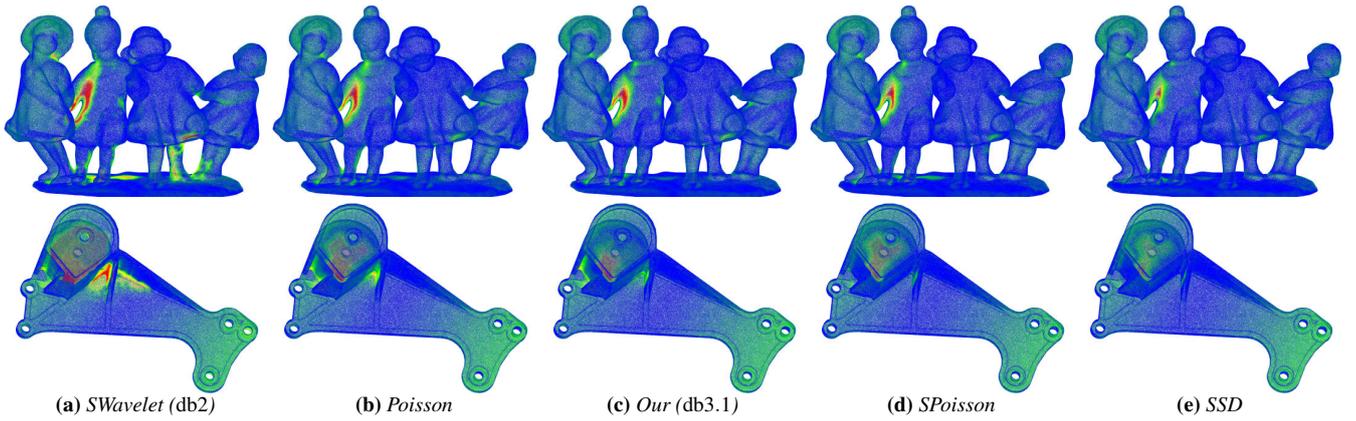


Figure 7: Visualization of errors for reconstructions of the Dancing Children model (top row) in Fig. 4 and the Daratech model (bottom row) in Fig. 8. The errors are the distance from the ground-truth reference samples to the reconstructed surfaces. Errors are visualized using a blue-green-yellow-red colormap, with blue corresponding to small error and red to large.

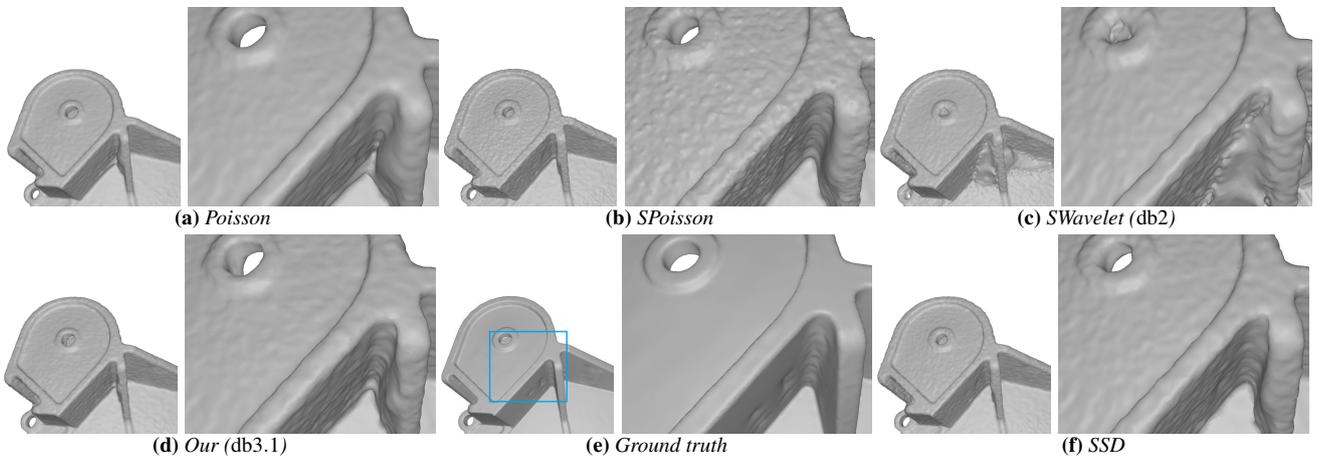


Figure 8: Comparisons of reconstructions of the last test (the samples are noisy and incomplete) of the Daratech model using SWavelet, Poisson, Screened Poisson, SSD and our method.

