A Volumetric Approach to Point Cloud Compression—Part I: Attribute Compression

Philip A. Chou, *Fellow, IEEE*, Maxim Koroteev, *Member, IEEE*, and Maja Krivokuća

***Abstract*—Compression of point clouds has so far been confined to coding the positions of a discrete set of points in space and the attributes of those discrete points. We introduce an alternative approach based on volumetric functions that are functions defined not just on a finite set of points but throughout space. As in regression analysis, volumetric functions are continuous functions that are able to interpolate values on a finite set of points as linear combinations of continuous basis functions. Using a B-spline wavelet basis, we are able to code volumetric functions representing both geometry and attributes. Geometry compression is addressed in Part II of this paper, while attribute compression is addressed in Part I. Attributes are represented by a volumetric function whose coefficients can be regarded as a critically sampled orthonormal transform that generalizes the recent successful Region-Adaptive Hierarchical (or Haar) Transform to higher orders. Experimental results show that attribute compression using higher order volumetric functions is an improvement over the first-order functions used in the emerging MPEG point cloud compression standard.**

***Index Terms*—Bézier volumes, B-splines, wavelets, point cloud compression, color coding, attribute coding, multiresolution representations, graph signal processing, counting measure.**

I. Introduction

A *POINT cloud* is a set of *points* (also known as *locations* or *positions*) in Euclidean space, in which a vector of attributes (such as a color triple) may be associated with each point. Point clouds may be *static* or *dynamic*. A dynamic point cloud is a sequence of static point clouds, each in its own *frame*. Fig. 1 shows several examples of point clouds. Point clouds have applications in robotics, tele-operation, virtual and augmented reality, cultural heritage preservation, geographic information systems, and so forth.

Point clouds offer a natural representation of *volumetric media*, popularly known as *holograms*. Broadly interpreted, a hologram is an object or scene whose representation permits rendering arbitrary points of view, such that the object or scene appears to occupy space, due to stereo or motion parallax. Point clouds offer a natural representation of holograms, because each point, with one or more color attributes, can naturally represent the color of rays that pass through that point.

Fig. 1. Examples of point clouds: facades, cultural artifacts, buildings, cities, and people.

Volumetric media, or holograms, have emerged as the first significant new modality for immersive communication since the introduction in the late nineteenth century of audio for audio recordings and video for motion pictures. Like audio and video, volumetric media will be used in three major communication scenarios: on-demand consumption of pre-recorded content, broadcast of live or pre-recorded content, and interactive communication such as telephony or conferencing.

Manuscript received October 16, 2018; revised February 15, 2019; accepted March 12, 2019. Date of publication March 28, 2019; date of current version January 10, 2020. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Marta Mrak. *(Corresponding author: Philip A. Chou.)*

P. A. Chou is with Google Inc., Seattle, Seattle, WA 98103 USA (e-mail: pachou@ieee.org).

M. Koroteev is with Imatrex Inc., 121205 Moscow, Russia.

M. Krivokuća is with the Inria Centre de Recherche Rennes–Bretagne Atlantique, 35042 Rennes, France.

Digital Object Identifier 10.1109/TIP.2019.2908095

For these scenarios, storage and transmission are essential; hence data compression is essential.

Interest in point cloud compression has been demonstrated in recent publications, as well as in the establishment of a standardization activity within MPEG (see next section).

The two fundamental problems in point cloud compression are *geometry* compression and *attribute* compression. Geometry compression, which is sometimes called *shape* compression, is the problem of compressing the point locations; attribute compression is the problem of compressing the attribute values, given the point locations.

In this paper, we introduce a volumetric approach to point cloud compression, which can be used for both geometry and attribute compression. In our volumetric approach, both geometry and attributes are represented by volumetric functions. We use the term *volumetric function* to refer to a scalar or vector valued function defined on a volume of space (in contrast, say, to a function defined on an image plane or on a finite set of points). To create a compressed representation of the geometry or attributes, the appropriate volumetric functions are transformed, quantized, and entropy coded. Once decoded, the decoded volumetric functions are used to reconstruct the geometry and attributes.

To represent geometry, we use a scalar volumetric function and reconstruct the geometry as the level set of the decoded volumetric function. We address geometry compression in Part II of this paper [1].

To represent the attributes, we use a vector-valued volumetric function having the same dimension as the attributes, and we reconstruct the attributes as the values of the decoded volumetric function at the points of the decoded geometry. However, as in regression, the function continuously interpolates the points, enabling arbitrary zoomability as required in many applications. We determine the volumetric function by solving a linear regression. The parameters of the function are B-spline wavelet coefficients, equivalent to the parameters of linked Bézier volumes. The functions exist in a Hilbert space in which norm and orthogonality are induced by an inner product defined by a novel counting measure supported by the decoded point locations.

Part I of this paper is organized as follows. Section II outlines prior work. Section III establishes the framework. Section IV applies the framework to attribute compression. Section V provides experimental results, and Section VI concludes. Part II of the paper addresses geometry compression.

II. Prior Work

Much of the prior work on point cloud compression has been rolled up into the emerging MPEG Point Cloud Compression (PCC) standards [2]. PCC can therefore be used as a template for describing prior work in point cloud compression. PCC is divided into two profiles, one based on video coding (V-PCC) and one based on native 3D coding (G-PCC). The approach taken in the video-based profile is an adaptation of the ideas of geometry images [3] and geometry videos [4], [5], which were previously used for mesh compression. The idea is to flatten patches of colored geometry onto the plane and code them as video, in order to re-use our substantial investment in video coding. More relevant to the present paper is the native 3D profile of G-PCC. In this profile, geometry is coded using octrees, and attributes are coded using critically sampled spatial transforms that are adapted to the geometry of each point cloud. We now give references to prior work in those areas.

Octrees, or more precisely, Sparse Voxel Octrees (SVOs) were developed in the 1980s to represent the geometry of three-dimensional objects [6], [7]. In the guise of occupancy grids, they have also had significant use in robotics [8]–[10]. Octrees were first used for point cloud compression in [11]. They were further developed for progressive point cloud coding, including color attribute compression, in [12]. Octrees were extended to coding dynamic point clouds in [13] and implemented in the Point Cloud Library [14]. SVOs have been shown to have highly efficient implementations suitable for encoding at video frame rates [15], based on Morton codes [16]. Entropy coding for octrees has been addressed in [17]–[19]. In G-PCC, octrees are pruned at a certain depth, and then represent geometry within each leaf as a single point (i.e., voxel), a list of points, or a geometric model. The geometric model in G-PCC are triangulations called a triangle soup. Polygon soup representations of geometry for point cloud compression were first explored in [20], [21].

Point cloud attribute coding using critically sampled spatial transforms adapted to the geometry of each point cloud was first explored in [22] using the graph Fourier transform (GFT) [23] by considering point cloud attributes as a signal defined on the nodes in a discrete graph corresponding to points of the point cloud. The GFT was also used for compressing point cloud attributes in [24]–[26]. Point cloud attribute coding using the Karhunen-Loève transform (KLT) of a signal defined on points of a spatial random process was explored in [27], [28]. Both the GFT and KLT require solving an eigen-decomposition for each arrangement of points. Point cloud attribute coding using a wavelet based transform, the Region-Adaptive Haar Transform (RAHT), was proposed and investigated in [29], [30]. RAHT is a hierarchical transform whose hierarchy is represented by the same SVO used for geometry compression. RAHT and a related lifting-based transform are used in G-PCC, as described in [2].

Our work in the present paper builds on both octree coding of geometry (Part II of this paper) and RAHT coding of attributes (Part I of this paper). For geometry processing, instead of using triangulations as geometric models within each leaf of the octree, we use an implicit surface defined by the wavelet coefficients of a volumetric B-spline of order $p = 2$. Compared to earlier approaches to geometry coding, our approach guarantees hole-free reconstruction of the surface, regardless of the level of quantization, due to the continuity properties of the B-spline. For attribute processing, we show that RAHT has an interpretation as a volumetric B-spline of order $p = 1$, and then we proceed to generalize it to B-splines of orders $p \geq 2$. Compared to RAHT ($p = 1$), blocking artifacts with $p \geq 2$ are eliminated, again due to continuity properties of the B-spline of orders $p \geq 2$.

III. Framework

A. Measure

Our unique definition of measure is the foundation of our approach.

Let Ω be a set, and let $\sigma(\Omega)$ be a set of subsets of Ω such that $\sigma(\Omega)$ is a sigma-algebra, that is, closed under complementation and countably infinite unions. A *measure* is a function $\mu : \sigma(\Omega) \to \mathbb{R}$ that assigns a real number to each set in $\sigma(\Omega)$ such that the measure of the union of any sequence of disjoint subsets is the sum of measures of the subsets. Examples are the Lebesgue measure on the real line, the counting measure on the integers, and any probability measure on a probability space [31]–[33].

We focus on the case where $\Omega = \mathbb{R}^3$ and $\sigma(\mathbb{R}^3)$ is the Borel sigma algebra of $\mathbb{R}^3$. We suppose we are given a finite set of points in $\mathbb{R}^3$, say $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$. For each set $\mathcal{M} \in \sigma(\mathbb{R}^3)$, we define $\mu(\mathcal{M}) = |\mathcal{M} \cap \mathcal{X}|$ to be the number of such points in $\mathcal{M}$. This is an example of a counting measure, albeit unconventional.

Let $f : \mathbb{R}^3 \to \mathbb{R}$ be a real-valued function on $\mathbb{R}^3$. The integral of f over a set $\mathcal{M} \in \sigma(\mathbb{R}^3)$ with respect to measure μ is denoted $\int_{\mathcal{M}} f(\boldsymbol{x}) d\mu(\boldsymbol{x})$. When μ is the counting measure with respect to $\mathcal{X}$, the integral is equal to

$$\int_{\mathcal{M}} f(\boldsymbol{x}) d\mu(\boldsymbol{x}) = \sum_{\boldsymbol{x}_n \in \mathcal{M}} f(\boldsymbol{x}_n). \tag{1}$$

B. Hilbert Space

A Hilbert space is a complete normed vector space equipped with an inner product that induces the norm. Consider the Hilbert space $\mathcal{F}$ of real-valued functions $f : \mathbb{R}^3 \to \mathbb{R}$ equipped with inner product

$$\langle f, g \rangle = \int f(\boldsymbol{x}) g(\boldsymbol{x}) d\mu(\boldsymbol{x}) = \sum_n f(\boldsymbol{x}_n) g(\boldsymbol{x}_n), \tag{2}$$

where $f, g \in \mathcal{F}$, and $\|f\| \overset{\Delta}{=} \sqrt{\langle f, f \rangle}$ is the induced norm.[1]

With the inner product and norm so determined, other properties of the Hilbert space follow. Specifically, a vector $g \in \mathcal{F}$ is *orthogonal* to a vector $f \in \mathcal{F}$ iff $\langle f, g \rangle = 0$. A vector $g \in \mathcal{F}$ is *orthogonal* to a subspace $\mathcal{F}_0 \subseteq \mathcal{F}$ iff g is orthogonal to all $f \in \mathcal{F}_0$. A subspace $\mathcal{G}_0 \subseteq \mathcal{F}$ is the *orthogonal complement* to a subspace $\mathcal{F}_0 \subseteq \mathcal{F}$ iff for all $g \in \mathcal{G}_0$, g is orthogonal to $\mathcal{F}_0$, and $\mathcal{F}$ is spanned by elements from $\mathcal{F}_0$ and $\mathcal{G}_0$. A vector $f_0^* \in \mathcal{F}_0$ is the *projection* of a vector $f \in \mathcal{F}$ onto the subspace $\mathcal{F}_0 \in \mathcal{F}$ iff it minimizes $\|f - f_0\|$ over $f_0 \in \mathcal{F}_0$. The projection f_0^* of f onto $\mathcal{F}_0$, denoted $f \circ \mathcal{F}_0$, exists and is unique almost everywhere with respect to the measure μ. A necessary and sufficient condition for f_0^* to be the projection of f onto $\mathcal{F}_0$ is that the *approximation error* $(f - f_0^*)$ is orthogonal to $\mathcal{F}_0$ [34, Thm. 2, p. 51]. Intuitively, this is the Pythagorean theorem.