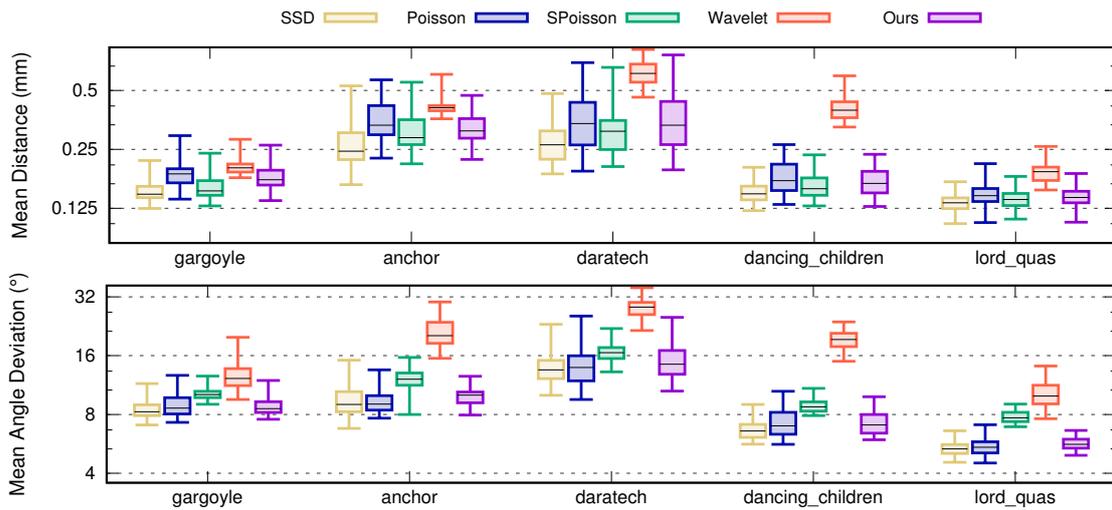


Figure 9: Numerical reconstruction accuracy of SWavelet (db2), Poisson, SSD, SPoisson and our method (db3.1) measured by the benchmark of Berger et al [BLN* 13].

Appendix A: Proof of Lemma 1

Proof: Let's first prove the "only if" part.

Taking the gradient of $\tilde{\chi}_M = \sum_i c_i B_i$ and substituting $\nabla \tilde{\chi}_M = \sum_i c_i \nabla B_i$ into (6d) and defining $\langle \mathbf{u}, \mathbf{v} \rangle = \int_D \mathbf{u} \cdot \mathbf{v} d\mathbf{x}$, we have

$$\begin{aligned} c_{\mathbf{k}} &= - \int_D \sum_i c_i \nabla B_i \cdot \tilde{\mathbf{F}}_{\mathbf{k}} d\mathbf{x} \\ &= \sum_i c_i \langle \nabla B_i, -\tilde{\mathbf{F}}_{\mathbf{k}} \rangle. \end{aligned} \quad (16a)$$

We can choose \mathbf{N} linearly independent functions from the subspace $\text{span}\{B_{\mathbf{k}}\}_{\mathbf{k} \in \{1, \dots, \mathbf{N}\}}$ and define the coefficient matrix \mathbf{C} of size \mathbf{N} with each column being the coefficients of each function. As the functions are linearly independent, \mathbf{C} is invertible. Since Eq. (16a) holds for any function in $\text{span}\{B_{\mathbf{k}}\}$, substituting the \mathbf{N} functions into it, we obtain an equation $\mathbf{C} = \mathbf{M} \cdot \mathbf{C}$, where $\mathbf{M} = [\langle \nabla B_i, -\tilde{\mathbf{F}}_{\mathbf{k}} \rangle]$ is a matrix whose (\mathbf{k}, i) entry is $\langle \nabla B_i, -\tilde{\mathbf{F}}_{\mathbf{k}} \rangle$. We can see that \mathbf{M} is an identity matrix of size \mathbf{N} . Since this holds for any large \mathbf{N} , we get $\langle \nabla B_i, -\tilde{\mathbf{F}}_{\mathbf{k}} \rangle = \delta_{i,\mathbf{k}}$. By Definition 2, $-\tilde{\mathbf{F}}_{\mathbf{k}}$ is a dual set of ∇B_i .

Now we prove the "if" part.

Given that $-\tilde{\mathbf{F}}_{\mathbf{k}}$ is a dual set of ∇B_i , we have $\langle \nabla B_i, -\tilde{\mathbf{F}}_{\mathbf{k}} \rangle = \delta_{i,\mathbf{k}}$. Substituting it into Eq. (16a), we get $c_{\mathbf{k}} = \sum_i c_i \delta_{i,\mathbf{k}} = c_{\mathbf{k}}$. \square

Appendix B: Proof of Corollary 1

Proof: We prove Eqs. (11a, 11b) one by one.