C. Bézier Volumes

A *Bézier curve* of degree m is a function on the unit interval $b : [0, 1] \to \mathbb{R}$ specified as a linear combination of Bernstein polynomials, namely,

$$b(x) = \sum_{i=0}^{m} B_i b_{m,i}(x), \tag{3}$$

where $B_0, \ldots, B_m$ are the coefficients of the linear combination, and

$$b_{m,i}(x) = \binom{m}{i} x^i (1 - x)^{m-i}, \tag{4}$$

$i = 0, \ldots, m$, are the mth order Bernstein polynomials, which are polynomials of degree m defined on the unit interval $[0, 1]$.

Analogously, a *Bézier volume* (BV) of degree m is a function on the unit cube $b : [0, 1]^3 \to \mathbb{R}$ specified as a linear combination of products of Bernstein polynomials, namely,

$$b(x, y, z) = \sum_{i=0}^{m} \sum_{j=0}^{m} \sum_{k=0}^{m} B_{ijk} b_{m,i}(x) b_{m,j}(y) b_{m,k}(z), \tag{5}$$

where B_{ijk}, $i, j, k \in \{0, \ldots, m\}$, are the coefficients of the linear combination.

A function $b(x, y, z)$ is *tri-polynomial* of degree m if it is a polynomial of degree m in each of its coordinates when its other coordinates have any fixed value. Thus a BV is tri-polynomial of degree m over the unit cube.

D. Cardinal B-Splines

A *cardinal B-spline function* of order p is a function on the real line $f : \mathbb{R} \to \mathbb{R}$ specified as a linear combination of B-spline basis functions of order p, namely

$$f(x) = \sum_{n \in \mathbb{Z}} F_n \phi^{(p)}(x - n), \tag{6}$$

where F_n, $n \in \mathbb{Z}$, are the coefficients of the linear combination, and $\phi^{(p)}(x - n)$ is the B-spline basis function of order p at integer shift n. The B-spline basis function $\phi^{(p)}(x)$ can be defined for $p = 1$ as

$$\phi^{(1)}(x) = \begin{cases} 1 & x \in [0, 1) \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

and recursively for $p > 1$ as

$$\phi^{(p)}(x) = \int \phi^{(1)}(t) \phi^{(p-1)}(x - t) dt \tag{8}$$

for all x. From this definition, it can be seen that $\phi^{(p)}(x)$ is the p-fold convolution of $\phi^{(1)}(x)$ with itself, and that the support of $\phi^{(p)}(x)$ is an interval of length p, as shown in Fig. 2.

[1] Note that $\|f\|$ depends on the value of f only at the points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$. Hence strictly speaking, $\|f\|$ is only a pseudo-norm, as there are non-zero functions $f : \mathbb{R}^3 \to \mathbb{R}$ such that $\|f\| = 0$. However, we call $\|f\|$ a norm with the usual understanding that $\|f\| = 0$ implies $f = 0$ a.e. (almost everywhere) with respect to measure μ. Alternatively, we consider the space of equivalence classes of functions, where two functions f and g are deemed equivalent if $f = g$ a.e. Denoting by $\tilde{f}$ the equivalence class of functions equivalent to f, one can show that the set of equivalence classes $\tilde{\mathcal{F}} = \{\tilde{f} | f : \mathbb{R}^3 \to \mathbb{R}\}$ is a vector space with a proper norm $\|\tilde{f}\|$ induced by the inner product $\left\langle \tilde{f}, \tilde{g} \right\rangle$, which in turn is induced by the inner product $\langle f, g \rangle$ between representatives. The vector space $\tilde{\mathcal{F}}$ is clearly isomorphic to $\mathbb{R}^N$, and is hence a Hilbert space. Thus we take $\mathcal{F}$ to be a Hilbert space with the usual "almost everywhere" understanding or with the equivalence class understanding [33, p. 165].

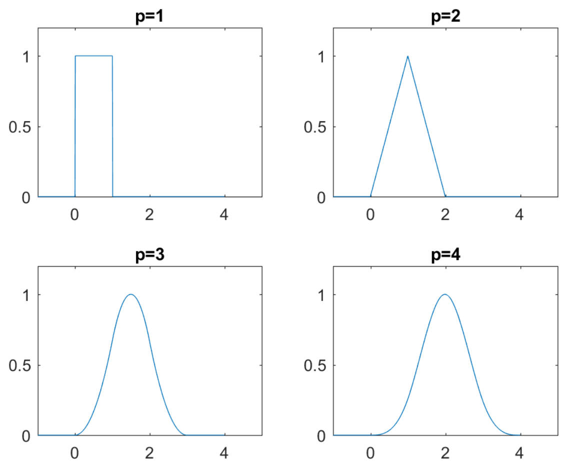


Fig. 2. Central B-spline basis functions of order p.

An alternative recursive definition for $p > 1$ is

$$\phi^{(p)}(x) = \frac{1}{p-1}\left[x\phi^{(p-1)}(x) + (p-x)\phi^{(p-1)}(x-1)\right] \quad (9)$$

for $x \in [0, p]$ and $\phi^{(p)}(x) = 0$ otherwise. From this second definition it follows that $\phi^{(p)}(x)$ is a polynomial of degree $p-1$ on all integer shifts of the unit interval, $[n, n+1)$, $n \in \mathbb{Z}$, which we call *blocks*, and that $f(x)$ is piecewise polynomial of degree $p-1$ over each block. It also follows that $f(x)$ is C^{p-2} continuous, meaning that $f(x)$ and all of its derivatives up to its $(p-2)$th derivative are continuous, even at the breakpoints between blocks, which are called *knots*.

Analogously, a *cardinal B-spline volume* of order p is a volumetric function $f : \mathbb{R}^3 \to \mathbb{R}$ specified as a linear combination of vector integer shifts of a product of B-spline basis functions of order p, namely

$$f(\boldsymbol{x}) = \sum_{\boldsymbol{n}\in\mathbb{Z}^3} F_{\boldsymbol{n}}\phi^{(p)}(\boldsymbol{x}-\boldsymbol{n}), \quad (10)$$

where $F_{\boldsymbol{n}}$ is the coefficient of the linear combination at vector integer shift $\boldsymbol{n} \in \mathbb{Z}^3$, and $\phi^{(p)}(\boldsymbol{x}) = \phi^{(p)}(x, y, z) = \phi^{(p)}(x)\phi^{(p)}(y)\phi^{(p)}(z)$ is the product of B-spline basis functions $\phi^{(p)}(x)$, $\phi^{(p)}(y)$, and $\phi^{(p)}(z)$.

It can be seen that $f(\boldsymbol{x})$ is tri-polynomial of degree $p-1$ over each shifted unit cube $[0, 1]^3 + \boldsymbol{n}$, or *block*. Further, it can be seen that $f(\boldsymbol{x})$ is C^{p-2} continuous between blocks. Thus $f(\boldsymbol{x})$ can be considered to be a collection of Bézier volumes of order $p-1$ linked together such that the overall function is C^{p-2} continuous.

E. Approximation

Let $\phi_{0,\boldsymbol{0}}(\boldsymbol{x}) = \phi^{(p)}(\boldsymbol{x} - \boldsymbol{n}_0)$ be the *central* cardinal B-spline basis function, that is, the cardinal B-spline basis function centered on the origin $\boldsymbol{0}$ (if p is even) or on the unit cube $[0, 1]^3$ (if p is odd). Let $\phi_{0,\boldsymbol{n}}(\boldsymbol{x}) = \phi_{0,\boldsymbol{0}}(\boldsymbol{x} - \boldsymbol{n})$ be this central cardinal B-spline basis function shifted by integer vector $\boldsymbol{n}$.

For notational simplicity, we here suppress the dependence of $\phi_{0,\boldsymbol{n}}$ on p. We discuss specific values of p in Section IV.

Let $\mathcal{F}$ be the Hilbert space of all functions $f : \mathbb{R}^3 \to \mathbb{R}$ under the inner product (2), as defined in Subsection III-B. Define the subspace $\mathcal{F}_0 \subseteq \mathcal{F}$ as

$$\mathcal{F}_0 = \left\{ f_0 \in \mathcal{F} \,\middle|\, \exists\{F_{0,\boldsymbol{n}}\} \text{ s.t. } f_0(\boldsymbol{x}) = \sum_{\boldsymbol{n}\in\mathbb{Z}^3} F_{0,\boldsymbol{n}}\phi_{0,\boldsymbol{n}}(\boldsymbol{x}) \right\}. \quad (11)$$

This is the subspace of all functions that are tri-polynomial of degree $p-1$ over the blocks $\{[0, 1)^3 + \boldsymbol{n} | \boldsymbol{n} \in \mathbb{Z}^3\}$.

Any function $f \in \mathcal{F}$ can be approximated by a function $f_0^* \in \mathcal{F}_0$, where f_0^* is the projection of f onto $\mathcal{F}_0$, denoted $f_0^* = f \circ \mathcal{F}_0$. Let $\{F_{0,\boldsymbol{n}}^*\}$ be coefficients such that $f_0^*(\boldsymbol{x}) = \sum_{\boldsymbol{n}\in\mathcal{Z}^3} F_{0,\boldsymbol{n}}^*\phi_{0,\boldsymbol{n}}(\boldsymbol{x})$. Under the inner product (2), the squared error between f and f_0^*,

$$\|f - f_0^*\|^2 = \sum_{i=1}^{N}(f(\boldsymbol{x}_i) - f_0^*(\boldsymbol{x}_i))^2, \quad (12)$$

depends on the values of f_0^* only at the points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$, which in turn depend on any particular coefficient $F_{0,\boldsymbol{n}}^*$ only if $\phi_{0,\boldsymbol{n}}(\boldsymbol{x}_i) \neq 0$ for some $\boldsymbol{x}_i$, $i = 1\ldots, N$. Let

$$\mathcal{N}_0 = \left\{\boldsymbol{n} \,\middle|\, \exists i \in \{1, \ldots, N\} \text{ s.t. } \phi_{0,\boldsymbol{n}}(\boldsymbol{x}_i) \neq 0\right\} \quad (13)$$

be the set of vector integer shifts $\boldsymbol{n}$ such that $\phi_{0,\boldsymbol{n}}(\boldsymbol{x}_i) \neq 0$ for some $\boldsymbol{x}_i$. This set is *finite* because $\phi_{0,\boldsymbol{0}}$ has bounded support, and any of its shifts far away from the points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ will not include any such point in its support. Let $\{\boldsymbol{n}_i\}$ denote the finite set of shifts in $\mathcal{N}_0$.

For any $\boldsymbol{n} \notin \mathcal{N}_0$, assign $F_{0,\boldsymbol{n}}^* = 0$, and for any $\boldsymbol{n} \in \mathcal{N}_0$, solve for $F_{0,\boldsymbol{n}}^*$ by noting that the approximation error $(f - f_0^*)$ must be orthogonal to $\phi_{0,\boldsymbol{n}}$ for all $\boldsymbol{n} \in \mathbb{Z}^3$. In particular, for all $\boldsymbol{n}_i \in \mathcal{N}_0$,

$$0 = \langle\phi_{0,\boldsymbol{n}_i}, f - f_0^*\rangle = \langle\phi_{0,\boldsymbol{n}_i}, f\rangle - \langle\phi_{0,\boldsymbol{n}_i}, f_0^*\rangle, \quad (14)$$

or

$$\langle\phi_{0,\boldsymbol{n}_i}, f\rangle = \langle\phi_{0,\boldsymbol{n}_i}, f_0^*\rangle = \sum_{\boldsymbol{n}_j\in\mathcal{N}_0} \langle\phi_{0,\boldsymbol{n}_i}, \phi_{0,\boldsymbol{n}_j}\rangle F_{0,\boldsymbol{n}_j}^*. \quad (15)$$

These are the *normal equations*. In vector form,

$$\Phi_0^T f = \Phi_0^T\Phi_0 F_0^* \quad (16)$$

where $\Phi_0^T f$ is shorthand for the $|\mathcal{N}_0| \times 1$ vector $\left[\langle\phi_{0,\boldsymbol{n}_i}, f\rangle\right]$, $\Phi_0^T\Phi_0$ is shorthand for the $|\mathcal{N}_0| \times |\mathcal{N}_0|$ matrix $\left[\langle\phi_{0,\boldsymbol{n}_i}, \phi_{0,\boldsymbol{n}_j}\rangle\right]$, and F_0^* is the $|\mathcal{N}_0| \times 1$ vector $[F_{0,\boldsymbol{n}_j}^*]$. If $\Phi_0^T\Phi_0$ is invertible, then one may solve for F_0^* explicitly as

$$F_0^* = (\Phi_0^T\Phi_0)^{-1}\Phi_0^T f. \quad (17)$$

F. Multiresolution Approximation

To obtain approximations at different resolutions, the cardinal B-spline basis functions can be scaled by a factor of $2^{-\ell}$, where ℓ is the *scale* or *level of detail* or simply *level*. To be specific, define

$$\phi_{\ell,\boldsymbol{n}}(\boldsymbol{x}) = \phi_{0,\boldsymbol{n}}(2^\ell\boldsymbol{x}) \quad (18)$$

as the cardinal B-spline basis function at level ℓ and shift $\boldsymbol{n}$. Define the subspace $\mathcal{F}_\ell \subseteq \mathcal{F}$ as

$$\mathcal{F}_\ell = \left\{ f_\ell \in \mathcal{F} \,\middle|\, \exists\{F_{\ell,\boldsymbol{n}}\} \text{ s.t. } f_\ell(\boldsymbol{x}) = \sum_{\boldsymbol{n}\in\mathbb{Z}^3} F_{\ell,\boldsymbol{n}}\phi_{\ell,\boldsymbol{n}}(\boldsymbol{x}) \right\}. \tag{19}$$

This is the subspace of all functions that are tri-polynomial of degree $p-1$ over the blocks at level ℓ, $\{2^{-\ell}([0,1)^3+\boldsymbol{n})| \boldsymbol{n}\in\mathbb{Z}^3\}$. Since the blocks at level ℓ are refined by the blocks at level $\ell+1$, it is clear that if a function f_ℓ is tri-polynomial over the blocks at level ℓ, i.e., $f_\ell \in \mathcal{F}_\ell$, then it is also tri-polynomial over the blocks at level $\ell+1$, i.e., $f_\ell \in \mathcal{F}_{\ell+1}$. Hence $\mathcal{F}_\ell \subseteq \mathcal{F}_{\ell+1}$ and

$$\mathcal{F}_0 \subseteq \mathcal{F}_1 \subseteq \ldots \subseteq \mathcal{F}_\ell \subseteq \mathcal{F}_{\ell+1} \subseteq \ldots \subseteq \mathcal{F} \tag{20}$$

is a nested sequence of subspaces whose resolution increases with ℓ.

Let $f_\ell^* = f \circ \mathcal{F}_\ell$ and $f_{\ell+1}^* = f \circ \mathcal{F}_{\ell+1}$ be the projections of f onto $\mathcal{F}_\ell$ and $\mathcal{F}_{\ell+1}$, respectively. Then by the Pythagorean theorem [34, Sec. 3.3], for all $f_\ell \in \mathcal{F}_\ell \subseteq \mathcal{F}_{\ell+1}$,

$$\|f - f_\ell\|^2 = \|f - f_{\ell+1}^*\|^2 + \|f_{\ell+1}^* - f_\ell\|^2. \tag{21}$$

Then since $f_\ell^* = f \circ \mathcal{F}_\ell$ minimizes $\|f - f_\ell\|^2$ over all $f_\ell \in \mathcal{F}_\ell$, by (21) f_ℓ^* must also minimize $\|f_{\ell+1}^* - f_\ell\|^2$ over all $f_\ell \in \mathcal{F}_\ell$, and hence $f_\ell^* = f_{\ell+1}^* \circ \mathcal{F}_\ell$. That is, projecting f onto $\mathcal{F}_\ell$ can be done in two steps, by first projecting onto $\mathcal{F}_{\ell+1}$ (i.e., $f_{\ell+1}^* = f \circ \mathcal{F}_{\ell+1}$) and then onto $\mathcal{F}_\ell$ (i.e., $f_\ell^* = f_{\ell+1}^* \circ \mathcal{F}_\ell$). Alternatively, $f \circ \mathcal{F}_\ell = f \circ \mathcal{F}_{\ell+1} \circ \mathcal{F}_\ell$.

Paralleling the development in the previous subsection, let

$$\mathcal{N}_\ell = \left\{\boldsymbol{n} \,\middle|\, \exists i \in \{1,\ldots,N\} \text{ s.t. } \phi_{\ell,\boldsymbol{n}}(\boldsymbol{x}_i) \neq 0\right\} \tag{22}$$

be the finite set of vector integer shifts $\boldsymbol{n}$ such that $\phi_{\ell,\boldsymbol{n}}(\boldsymbol{x}_i) \neq 0$ for some $\boldsymbol{x}_i$. Then for all $\boldsymbol{n}_i \in \mathcal{N}_\ell$,

$$0 = \langle \phi_{\ell,\boldsymbol{n}_i}, f - f_\ell^* \rangle = \langle \phi_{\ell,\boldsymbol{n}_i}, f \rangle - \langle \phi_{\ell,\boldsymbol{n}_i}, f_\ell^* \rangle, \tag{23}$$

or

$$\langle \phi_{\ell,\boldsymbol{n}_i}, f \rangle = \langle \phi_{\ell,\boldsymbol{n}_i}, f_\ell^* \rangle = \sum_{\boldsymbol{n}_j \in \mathcal{N}_\ell} \langle \phi_{\ell,\boldsymbol{n}_i}, \phi_{\ell,\boldsymbol{n}_j} \rangle F_{\ell,\boldsymbol{n}_j}^*, \tag{24}$$

where $f_\ell^* = \sum_{\boldsymbol{n}_j \in \mathcal{N}_\ell} F_{\ell,\boldsymbol{n}_j}^* \phi_{\ell,\boldsymbol{n}_j}$. In vector form, (24) can be expressed

$$\Phi_\ell^T f = \Phi_\ell^T \Phi_\ell F_\ell^* \tag{25}$$

where $\Phi_\ell^T f$ is shorthand for the $|\mathcal{N}_\ell| \times 1$ vector $[\langle \phi_{\ell,\boldsymbol{n}_i}, f \rangle]$, $\Phi_\ell^T \Phi_\ell$ is shorthand for the $|\mathcal{N}_\ell| \times |\mathcal{N}_\ell|$ matrix $[\langle \phi_{\ell,\boldsymbol{n}_i}, \phi_{\ell,\boldsymbol{n}_j} \rangle]$, and F_ℓ^* is the $|\mathcal{N}_\ell| \times 1$ vector $[F_{\ell,\boldsymbol{n}_j}^*]$. If $\Phi_\ell^T \Phi_\ell$ is invertible, then one may solve for F_ℓ^* explicitly as

$$F_\ell^* = (\Phi_\ell^T \Phi_\ell)^{-1} \Phi_\ell^T f. \tag{26}$$

In turn, one may compute $\Phi_\ell^T f$ and $\Phi_\ell^T \Phi_\ell$ recursively from $\Phi_{\ell+1}^T f$ and $\Phi_{\ell+1}^T \Phi_{\ell+1}$, respectively, as follows.

Since $\phi_{\ell,\boldsymbol{n}} \in \mathcal{F}_\ell \subseteq \mathcal{F}_{\ell+1}$, there exist coefficients $\{a_{\boldsymbol{k}}\}$ not depending on ℓ or $\boldsymbol{n}$ such that

$$\phi_{\ell,\boldsymbol{n}} = \sum_{\boldsymbol{k}\in\mathbb{Z}^3} a_{\boldsymbol{k}} \phi_{\ell+1,2\boldsymbol{n}+\boldsymbol{k}} = \sum_{\boldsymbol{k}'\in\mathbb{Z}^3} a_{\boldsymbol{k}'-2\boldsymbol{n}} \phi_{\ell+1,\boldsymbol{k}'}. \tag{27}$$

Simply put, the coefficients $\{a_{\boldsymbol{k}}\}$ are the coefficients that express $\phi_{\ell,\boldsymbol{n}}$ in the basis $\{\phi_{\ell+1,\boldsymbol{n}}\}$ of $\mathcal{F}_{\ell+1}$. The number of non-zero coefficients in $\{a_{\boldsymbol{k}}\}$ is finite because both $\phi_{\ell,\boldsymbol{n}}$ and the functions in the basis $\{\phi_{\ell_1,\boldsymbol{n}}\}$ have bounded support. The coefficients $\{a_{\boldsymbol{k}}\}$ do not depend on ℓ or $\boldsymbol{n}$ because of the scale and shift invariance of the basis functions at each level. Moreover, they do not depend on the measure because (27) is a purely functional relationship.

Equation (27) is known as the *two-scale equation*. From this equation, it follows that

$$\langle \phi_{\ell,\boldsymbol{n}_i}, f \rangle = \sum_{\boldsymbol{n}_j \in \mathcal{N}_{\ell+1}} a_{\boldsymbol{n}_j - 2\boldsymbol{n}_i} \langle \phi_{\ell+1,\boldsymbol{n}_j}, f \rangle \tag{28}$$

and

$$\langle \phi_{\ell,\boldsymbol{n}_i}, \phi_{\ell,\boldsymbol{n}_j} \rangle = \sum_{\boldsymbol{n}_k \in \mathcal{N}_{\ell+1}} \sum_{\boldsymbol{n}_l \in \mathcal{N}_{\ell+1}} a_{\boldsymbol{n}_k - 2\boldsymbol{n}_i} \langle \phi_{\ell+1,\boldsymbol{n}_k}, \phi_{\ell+1,\boldsymbol{n}_l} \rangle a_{\boldsymbol{n}_l - 2\boldsymbol{n}_j}. \tag{29}$$

In vector form,

$$\Phi_\ell^T f = A_\ell \Phi_{\ell+1}^T f \tag{30}$$

and

$$\Phi_\ell^T \Phi_\ell = A_\ell \Phi_{\ell+1}^T \Phi_{\ell+1} A_\ell^T, \tag{31}$$

where $A_\ell = [a_{\boldsymbol{n}_j - 2\boldsymbol{n}_i}]$. Another useful recursion is

$$\langle \phi_{\ell,\boldsymbol{n}_i}, \phi_{\ell+1,\boldsymbol{n}_j} \rangle = \sum_{\boldsymbol{n}_k \in \mathcal{N}_{\ell+1}} a_{\boldsymbol{n}_k - 2\boldsymbol{n}_i} \langle \phi_{\ell+1,\boldsymbol{n}_k}, \phi_{\ell+1,\boldsymbol{n}_j} \rangle, \tag{32}$$

or

$$\Phi_\ell^T \Phi_{\ell+1} = A_\ell \Phi_{\ell+1}^T \Phi_{\ell+1} \tag{33}$$

in vector form. The matrix A_ℓ, which has size $|\mathcal{N}_l| \times |\mathcal{N}_{\ell+1}|$, is sparse with its non-zero entries at locations (i, j) determined by the relative positions of $\boldsymbol{n}_i \in \mathcal{N}_\ell$ and $\boldsymbol{n}_j \in \mathcal{N}_{\ell+1}$. Thus, A_ℓ depends on both ℓ and the measure (through the set of points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$), even though its coefficients are drawn from the set $\{a_{\boldsymbol{k}}\}$, which is independent of ℓ and the measure.

G. Wavelets

Let $\mathcal{G}_\ell$ be the orthogonal complement of $\mathcal{F}_\ell$ in $\mathcal{F}_{\ell+1}$, i.e.,

$$\mathcal{F}_{\ell+1} = \mathcal{F}_\ell \oplus \mathcal{G}_\ell. \tag{34}$$

Applying this recursively, we have

$$\mathcal{F}_{\ell+1} = \mathcal{F}_0 \oplus \mathcal{G}_0 \oplus \mathcal{G}_1 \oplus \cdots \oplus \mathcal{G}_\ell, \tag{35}$$

so that any function $f_{\ell+1} \in \mathcal{F}_{\ell+1}$ can be written as the sum of orthogonal functions,

$$f_{\ell+1} = f_0 + g_0 + g_1 + \cdots + g_\ell. \tag{36}$$

The coefficients of f_0 in the basis for $\mathcal{F}_0$ are *low pass* coefficients, while the coefficients of g_ℓ in the basis for $\mathcal{G}_\ell$ are *high pass* or *wavelet* coefficients. The function f can be communicated by quantizing and entropy coding its low pass and wavelet coefficients. This is efficient because most of the energy in f is in its low pass coefficients and its low-level wavelet coefficients.