(1). By Eq. (3), substituting $\psi^1 = \sum_k g_k^1 \phi^1(2t - k)$ into Eq. (10a), we have

$$\begin{aligned} 4\psi^0(t) &= \left(\sum_k g_k^1 \phi^1(2t - k) \right)'(t) \\ &= 2 \sum_k g_k^1 (\phi^0(2t - k) - \phi^0(2t - k - 1)) \end{aligned} \quad (17a)$$

$$= 2 \sum_k (g_k^1 - g_{k-1}^1) \phi^0(2t - k) \quad (17b)$$

$$\begin{aligned} \Rightarrow 2[\phi^0(2t - k)] \mathbf{g}^0 &= [\phi^0(2t - k)] \mathbf{D} \cdot \mathbf{g}^1 \\ \Rightarrow \mathbf{g}^0 &= \frac{1}{2} \mathbf{D} \cdot \mathbf{g}^1 \end{aligned} \quad (17c)$$

Eq. (17a) follows from Eq. (9a). Rewriting Eq. (17b) in matrix form yields Eq. (17c), where $[\phi^0(2t - k)]$ denotes a matrix whose each column is $\phi^0(2t - k)$ for all $k \in [b_1 - 1, n_1 + 2]$ and \mathbf{D} is the difference matrix defined by Eq. (12). Finally, we get the result.

(2). By Eq. (4), substituting $\tilde{\psi}^0 = \sum_k \tilde{g}_k^0 \tilde{\phi}^0(2t - k)$ into Eq. (10b), we have

$$\begin{aligned} \sum_k \tilde{g}_k^0 (\tilde{\phi}^1(2t - k))' &= -4\tilde{\psi}^1(t) \\ 2 \sum_k \tilde{g}_k^0 (\tilde{\phi}^1(2t - k + 1) - \tilde{\phi}^1(2t - k)) &= -4\tilde{\psi}^1(t) \end{aligned} \quad (18a)$$

$$\begin{aligned} [\tilde{\phi}^1(2t - k)] \cdot \tilde{\mathbf{D}} \cdot \tilde{\mathbf{g}}^0 &= -2[\tilde{\phi}^1(2t - k)] \cdot \tilde{\mathbf{g}}^1 \\ \Rightarrow \tilde{\mathbf{g}}^0 &= -2\tilde{\mathbf{D}}^{-1} \cdot \tilde{\mathbf{g}}^1 \end{aligned} \quad (18b)$$

Eq. (18a) follows from Eq. (9b). Rewriting Eq. (18a) in matrix form yields Eq. (18b), where $[\tilde{\phi}^1(2t - k)]$ denotes a matrix whose each column is $\tilde{\phi}^1(2t - k)$ for all $k \in [\tilde{b}_1 - 1, \tilde{n}_1 + 2]$ and $\tilde{\mathbf{D}}$ is the difference matrix defined by Eq. (12). Finally, we get Eq. (11b) by a pseudo-inverse of $\tilde{\mathbf{D}}$. \square

Appendix C: Definition and Derivation in Sec. 5

(1) *Derivation of Eq. (13).* Let $\{\Phi_{0,0}^1, \Psi_{j,k}^1\}$ with $j \in \mathbb{Z}, k \in \mathbb{Z}$ be a 1D wavelet base. By 3D tensor-products of the base, we can define a 3D wavelet base with the following eight types of wavelets

$$\begin{aligned} \Psi_{j_{[0,0,0]}, \mathbf{0}}^1 &= \Phi_{0,0}^1 \Phi_{0,0}^1 \Phi_{0,0}^1, & \Psi_{j_{[1,0,0]}, \mathbf{k}}^1 &= \Psi_{j_1, k_1}^1 \Phi_{0,0}^1 \Phi_{0,0}^1, \\ \Psi_{j_{[0,1,0]}, \mathbf{k}}^1 &= \Phi_{0,0}^1 \Psi_{j_2, k_2}^1 \Phi_{0,0}^1, & \Psi_{j_{[0,0,1]}, \mathbf{k}}^1 &= \Phi_{0,0}^1 \Phi_{0,0}^1 \Psi_{j_3, k_3}^1, \\ \Psi_{j_{[1,1,0]}, \mathbf{k}}^1 &= \Psi_{j_1, k_1}^1 \Psi_{j_2, k_2}^1 \Phi_{0,0}^1, & \Psi_{j_{[0,1,1]}, \mathbf{k}}^1 &= \Phi_{0,0}^1 \Psi_{j_2, k_2}^1 \Psi_{j_3, k_3}^1, \\ \Psi_{j_{[1,0,1]}, \mathbf{k}}^1 &= \Psi_{j_1, k_1}^1 \Phi_{0,0}^1 \Psi_{j_3, k_3}^1, & \Psi_{j_{[1,1,1]}, \mathbf{k}}^1 &= \Psi_{j_1, k_1}^1 \Psi_{j_2, k_2}^1 \Psi_{j_3, k_3}^1. \end{aligned}$$

We can write the wavelet representation as Eq. (13) for any function in the space spanned by these wavelets.