To compute the low pass coefficients, (17) can be used, while to compute the wavelet coefficients, we first need to establish a basis for each $\mathcal{G}_\ell$.

First, some definitions: For each ℓ, let $\Phi_\ell = [\phi_{\ell,\boldsymbol{n}_j}]$ be a row vector whose $|\mathcal{N}_\ell|$ elements are the *functions* $\phi_{\ell,\boldsymbol{n}_j}$, $\boldsymbol{n}_j \in \mathcal{N}_\ell$, and let F_ℓ be a column vector of $|\mathcal{N}_\ell|$ coefficients. Let $\Phi_\ell F_\ell$ denote the function $f_\ell = \sum_{\boldsymbol{n}_j \in \mathcal{N}_\ell} F_{\ell,\boldsymbol{n}_j} \phi_{\ell,\boldsymbol{n}_j}$ in $\mathcal{F}_{\ell+1}$. Let $\Phi_\ell^T f = [\langle \phi_{\ell,\boldsymbol{n}_i}, f \rangle]$ denote the column vector of $|\mathcal{N}_\ell|$ inner products of the functions $\phi_{\ell,\boldsymbol{n}_i}$, $\boldsymbol{n}_i \in \mathcal{N}_\ell$, with the function f. Similarly, let $\Phi_\ell^T[f_1, \ldots, f_n] = [\langle \phi_{\ell,\boldsymbol{n}_i}, f_j \rangle]$ denote the matrix of inner products of the functions $\phi_{\ell,\boldsymbol{n}_i}$, $\boldsymbol{n}_i \in \mathcal{N}_\ell$, with the functions f_j $j = 1, \ldots, n$.

Consider now the subspace $\mathcal{G}_\ell \subseteq \mathcal{F}_{\ell+1}$ defined by

$$\mathcal{G}_\ell = \left\{ \Phi_{\ell+1} F_{\ell+1} \middle| \Phi_\ell^T \Phi_{\ell+1} F_{\ell+1} = 0 \right\}. \tag{37}$$

This is the set of all linear combinations $f_{\ell+1} = \Phi_{\ell+1} F_{\ell+1}$ of the basis functions of $\mathcal{F}_{\ell+1}$ for which the linear combinations are orthogonal to all basis functions of $\mathcal{F}_\ell$ (i.e., $\Phi_\ell^T f_{\ell+1} = 0$). Clearly, $\mathcal{G}_\ell$ is a subspace of $\mathcal{F}_{\ell+1}$ and is orthogonal to $\mathcal{F}_\ell = \{\Phi_\ell F_\ell\}$, and hence $\mathcal{G}_\ell$ is the orthogonal complement of $\mathcal{F}_\ell$ in $\mathcal{F}_{\ell+1}$. If $N_{\ell+1}$ is the dimension of $\mathcal{F}_{\ell+1}$ and N_ℓ is the dimension of $\mathcal{F}_\ell$, then $N_{\ell+1} - N_\ell$ is the dimension of $\mathcal{G}_\ell$. Typically, $N_\ell = |\mathcal{N}_\ell|$. (When ℓ is large, the dimension N_ℓ of $\mathcal{F}_\ell$ may be lower than $|\mathcal{N}_\ell|$, due to the finite number of points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$. We will point out what to do about this case later in this subsection.)

One way to construct an explicit basis for $\mathcal{G}_\ell$ is as follows. Here we assume that $N_\ell = |\mathcal{N}_\ell|$, that is, $\Phi_\ell^T \Phi_{\ell+1}$ has full rank. Partition the $N_\ell \times N_{\ell+1}$ matrix $\Phi_\ell^T \Phi_{\ell+1}$ into an $N_\ell \times N_\ell$ matrix A and an $N_\ell \times (N_{\ell+1} - N_\ell)$ matrix B, as $\Phi_\ell^T \Phi_{\ell+1} = [A\ B]$. Similarly partition the $N_{\ell+1}$ dimensional vector $F_{\ell+1}$ into an N_ℓ dimensional vector F^a and an $N_{\ell+1} - N_\ell$ dimensional vector F^b, as

$$F_{\ell+1} = \begin{bmatrix} F^a \\ F^b \end{bmatrix}. \tag{38}$$

Then for all $F_{\ell+1}$ satisfying $\Phi_\ell^T \Phi_{\ell+1} F_{\ell+1} = 0$ in (37),

$$[A\ B] \begin{bmatrix} F^a \\ F^b \end{bmatrix} = 0 \tag{39}$$

$$A^{-1}[A\ B] \begin{bmatrix} F^a \\ F^b \end{bmatrix} = [I^a\ A^{-1}B] \begin{bmatrix} F^a \\ F^b \end{bmatrix} = 0 \tag{40}$$

where I^a is the $N_\ell \times N_\ell$ identity matrix. (Note that A^{-1} exists because we have assumed that $[A\ B]$ has full rank.) Hence $F^a = -A^{-1}BF^b$, and therefore $F_{\ell+1} = ZF^b$, where

$$Z = \begin{bmatrix} -A^{-1}B \\ I^b \end{bmatrix} \tag{41}$$

is an $N_{\ell+1} \times (N_{\ell+1} - N_\ell)$ matrix containing the $(N_{\ell+1} - N_\ell) \times (N_{\ell+1} - N_\ell)$ identity matrix I^b. Thus

$$\mathcal{G}_\ell = \left\{ \Phi_{\ell+1} Z F^b \middle| F^b \in \mathbb{R}^{N_{\ell+1} - N_\ell} \right\}, \tag{42}$$

and the $(N_{\ell+1} - N_\ell)$ *functions* in the row vector

$$\Psi_\ell = \Phi_{\ell+1} Z \tag{43}$$

form an explicit basis for $\mathcal{G}_\ell$.

More generally, instead of (41), any matrix Z whose columns span the null space of $\Phi_\ell^T \Phi_{\ell+1}$ may be used to form a basis for $\mathcal{G}_\ell$.

If the rank N_ℓ of $\Phi_\ell^T \Phi_{\ell+1}$ is less than $|\mathcal{N}_\ell|$, then indices $\boldsymbol{n}_i \in \mathcal{N}_\ell$ (corresponding to rows of $\Phi_\ell^T \Phi_{\ell+1}$) may be removed until $|\mathcal{N}_\ell|$ is equal to the rank N_ℓ of $\Phi_\ell^T \Phi_{\ell+1}$, i.e., until $\Phi_\ell^T \Phi_{\ell+1}$ has full rank. Then, the above full-rank procedure may be followed. As will be seen later, this will ensure a critically sampled transform.

Now that we have established a basis for $\mathcal{G}_\ell$, any function $f_{\ell+1} = \Phi_{\ell+1} F_{\ell+1} \in \mathcal{F}_{\ell+1}$ can be decomposed as the sum of functions $f_\ell = \Phi_\ell F_\ell \in \mathcal{F}_\ell$ and $g_\ell = \Psi_\ell G_\ell \in \mathcal{G}_\ell$, specifically,

$$\Phi_{\ell+1} F_{\ell+1} = \begin{bmatrix} \Phi_\ell & \Psi_\ell \end{bmatrix} \begin{bmatrix} F_\ell \\ G_\ell \end{bmatrix}. \tag{44}$$

How to obtain F_ℓ and G_ℓ from $F_{\ell+1}$, and vice-versa, is the subject of the next subsection.

H. Two-Channel Filter Banks

Formulas for an analysis filter bank (which produce coefficients F_ℓ and G_ℓ from coefficients $F_{\ell+1}$) and a synthesis filter bank (which produce coefficients $F_{\ell+1}$ from coefficients F_ℓ and G_ℓ) may be obtained by taking the inner product of (44) with the N_ℓ functions in Φ_ℓ and the $N_{\ell+1} - N_\ell$ functions in Ψ_ℓ,

$$\begin{bmatrix} \Phi_\ell & \Psi_\ell \end{bmatrix}^T \Phi_{\ell+1} F_{\ell+1} = \begin{bmatrix} \Phi_\ell^T \Phi_\ell & 0 \\ 0 & \Psi_\ell^T \Psi_\ell \end{bmatrix} \begin{bmatrix} F_\ell \\ G_\ell \end{bmatrix}, \tag{45}$$

yielding

$$\begin{bmatrix} F_\ell \\ G_\ell \end{bmatrix} = \begin{bmatrix} \Phi_\ell^T \Phi_\ell & 0 \\ 0 & \Psi_\ell^T \Psi_\ell \end{bmatrix}^{-1} \begin{bmatrix} \Phi_\ell^T \Phi_{\ell+1} \\ \Psi_\ell^T \Phi_{\ell+1} \end{bmatrix} F_{\ell+1} \tag{46}$$

for the analysis and

$$F_{\ell+1} = \begin{bmatrix} \Phi_\ell^T \Phi_{\ell+1} \\ \Psi_\ell^T \Phi_{\ell+1} \end{bmatrix}^{-1} \begin{bmatrix} \Phi_\ell^T \Phi_\ell & 0 \\ 0 & \Psi_\ell^T \Psi_\ell \end{bmatrix} \begin{bmatrix} F_\ell \\ G_\ell \end{bmatrix} \tag{47}$$

for the synthesis.

The elements of $F_{\ell+1}$ may be re-ordered such that the first N_ℓ elements F^a are designated *even* and the last $N_{\ell+1} - N_\ell$ elements F^b are designated *odd*, as in (38), analogous to a polyphase decomposition. Correspondingly, the elements of $\Phi_{\ell+1}$ may be re-ordered as $\Phi_{\ell+1} = \begin{bmatrix} \Phi_{\ell+1}^a & \Phi_{\ell+1}^b \end{bmatrix}$. Then the analysis and synthesis (46) and (47) may be re-written

$$\begin{bmatrix} F_\ell \\ G_\ell \end{bmatrix} = T_\ell \begin{bmatrix} F_{\ell+1}^a \\ F_{\ell+1}^b \end{bmatrix}, \quad \begin{bmatrix} F_{\ell+1}^a \\ F_{\ell+1}^b \end{bmatrix} = T_\ell^{-1} \begin{bmatrix} F_\ell \\ G_\ell \end{bmatrix}, \tag{48}$$

where

$$T_\ell = \begin{bmatrix} (\Phi_\ell^T \Phi_\ell)^{-1} \Phi_\ell^T \Phi_{\ell+1}^a & (\Phi_\ell^T \Phi_\ell)^{-1} \Phi_\ell^T \Phi_{\ell+1}^b \\ (\Psi_\ell^T \Psi_\ell)^{-1} \Psi_\ell^T \Phi_{\ell+1}^a & (\Psi_\ell^T \Psi_\ell)^{-1} \Psi_\ell^T \Phi_{\ell+1}^b \end{bmatrix} \tag{49}$$

$$= \begin{bmatrix} A & B \\ C & D \end{bmatrix}, \tag{50}$$

where A, B, C, and D are the blocks of the matrix, and T_ℓ^{-1} may be expressed using either of the block matrix inverse formulas (51) and (52), shown at the top of the next page.

The block matrix expressions for T_ℓ and T_ℓ^{-1} lead to three equivalent analysis and synthesis filter banks, shown respectively in Fig. 3 and Fig. 4. The filter banks may be nested, recursively decomposing the low pass coefficients.

$$T_\ell^{-1} = \begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} A^{-1}+A^{-1}B(D-CA^{-1}B)^{-1}CA^{-1} & -A^{-1}B(D-CA^{-1}B)^{-1} \\ -(D-CA^{-1}B)^{-1}CA^{-1} & (D-CA^{-1}B)^{-1}CA^{-1} \end{bmatrix} \tag{51}$$

$$= \begin{bmatrix} (A-BD^{-1}C)^{-1} & -(A-BD^{-1}C)^{-1}BD^{-1} \\ -D^{-1}C(A-BD^{-1}C)^{-1} & D^{-1}+D^{-1}C(A-BD^{-1}C)^{-1}BD^{-1} \end{bmatrix} \tag{52}$$

I. Orthogonality

From the above construction, the elements of Φ_ℓ are orthogonal to the elements of Ψ_ℓ, but the elements of Φ_ℓ are not necessarily orthogonal to each other, and the elements of Ψ_ℓ are not necessarily orthogonal to each other. However, if desired, they may be orthogonalized (and normalized) by finding an $N_\ell \times N_\ell$ matrix R_ℓ and an $(N_{\ell+1}-N_\ell)\times(N_{\ell+1}-N_\ell)$ matrix S_ℓ such that the elements of the row vector $\Phi_\ell R_\ell$ are orthonormal (i.e., $R_\ell^T\Phi_\ell^T\Phi_\ell R_\ell = I$) and the elements of the row vector $\Psi_\ell S_\ell$ are orthonormal (i.e., $S_\ell^T\Psi_\ell^T\Psi_\ell S_\ell = I$), respectively. To find R_ℓ, perform the eigen-decomposition of the positive definite matrix $\Phi_\ell^T\Phi_\ell = U\Lambda U^T$, where U is the matrix whose columns are orthonormal eigenvectors and Λ is the diagonal matrix of positive eigenvalues. Then $R_\ell = U\Lambda^{-1/2}$ since in that case $R_\ell^T\Phi_\ell^T\Phi_\ell R_\ell = R_\ell^T U\Lambda U^T R_\ell = I$. Similarly, $S_\ell = V\Delta^{-1/2}$, where V is the matrix of eigenvectors of $\Psi_\ell^T\Psi_\ell$ and Δ is the diagonal matrix of positive eigenvalues.

Denoting the row vectors of orthonormal basis functions for $\mathcal{F}_\ell$ and $\mathcal{G}_\ell$ by

$$\bar\Phi_\ell = \Phi_\ell R_\ell, \quad \bar\Psi_\ell = \Psi_\ell S_\ell, \tag{53}$$

and denoting their coefficients by

$$\bar F_\ell = R_\ell^{-1} F_\ell, \quad \bar G_\ell = S_\ell^{-1} G_\ell, \tag{54}$$

orthonormal versions of the analysis and synthesis may be written

$$\begin{bmatrix} \bar F_\ell \\ \bar G_\ell \end{bmatrix} = \begin{bmatrix} R_\ell^{-1}(\Phi_\ell^T\Phi_\ell)^{-1}\Phi_\ell^T\Phi_{\ell+1} \\ S_\ell^{-1}(\Psi_\ell^T\Psi_\ell)^{-1}\Psi_\ell^T\Phi_{\ell+1} \end{bmatrix} F_{\ell+1} \tag{55}$$

$$= \begin{bmatrix} (R_\ell^T\Phi_\ell^T\Phi_\ell R_\ell)^{-1}R_\ell^T\Phi_\ell^T\Phi_{\ell+1} \\ (S_\ell^T\Psi_\ell^T\Psi_\ell S_\ell)^{-1}S_\ell^T\Psi_\ell^T\Phi_{\ell+1} \end{bmatrix} F_{\ell+1} \tag{56}$$

$$= \begin{bmatrix} R_\ell^T\Phi_\ell^T\Phi_{\ell+1}R_{\ell+1} \\ S_\ell^T\Psi_\ell^T\Phi_{\ell+1}R_{\ell+1} \end{bmatrix} \bar F_{\ell+1} \tag{57}$$

$$= \begin{bmatrix} \bar\Phi_\ell^T\bar\Phi_{\ell+1} \\ \bar\Psi_\ell^T\bar\Phi_{\ell+1} \end{bmatrix} \bar F_{\ell+1} = \bar T_\ell \bar F_{\ell+1}, \tag{58}$$

where the forward transform $\bar T_\ell$ is orthonormal, and hence its inverse is simply its transpose, $\bar T_\ell^{-1} = \bar T_\ell^T$. Thus

$$\bar F_{\ell+1} = \begin{bmatrix} \bar\Phi_\ell^T\bar\Phi_{\ell+1} \\ \bar\Psi_\ell^T\bar\Phi_{\ell+1} \end{bmatrix}^T \begin{bmatrix} \bar F_\ell \\ \bar G_\ell \end{bmatrix} = \bar T_\ell^T \begin{bmatrix} \bar F_\ell \\ \bar G_\ell \end{bmatrix}. \tag{59}$$

Like T_ℓ, $\bar T_\ell$ can also be written as a block matrix,

$$\bar T_\ell = \begin{bmatrix} \bar\Phi_\ell^T\bar\Phi_{\ell+1}^a & \bar\Phi_\ell^T\bar\Phi_{\ell+1}^b \\ \bar\Psi_\ell^T\bar\Phi_{\ell+1}^a & \bar\Psi_\ell^T\bar\Phi_{\ell+1}^b \end{bmatrix} = \begin{bmatrix} \bar A & \bar B \\ \bar C & \bar D \end{bmatrix}, \tag{60}$$

and hence $\bar T_\ell$ and $\bar T_\ell^{-1}$ have the same structures shown in Figs. 3 and 4, except that the expressions for the synthesis butterfly are much simpler: $\bar A^T$, $\bar C^T$, $\bar B^T$, and $\bar D^T$.

Note that because the transforms are orthonormal at each stage of the wavelet transform, the overall transform is orthonormal as well as critically sampled.

IV. Attribute Compression Using Volumetric Functions

In this section, we show how to represent and compress the real-valued attributes of a point cloud using volumetric functions. We assume that we are given, at the encoder, a set of point locations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ and a set of corresponding attributes $f_1, \ldots, f_N$. We also assume that the point locations can be communicated to the decoder without loss.[2] The problem of attribute compression is to reproduce approximate attributes, $\hat f_1, \ldots, \hat f_N$, at the decoder, subject to a constraint on the number of bits communicated, given the point locations as side information. We assume for simplicity that the attributes are scalar. Vector attributes can be treated component-wise.

Our approach is the following. At the encoder, a volumetric B-spline of order p is fit to the values $f_1, \ldots, f_N$ at locations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$, and its wavelet coefficients are quantized and entropy coded. At the decoder, the wavelet coefficients are entropy decoded and dequantized, and the volumetric B-spline is reconstructed. Finally the reconstructed volumetric B-spline is sampled at the locations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$, and the corresponding values $\hat f_1, \ldots, \hat f_N$ are used as reproductions of the attributes. If desired, the reconstructed volumetric B-spline may also be sampled at other locations to continuously interpolate the attributes, a unique feature of our approach.

The next three sections deal with volumetric B-splines of orders 1, 2, and higher orders.

A. Constant B-Splines

In this subsection, we treat the case of volumetric B-splines of order $p = 1$, or *constant* B-splines. We show that constant B-splines are equivalent to the Region-Adaptive Haar Transform (RAHT) introduced in [29], [30]. To show their equivalence, we first review RAHT.

The Region-Adaptive Haar Transform was introduced as a generalization of the Haar Transform. The Haar Transform of a sequence of 2^d coefficients $f_0, \ldots, f_{2^d-1}$ is frequently described as a series of orthonormal butterfly transforms,

$$\begin{bmatrix} \bar F_{\ell,n} \\ \bar G_{\ell,n} \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} \bar F_{\ell+1,2n} \\ \bar F_{\ell+1,2n+1} \end{bmatrix}, \tag{61}$$

for $\ell = d-1, \ldots, 0$ and $n = 0, \ldots, 2^\ell - 1$, beginning with $\bar F_{d,n} = f_n$, $n = 0, \ldots, 2^d - 1$, The butterfly structure for $d = 3$ is shown in Fig. 5. This can equally well be regarded as a full

[2] These assumptions do not preclude lossy geometry coding, in which case we take $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ to be the result of compressing and decompressing original point locations $\tilde{\boldsymbol{x}}_1, \ldots, \tilde{\boldsymbol{x}}_{\tilde N}$, and we take $f_1, \ldots, f_N$ to be the result of transferring original attributes $\tilde f_1, \ldots, \tilde f_{\tilde N}$ to the decompressed point locations. The process of *attribute transfer* is unique to point cloud compression [2], and it affects the interplay of geometry and attribute compression, but it is outside the scope of this paper.

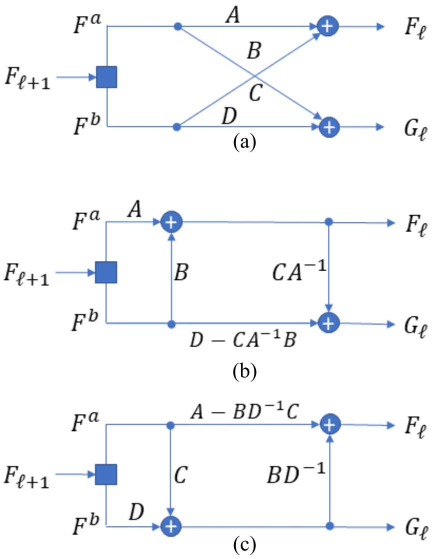


Fig. 3. Three equivalent analysis filter banks, from top: (a) butterfly, (b) lifting with prediction followed by update, and (c) lifting with update followed by prediction.

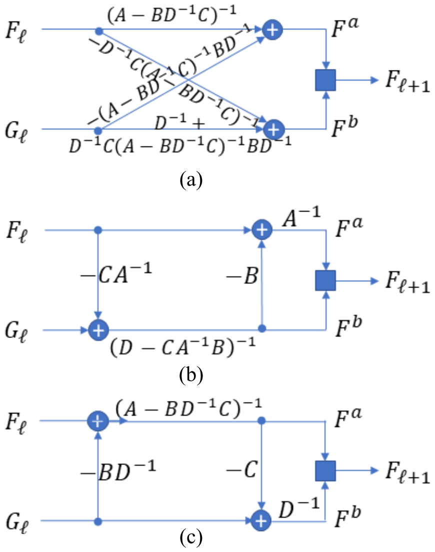


Fig. 4. Three equivalent synthesis filter banks, from top: (a) butterfly using (51) or (52), (b) lifting with prediction followed by update, and (c) lifting with update followed by prediction.

binary tree, as shown in Fig. 6, in which the signal samples $f_0, \ldots, f_{2^d-1}$ are located at the leaves of the tree, the high pass coefficients $\bar{G}_{\ell,n}$ are located at the intermediate nodes of the tree, and the DC coefficient is located at the root of the tree.

The Region-Adaptive Haar Transform is a generalization of the Haar Transform in that the tree is not necessarily full, in that the internal nodes of the tree may be either binary or unary, and the butterfly transform at each internal binary node of the tree is a Givens rotation,

$$\begin{bmatrix} \bar{F}_{\ell,n} \\ \bar{G}_{\ell,n} \end{bmatrix} = \begin{bmatrix} a & b \\ -b & a \end{bmatrix} \begin{bmatrix} \bar{F}_{\ell+1,2n} \\ \bar{F}_{\ell+1,2n+1} \end{bmatrix}, \tag{62}$$

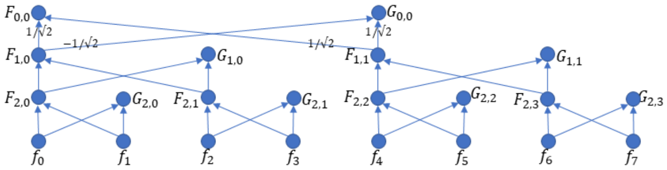


Fig. 5. Haar Transform butterfly structure for depth $d = 3$.

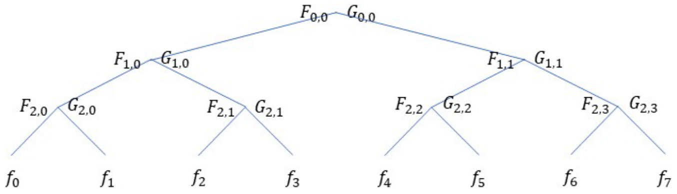


Fig. 6. Haar Transform tree structure for depth $d = 3$.

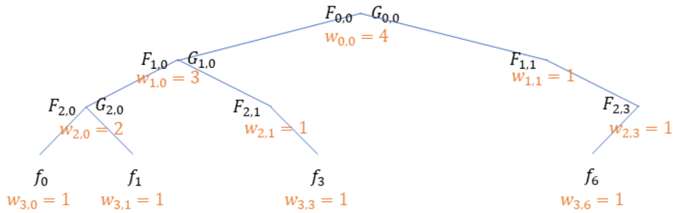


Fig. 7. RAHT tree structure for depth $d = 3$.

where

$$a = \frac{\sqrt{w_{\ell+1,2n}}}{\sqrt{w_{\ell+1,2n} + w_{\ell+1,2n+1}}}, \tag{63}$$

$$b = \frac{\sqrt{w_{\ell+1,2n+1}}}{\sqrt{w_{\ell+1,2n} + w_{\ell+1,2n+1}}}, \tag{64}$$

$$w_{\ell,n} = w_{\ell+1,2n} + w_{\ell+1,2n+1} \tag{65}$$

for $\ell = 0, \ldots, d-1$, and $w_{d,n}$ equals 1 for all n in the signal and equals 0 otherwise. The quantity $w_{\ell,n}$ is called the *weight* of node n at level ℓ and is equal to the total of the weights of all the leaves descended from node n in level ℓ. The tree for RAHT is shown in Fig. 7.