(2) *Derivation of Eq. (14).* Taking the gradient of Eq. (13), we have

$$\nabla \tilde{\chi}_M(\mathbf{x}) = \sum_{\mathbf{e}} \sum_{\mathbf{j}_{\mathbf{e}}} \sum_{\mathbf{k}} c_{\mathbf{j}_{\mathbf{e}}, \mathbf{k}} \nabla \Psi_{\mathbf{j}_{\mathbf{e}}, \mathbf{k}}^1(\mathbf{x}).$$

We can rewrite $\nabla \Psi_{\mathbf{j}_{\mathbf{e}}, \mathbf{k}}^1(\mathbf{x}) = \mathbf{w}_{\mathbf{j}_{\mathbf{e}}} \Psi_{\mathbf{j}_{\mathbf{e}}, \mathbf{k}}$ as the component-wise vector product of its coefficient part $\mathbf{w}_{\mathbf{j}_{\mathbf{e}}}$ and its function part $\Psi_{\mathbf{j}_{\mathbf{e}}, \mathbf{k}}$. The coefficient vector $\mathbf{w}_{\mathbf{j}_{\mathbf{e}}} = 4\mathbf{e}^T [2^{j_1}, 2^{j_2}, 2^{j_3}]^T$ can be computed using Eqs. (9a, 10a) and the fact that $(\phi_{0,0}^1)'(t) = 0$ when periodic boundary condition is used. Let $\mathbf{u} = \nabla \tilde{\chi}_M$ and $\tilde{\mathbf{u}}_{\mathbf{j}_{\mathbf{e}}, \mathbf{k}} = c_{\mathbf{j}_{\mathbf{e}}, \mathbf{k}} \mathbf{w}_{\mathbf{j}_{\mathbf{e}}}$, the above equation can be rewritten as Eq. (14), i.e.

$$\mathbf{u} = \sum_{\mathbf{e}} \sum_{\mathbf{j}_{\mathbf{e}}} \sum_{\mathbf{k}} \tilde{\mathbf{u}}_{\mathbf{j}_{\mathbf{e}}, \mathbf{k}} \Psi_{\mathbf{j}_{\mathbf{e}}, \mathbf{k}}. \quad (19)$$

(3) *Derivation of Eq. (15).* Recalling the construction of $\tilde{\mathbf{F}}_{\mathbf{k}}$ in Eq. (8), we have

$$\tilde{\mathbf{F}}_{\mathbf{j}_{\mathbf{e}}, \mathbf{k}} = - \frac{\mathbf{w}_{\mathbf{j}_{\mathbf{e}}}}{\|\mathbf{w}_{\mathbf{j}_{\mathbf{e}}}\|^2} \tilde{\Psi}_{\mathbf{j}_{\mathbf{e}}, \mathbf{k}}, \quad (20)$$

where $\tilde{\Psi}_{\mathbf{j}_{\mathbf{e}}, \mathbf{k}}$ are the dual functions of $\Psi_{\mathbf{j}_{\mathbf{e}}, \mathbf{k}}$. Substituting Eqs. (19, 20) into Eq. (6d), we have

$$\begin{aligned} c_{\mathbf{j}_{\mathbf{e}}, \mathbf{k}} &= \int_D \sum_{\mathbf{e}} \sum_{\mathbf{j}_{\mathbf{e}}} \sum_{\mathbf{k}} \tilde{\mathbf{u}}_{\mathbf{j}_{\mathbf{e}}, \mathbf{k}} \Psi_{\mathbf{j}_{\mathbf{e}}, \mathbf{k}} \cdot \frac{\mathbf{w}_{\mathbf{j}_{\mathbf{e}}}}{\|\mathbf{w}_{\mathbf{j}_{\mathbf{e}}}\|^2} \tilde{\Psi}_{\mathbf{j}_{\mathbf{e}}, \mathbf{k}} d\mathbf{x} \\ &= \frac{\langle \mathbf{w}_{\mathbf{j}_{\mathbf{e}}}, \tilde{\mathbf{u}}_{\mathbf{j}_{\mathbf{e}}, \mathbf{k}} \rangle}{\|\mathbf{w}_{\mathbf{j}_{\mathbf{e}}}\|^2}. \end{aligned}$$