To apply RAHT to a signal $f_1, \ldots, f_N \in \mathbb{R}$ defined on point locations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \in \mathbb{R}^3$, first scale the point locations so that the set $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ fits within the unit cube, $\mathcal{X} \subset [0,1)^3$, and choose d sufficiently large so that each point location $\boldsymbol{x} = (x, y, z) \in \mathcal{X}$ can be represented uniquely with d bits of precision, as $x = \sum_{b=1}^{d} x_b 2^{-b}$, $y = \sum_{b=1}^{d} y_b 2^{-b}$, and $z = \sum_{b=1}^{d} z_b 2^{-b}$, or in more conventional notation,

$$x = .x_1 \cdots x_d\ , \tag{66}$$

$$y = .y_1 \cdots y_d\ , \tag{67}$$

$$z = .z_1 \cdots z_d\ . \tag{68}$$

The *Morton code* of point location $\boldsymbol{x} = (x, y, z)$ is defined as the interleaving of its coefficients' bits,

$$\text{Morton}(\boldsymbol{x}) = .x_1 y_1 z_1 \cdots x_d y_d z_d. \tag{69}$$

Now the RAHT tree of depth $3d$ can be constructed with N leaves, with f_i at leaf i, such that the path from the root of the tree to leaf i is given by the Morton code of $\boldsymbol{x}_i$.

Thus each node in the tree, if it is at level ℓ, is associated with the common length-ℓ prefix shared by the Morton codes of the node's descendants. Two nodes at level $\ell+1$ are siblings if their length-ℓ Morton prefixes are identical. The Givens rotation is applied to such siblings.

As described above, RAHT is a discrete transform of a signal defined on a discrete set of points. But it also has an interpretation as a continuous transform of a signal defined on a volume. To see that, note that the nodes at level ℓ of the RAHT tree partition not only the points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$, but also the volume $[0,1)^3$. For example, consider a node at a level $\ell = 3\ell'$ whose Morton prefix is $.x_1y_1z_1\cdots x_{\ell'}y_{\ell'}z_{\ell'}$. This node corresponds to the set of all points $\boldsymbol{x} \in [0,1)^3$ that have this same Morton prefix, namely the cube

$$\begin{aligned}\mathcal{B}_{\ell,(n_x,n_y,n_z)} = {} & [2^{-\ell'}n_x, 2^{-\ell'}(n_x+1)) \\ & \times [2^{-\ell'}n_y, 2^{-\ell'}(n_y+1)) \\ & \times [2^{-\ell'}n_z, 2^{-\ell'}(n_z+1)),\end{aligned} \tag{70}$$

where $n_x = \sum_{b=1}^{\ell'} x_b 2^{\ell'-b}$, $n_y = \sum_{b=1}^{\ell'} y_b 2^{\ell'-b}$, and $n_z = \sum_{b=1}^{\ell'} z_b 2^{\ell'-b}$. This is a $2^{-\ell'}\times 2^{-\ell'}\times 2^{-\ell'}$ cube at vector integer shift $\boldsymbol{n} = (n_x, x_y, n_z)$, where $n_x, n_y, n_z \in \{0, \ldots, 2^{\ell'}-1\}$. For nodes at levels ℓ that are not a multiple of three, the notation is a little clumsy, but similar and straightforward. Specifically, a node at level $\ell = 3\ell'+1$ corresponds to a $2^{-(\ell'+1)} \times 2^{-\ell'} \times 2^{-\ell'}$ cuboid $\mathcal{B}_{\ell,\boldsymbol{n}}$ at vector integer shift $\boldsymbol{n} = (n_x, x_y, n_z)$, where $n_x \in \{0, \ldots, 2^{\ell'+1}-1\}$ and $n_y, n_z \in \{0, \ldots, 2^{\ell'}-1\}$, and a node at level $\ell = 3\ell'+2$ corresponds to a $2^{-(\ell'+1)}\times 2^{-(\ell'+1)} \times 2^{-\ell'}$ cuboid $\mathcal{B}_{\ell,\boldsymbol{n}}$ at vector integer shift $\boldsymbol{n} = (n_x, x_y, n_z)$, where $n_x, n_y \in \{0, \ldots, 2^{\ell'+1}-1\}$ and $n_z \in \{0, \ldots, 2^{\ell'}-1\}$. At any level, we call these cuboids *blocks*. A block $\mathcal{B}_{\ell,\boldsymbol{n}}$ is said to be *occupied* if it contains a point, that is, if $\mathcal{B}_{\ell,\boldsymbol{n}} \cap \mathcal{X} \neq \emptyset$. Each node in the RAHT tree at level ℓ correspond to an occupied block at level ℓ.

Let $\phi_{\ell,\boldsymbol{n}}(\boldsymbol{x})$ the indicator function for block $\mathcal{B}_{\ell,\boldsymbol{n}}$, namely

$$\phi_{\ell,\boldsymbol{n}}(\boldsymbol{x}) = \begin{cases} 1 & \boldsymbol{x} \in \mathcal{B}_{\ell,\boldsymbol{n}} \\ 0 & \text{otherwise,} \end{cases} \tag{71}$$

and, as in (22), let $\mathcal{N}_\ell$ be the set of vector integer shifts $\boldsymbol{n}$ such that $\phi_{\ell,\boldsymbol{n}}(\boldsymbol{x}_i) \neq 0$ for some $i = 1, \ldots, N$, that is, let $\mathcal{N}_\ell$ be the set of vector integer shifts $\boldsymbol{n}$ such that $\mathcal{B}_{\ell,\boldsymbol{n}}$ is occupied. Then, as in (19), let $\mathcal{F}_\ell$ be the subspace of functions that are linear combinations of $\phi_{\ell,\boldsymbol{n}}$ for $\boldsymbol{n} \in \mathcal{N}_\ell$,

$$\mathcal{F}_\ell = \left\{ f_\ell \in \mathcal{F} \;\middle|\; \exists\{F_{\ell,\boldsymbol{n}}\} \text{ s.t. } f_\ell = \sum_{\boldsymbol{n}\in\mathcal{N}_\ell} F_{\ell,\boldsymbol{n}}\phi_{\ell,\boldsymbol{n}} \right\}. \tag{72}$$

Let $f : [0,1)^3 \to \mathbb{R}$ be any function that agrees with f_i on $\boldsymbol{x}_i$, i.e., $f(\boldsymbol{x}_i) = f_i$, $i = 1, \ldots, N$, and let $f_\ell^* = \sum_{\boldsymbol{n}\in\mathcal{N}_\ell} F_{\ell,\boldsymbol{n}}^*\phi_{\ell,\boldsymbol{n}}$ be the projection of f onto $\mathcal{F}_\ell$. As in Subsection III-F, let $\Phi_\ell^T f$ denote the $|\mathcal{N}_\ell| \times 1$ vector $[\langle\phi_{\ell,\boldsymbol{n}_i}, f\rangle]$, let $\Phi_\ell^T\Phi_\ell$ denote the $|\mathcal{N}_\ell| \times |\mathcal{N}_\ell|$ matrix $[\langle\phi_{\ell,\boldsymbol{n}_i}, \phi_{\ell,\boldsymbol{n}_j}\rangle]$, and let F_ℓ^* denote the $|\mathcal{N}_\ell| \times 1$ vector $[F_{\ell,\boldsymbol{n}_j}^*]$. If $\Phi_\ell^T\Phi_\ell$ is invertible, then

$$F_\ell^* = (\Phi_\ell^T\Phi_\ell)^{-1}\Phi_\ell^T f. \tag{73}$$

From the definition of $\phi_{\ell,\boldsymbol{n}}$ (71) and the inner product (2), it can be seen that $\langle\phi_{\ell,\boldsymbol{n}}, f\rangle = \sum_{\boldsymbol{x}_i\in\mathcal{B}_{\ell,\boldsymbol{n}}} f(\boldsymbol{x}_i)$ and $\langle\phi_{\ell,\boldsymbol{n}}, \phi_{\ell,\boldsymbol{n}'}\rangle$ equals $w_{\ell,\boldsymbol{n}}$ if $\boldsymbol{n} = \boldsymbol{n}'$ and equals 0 otherwise, where $w_{\ell,\boldsymbol{n}} = \mu(\mathcal{B}_{\ell,\boldsymbol{n}})$ is the number of points in the set $\mathcal{B}_{\ell,\boldsymbol{n}}$. Hence $F_{\ell,\boldsymbol{n}}^*$ is the average value of the attributes of the points in $\mathcal{B}_{\ell,\boldsymbol{n}}$, namely

$$F_{\ell,\boldsymbol{n}}^* = \frac{\langle\phi_{\ell,\boldsymbol{n}}, f\rangle}{w_{\ell,\boldsymbol{n}}} = \frac{1}{w_{\ell,\boldsymbol{n}}}\sum_{\boldsymbol{x}_i\in\mathcal{B}_{\ell,\boldsymbol{n}}} f(\boldsymbol{x}_i), \tag{74}$$

for $\boldsymbol{n} \in \mathcal{N}_\ell$.

To express the two-scale equation for $\phi_{\ell,\boldsymbol{n}}$ succinctly regardless of whether ℓ is a multiple of three or not, we use the following notation. If $\ell, \boldsymbol{n}$ are the level and shift of a block, then "$\ell+1, 2\boldsymbol{n}$" and "$\ell+1, 2\boldsymbol{n}+1$" mean the level and shifts of its two sub-blocks. To be pedantic, "$\ell+1, 2\boldsymbol{n}$" and "$\ell+1, 2\boldsymbol{n}+1$" mean $\ell+1, (2n_x, n_y, n_z)$ and $\ell+1, (2n_x+1, n_y, n_z)$ if $\ell = 3\ell'$ (i.e., ℓ is a multiple of 3); they mean $\ell+1, (n_x, 2n_y, n_z)$ and $\ell+1, (n_x, 2n_y+1, n_z)$ if $\ell = 3\ell'+1$ (i.e., $\ell \equiv 1 \bmod 3$); and they mean $\ell+1, (n_x, n_y, 2n_z)$ and $\ell+1, (n_x, n_y, 2n_z+1)$ if $\ell = 3\ell'+2$ (i.e., $\ell \equiv 2 \bmod 3$).

Now the two-scale equation for $\phi_{\ell,\boldsymbol{n}}$ can be easily expressed

$$\phi_{\ell,\boldsymbol{n}} = \phi_{\ell+1,2\boldsymbol{n}} + \phi_{\ell+1,2\boldsymbol{n}+1}, \tag{75}$$

so that combining (74) and (75),

$$\begin{aligned} F_{\ell,\boldsymbol{n}}^* &= \frac{\langle\phi_{\ell+1,2\boldsymbol{n}}, f\rangle}{w_{\ell,\boldsymbol{n}}} + \frac{\langle\phi_{\ell+1,2\boldsymbol{n}+1}, f\rangle}{w_{\ell,\boldsymbol{n}}} && (76)\\ &= \frac{w_{\ell+1,2\boldsymbol{n}}}{w_{\ell,\boldsymbol{n}}}F_{\ell+1,2\boldsymbol{n}}^* + \frac{w_{\ell+1,2\boldsymbol{n}+1}}{w_{\ell,\boldsymbol{n}}}F_{\ell+1,2\boldsymbol{n}+1}^* && (77)\\ &= \frac{w_0}{w_0+w_1}F_{\ell+1,2\boldsymbol{n}}^* + \frac{w_1}{w_0+w_1}F_{\ell+1,2\boldsymbol{n}+1}^*, && (78)\end{aligned}$$

where to be more concise we have abbreviated $w_0 = w_{\ell+1,2\boldsymbol{n}}$ and $w_1 = w_{\ell+1,2\boldsymbol{n}+1}$. Both w_0 and w_1 will be non-zero when $\ell, \boldsymbol{n}$ correspond to a node in the tree with two children. For such $\ell, \boldsymbol{n}$, define the function

$$\psi_{\ell,\boldsymbol{n}} = -\frac{\phi_{\ell+1,2\boldsymbol{n}}}{w_0} + \frac{\phi_{\ell+1,2\boldsymbol{n}+1}}{w_1}, \tag{79}$$

and define

$$G_{\ell,\boldsymbol{n}}^* = \frac{w_0w_1}{w_0+w_1}\langle\psi_{\ell,\boldsymbol{n}}, f\rangle. \tag{80}$$

Then combining (80), (79), and (74),

$$\begin{aligned} G_{\ell,\boldsymbol{n}}^* &= \frac{w_0w_1}{w_0+w_1}\left(-\frac{\langle\phi_{\ell+1,2\boldsymbol{n}}, f\rangle}{w_0} + \frac{\langle\phi_{\ell+1,2\boldsymbol{n}+1}, f\rangle}{w_1}\right) && (81)\\ &= \frac{w_0w_1}{w_0+w_1}\left(-F_{\ell+1,2\boldsymbol{n}}^* + F_{\ell+1,2\boldsymbol{n}+1}^*\right), && (82)\end{aligned}$$

which is the scaled difference between the average values of the attributes of the points in the two sub-blocks $\mathcal{B}_{\ell+1,2\boldsymbol{n}}$ and $\mathcal{B}_{\ell+1,2\boldsymbol{n}+1}$ of $\mathcal{B}_{\ell,\boldsymbol{n}}$. When f is smooth, $G_{\ell,\boldsymbol{n}}^*$ will be close to zero. Putting (78) and (82) in matrix form,

$$\begin{bmatrix} F_{\ell,\boldsymbol{n}}^* \\ G_{\ell,\boldsymbol{n}}^* \end{bmatrix} = \begin{bmatrix} \dfrac{w_0}{w_0+w_1} & \dfrac{w_1}{w_0+w_1} \\ -\dfrac{w_0w_1}{w_0+w_1} & \dfrac{w_0w_1}{w_0+w_1} \end{bmatrix} \begin{bmatrix} F_{\ell+1,2\boldsymbol{n}}^* \\ F_{\ell+1,2\boldsymbol{n}+1}^* \end{bmatrix}. \tag{83}$$

Clearly, both $\phi_{\ell,\boldsymbol{n}}$ and $\psi_{\ell,\boldsymbol{n}}$ have support only on $\mathcal{B}_{\ell,\boldsymbol{n}}$, and hence they are both orthogonal to both $\phi_{\ell,\boldsymbol{n}'}$ and $\psi_{\ell,\boldsymbol{n}'}$ for vector integer shifts $\boldsymbol{n}' \neq \boldsymbol{n}$. But it can also be seen that $\phi_{\ell,\boldsymbol{n}}$

and $\psi_{\ell,\boldsymbol{n}}$ are orthogonal to each other, since $\langle\phi_{\ell,\boldsymbol{n}}, \psi_{\ell,\boldsymbol{n}}\rangle = -1+1 = 0$. Thus the orthogonal complement of $\mathcal{F}_\ell$ in $\mathcal{F}_{\ell+1}$ is

$$\mathcal{G}_\ell = \left\{ g_\ell \,\middle|\, g_\ell = \sum_{\boldsymbol{n}\in\mathcal{N}_\ell^b} G_{\ell,\boldsymbol{n}}\psi_{\ell,\boldsymbol{n}} \right\}, \tag{84}$$

where the sum is over only those vector integer shifts $\mathcal{N}_\ell^b \subseteq \mathcal{N}_\ell$ for which $\ell, \boldsymbol{n}$ correspond to a node in the tree with two children. However, as defined, the orthogonal basis functions $\phi_{\ell,\boldsymbol{n}}$, and $\psi_{\ell,\boldsymbol{n}}$ are not normalized. Define their normalized versions

$$\bar{\phi}_{\ell,\boldsymbol{n}} = \frac{\phi_{\ell,\boldsymbol{n}}}{\|\phi_{\ell,\boldsymbol{n}}\|} = \frac{\phi_{\ell,\boldsymbol{n}}}{\sqrt{w_0+w_1}}, \tag{85}$$

$$\bar{\psi}_{\ell,\boldsymbol{n}} = \frac{\psi_{\ell,\boldsymbol{n}}}{\|\psi_{\ell,\boldsymbol{n}}\|} = \frac{\psi_{\ell,\boldsymbol{n}}}{\sqrt{\frac{1}{w_0}+\frac{1}{w_1}}} = \frac{\sqrt{w_0w_1}}{\sqrt{w_0+w_1}}\psi_{\ell,\boldsymbol{n}}. \tag{86}$$

Then $f_\ell = \sum_{\boldsymbol{n}\in\mathcal{N}_\ell} F_{\ell,\boldsymbol{n}}\phi_{\ell,\boldsymbol{n}} = \sum_{\boldsymbol{n}\in\mathcal{N}_\ell} \bar{F}_{\ell,\boldsymbol{n}}\bar{\phi}_{\ell,\boldsymbol{n}}$ and $g_\ell = \sum_{\boldsymbol{n}\in\mathcal{N}_\ell^b} G_{\ell,\boldsymbol{n}}\psi_{\ell,\boldsymbol{n}} = \sum_{\boldsymbol{n}\in\mathcal{N}_\ell^b} \bar{G}_{\ell,\boldsymbol{n}}\bar{\psi}_{\ell,\boldsymbol{n}}$, where

$$\bar{F}_{\ell,\boldsymbol{n}} = \sqrt{w_0+w_1}F_{\ell,\boldsymbol{n}}, \tag{87}$$

$$\bar{G}_{\ell,\boldsymbol{n}} = \frac{\sqrt{w_0+w_1}}{\sqrt{w_0w_1}}G_{\ell,\boldsymbol{n}}. \tag{88}$$

Rewriting (83),

$$\begin{bmatrix} \frac{1}{\sqrt{w_0+w_1}}\bar{F}^*_{\ell,\boldsymbol{n}} \\ \frac{\sqrt{w_0w_1}}{\sqrt{w_0+w_1}}\bar{G}^*_{\ell,\boldsymbol{n}} \end{bmatrix} = \begin{bmatrix} \frac{w_0}{w_0+w_1} & \frac{w_1}{w_0+w_1} \\ -\frac{w_0w_1}{w_0+w_1} & \frac{w_0w_1}{w_0+w_1} \end{bmatrix} \times \begin{bmatrix} \frac{\bar{F}^*_{\ell+1,2\boldsymbol{n}}}{\sqrt{w_0}} \\ \frac{\bar{F}^*_{\ell+1,2\boldsymbol{n}+1}}{\sqrt{w_1}} \end{bmatrix}, \tag{89}$$

or

$$\begin{bmatrix} \bar{F}^*_{\ell,\boldsymbol{n}} \\ \bar{G}^*_{\ell,\boldsymbol{n}} \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{w_0}}{\sqrt{w_0+w_1}} & \frac{\sqrt{w_1}}{\sqrt{w_0+w_1}} \\ -\frac{\sqrt{w_1}}{\sqrt{w_0+w_1}} & \frac{\sqrt{w_0}}{\sqrt{w_0+w_1}} \end{bmatrix} \begin{bmatrix} \bar{F}^*_{\ell+1,2\boldsymbol{n}} \\ \bar{F}^*_{\ell+1,2\boldsymbol{n}+1} \end{bmatrix}. \tag{90}$$

This is identical to (62). Thus RAHT has a volumetric interpretation.

B. Tri-linear B-Splines

In this subsection, we treat the case of volumetric B-splines of order $p = 2$, or *tri-linear* B-splines. These splines are continuous, unlike constant B-splines. Tri-linear B-splines reduce blocking artifacts, and perhaps more importantly, do not develop rips, tears, or holes when the surface or motion representation is quantized, due to their guaranteed continuity.

We begin with the central cardinal B-spline of order $p = 2$,

$$\phi^{(2)}(x-1) = \begin{cases} 1+x & x\in[-1,0] \\ 1-x & x\in[0,1] \\ 0 & \text{otherwise}, \end{cases} \tag{91}$$

and define the volumetric version

$$\phi_{0,\boldsymbol{0}}(\boldsymbol{x}) = \phi^{(2)}(x-1)\phi^{(2)}(y-1)\phi^{(2)}(z-1). \tag{92}$$

Then

$$\phi_{\ell,\boldsymbol{n}}(\boldsymbol{x}) = \phi_{0,\boldsymbol{0}}(2^\ell\boldsymbol{x}-\boldsymbol{n}) \tag{93}$$

is the tri-linear B-spline basis function at level ℓ with vector integer shift $\boldsymbol{n}$. As in (19), let

$$\mathcal{F}_\ell = \left\{ f_\ell \in \mathcal{F} \,\middle|\, \exists\{F_{\ell,\boldsymbol{n}}\} \text{ s.t. } f_\ell(\boldsymbol{x}) = \sum_{\boldsymbol{n}\in\mathcal{N}_\ell} F_{\ell,\boldsymbol{n}}\phi_{\ell,\boldsymbol{n}}(\boldsymbol{x}) \right\}. \tag{94}$$

be the subspace of all functions in the Hilbert space $\mathcal{F}$ that are tri-linear over all blocks $\mathcal{B}_{\ell,\boldsymbol{n}}$, $\boldsymbol{n}\in\mathbb{Z}^3$, at level ℓ. It suffices to take the sum over vector integer shifts $\boldsymbol{n}\in\mathcal{N}_\ell$, where $2^{-\ell}\mathcal{N}_\ell$ is a subset of the *corners* of the occupied blocks $\mathcal{B}_{\ell,\boldsymbol{n}}$. This is because, in the case of tri-linear splines, whenever $2^{-\ell}\boldsymbol{n}$ is *not* adjacent to any occupied block, $\phi_{\ell,\boldsymbol{n}}(\boldsymbol{x}_i) = 0$ for all $\boldsymbol{x}_i$.

As in (26), the projection of $f\in\mathcal{F}$ onto $\mathcal{F}_\ell$ is given by $F_\ell^* = (\Phi_\ell^T\Phi_\ell)^{-1}\Phi_\ell^T f$, where $\Phi_\ell^T f$ is the $|\mathcal{N}_\ell|\times 1$ vector $[\langle\phi_{\ell,\boldsymbol{n}_i}, f\rangle]$, $\Phi_\ell^T\Phi_\ell$ is the $|\mathcal{N}_\ell|\times|\mathcal{N}_\ell|$ matrix $[\langle\phi_{\ell,\boldsymbol{n}_i}, \phi_{\ell,\boldsymbol{n}_j}\rangle]$, and F_ℓ^* is the $|\mathcal{N}_\ell|\times 1$ vector $[F^*_{\ell,\boldsymbol{n}_j}]$. However, the matrix $\Phi_\ell^T\Phi_\ell$ in the case of tri-linear splines, unlike the case of constant splines, is block tri-diagonal rather than diagonal, and hence is not so trivial to invert. Nevertheless, $\Phi_\ell^T\Phi_\ell$ is sparse, and thus inversion by iterative methods is quite feasible.

The elements of $\Phi_\ell^T\Phi_\ell$ and $\Phi_\ell^T f$ do not have to be computed directly from their definitions. Rather, they can be computed directly from their definitions in the simple case of $\ell = d$, and then they can be computed recursively for $\ell < d$, using the two-scale equation (27). For the tri-linear B-spline, the coefficients in the two-scale equation are

$$a_{\boldsymbol{k}} = \begin{cases} 2^{-\|\boldsymbol{k}\|_1} & \boldsymbol{k}\in\{-1,0,1\}^3 \\ 0 & \text{otherwise}, \end{cases} \tag{95}$$

where $\|\boldsymbol{k}\|_1 = |k_x|+|k_y|+|k_z|$ is the 1-norm of $\boldsymbol{k} = (k_x,k_y,k_z)$. Thus, as in (30) and (31), for $\ell < d$,

$$\Phi_\ell^T f = A_\ell\Phi_{\ell+1}^T f \tag{96}$$

and

$$\Phi_\ell^T\Phi_\ell = A_\ell\Phi_{\ell+1}^T\Phi_{\ell+1}A_\ell^T, \tag{97}$$

where $A_\ell = [a_{\boldsymbol{n}_j-2\boldsymbol{n}_i}]$ for $\boldsymbol{n}_i\in\mathcal{N}_\ell$, $\boldsymbol{n}_j\in\mathcal{N}_{\ell+1}$. For $\ell = d$, we may use simply

$$\Phi_d^T f = [f_i] \tag{98}$$

and

$$\Phi_d^T\Phi_d = I_N, \tag{99}$$

where I_N is the $N\times N$ identity matrix. This is possible because the point locations $\boldsymbol{x}_i$ can be taken to be the origins of the occupied *voxels*, or blocks $\mathcal{B}_{d,\boldsymbol{n}}$ at level d.

Consider now the orthogonal complement $\mathcal{G}_\ell$ of $\mathcal{F}_\ell$ in $\mathcal{F}_{\ell+1}$. As in the case of constant B-splines, in the case of tri-linear B-splines, the basis functions $\psi_{\ell,\boldsymbol{n}}$ of $\mathcal{G}_\ell$ depend locally on the point locations $\{\boldsymbol{x}_i\}$, hence are not shifts of each other as in the

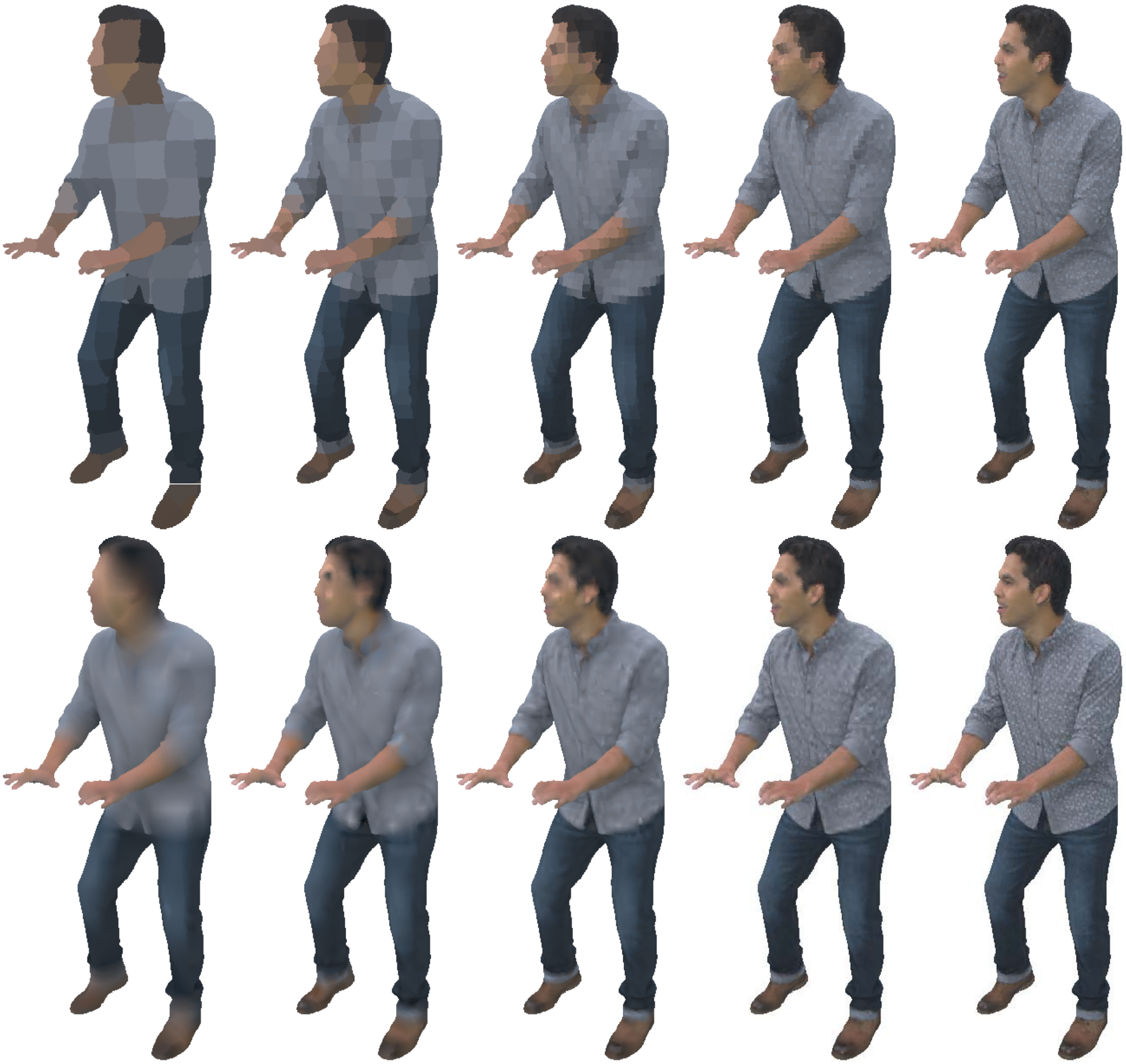


Fig. 8. Smoothing the *loot* dataset for $p = 1$ (RAHT, top) and $p = 2$ (BV, bottom), at levels 4 (left), 5, 6, 7, and 8 (right).

case of Lebesgue measure. In the case of constant B-splines, we were able to define explicitly basis functions $\psi_{\ell,\boldsymbol{n}}$ orthogonal to each other and to $\mathcal{F}_\ell$. Unfortunately, in the case of tri-linear B-splines, this is less easy to do. Nevertheless, for tri-linear B-splines, the procedure in Section III-G may be followed for constructing a basis for $\mathcal{G}_\ell$. For smoothing applications in which high pass coefficients are simply set to zero, this procedure is sufficient. However, for coding applications, it is additionally important to orthonormalize the basis functions, so that scalar quantization of their coefficients does not introduce more error in the signal domain than in the transform domain. To orthonormalize the basis functions, the procedure in Section III-I may be followed. Unfortunately, more explicit constructions of the basis functions are not known at this time.

C. Higher-Order B-Splines

Volumetric B-splines of order $p \geq 3$ are possible, and offer higher order continuity properties. However, they are more complex to compute, and at each level ℓ, they require significantly more coefficients per occupied block. Specifically, a function in $\mathcal{F}_\ell$ requires $O(p^3)$ coefficients per occupied block. Although coefficients beyond the first N need not be transmitted as they are linear combinations of the first N, extra smoothness gained from the higher order may not be worth the added complexity.

V. Experimental Results

The framework presented in this paper allows us to extend RAHT from order $p = 1$ to more general Bézier Volumes $p \geq 1$. In this section, we show the improvement of $p = 2$

TABLE I

DATASETS USED FOR ATTRIBUTE REPRESENTATION AND COMPRESSION

Dataset name	Original resolution	Original point count	10-bit point count	7-bit point count
loot_vox10_1200	10 bit	805285	805285	14711
queen_0200	10 bit	1000993	1000993	14683
soldier_vox10_0690	10 bit	1089091	1089091	19993
shiva_00035_vox12	12 bit	1009132	900662	30045
longdress_vox10_1300	10 bit	857966	857966	15688

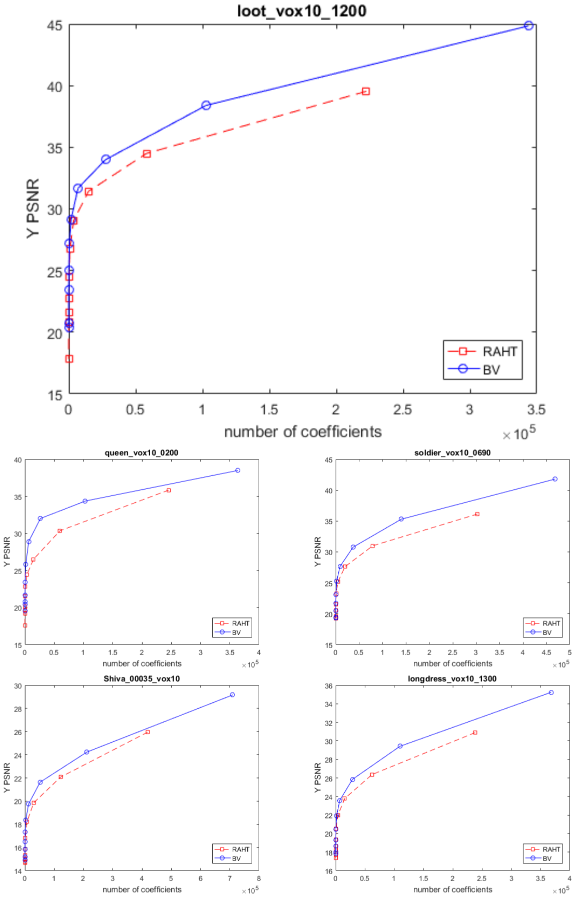


Fig. 9. Energy compaction for $p = 2$ (BV) vs $p = 1$ (RAHT).

over $p = 1$ for attribute representation and compression. We assume the geometry is known, and examine only color attributes. Evaluations are performed on the datasets listed in Table I. These datasets are the ones used in the MPEG Point Cloud Coding standardization activity to compare attribute coding methods, assuming lossless geometry [35]. The original datasets all have 10-bit resolution, with the exception of *shiva*, which has 12-bit resolution. (d-bit resolution means that the point locations are represented as d-bit integers.) For our experiments, all original datasets are first voxelized to either 10-bit or 7-bit resolution.

Fig. 8 shows the qualitative results of smoothing the 10-bit *loot* dataset by setting the wavelet coefficients G_ℓ to zero for all levels $\ell \geq L$, where $L = 4$ (left) through $L = 8$ (right), for $p = 1$ (RAHT, top) and $p = 2$ (BV, bottom). The corresponding blocks have blockwidths 2^{10-L}, i.e., the blocks are $64 \times 64 \times 64$ voxels (left) through $4 \times 4 \times 4$ voxels (right). For RAHT, blocking artifacts are clearly visible, while for BV, blocking artifacts are much less visible because the colors are guaranteed to be continuous between blocks, at every L. The improvement from $p = 1$ to $p = 2$ is striking.

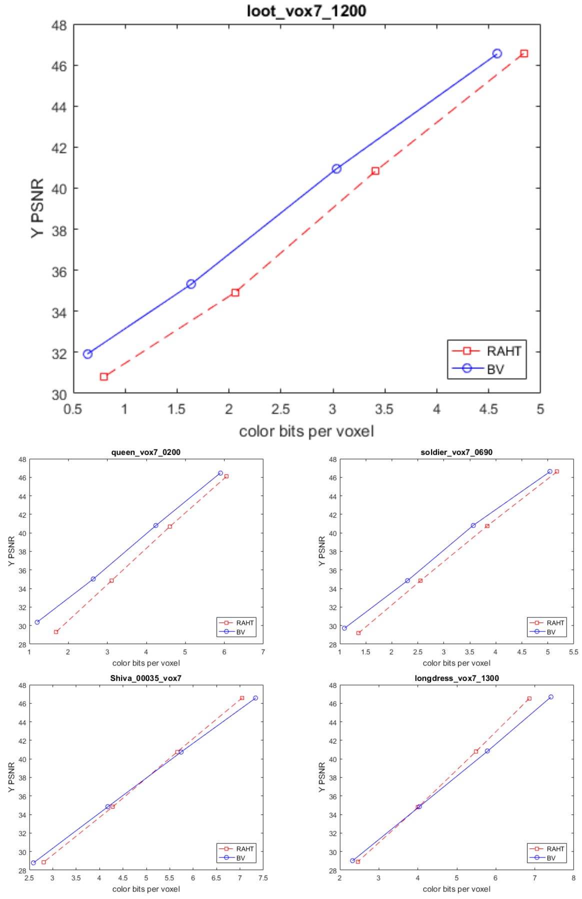


Fig. 10. Distortion-rate performance for $p = 2$ (BV) vs $p = 1$ (RAHT).

Fig. 9 shows the smoothing results quantitatively, as L is swept from 0 to 9, on all five 10-bit datasets. Each plot is a graph of Y (luma) PSNR as a function of the number of non-zero wavelet coefficients in levels $\ell < L$, for $p = 1$ (RAHT, red dashed line with squares) and $p = 2$ (BV, blue solid line with circles). The gap between the lines indicates the improvement of BV over RAHT in energy compaction gain, or transform coding gain, about 3-4 dB.

Fig. 10 shows how the improvement in energy compaction gain translates into actual coding gain, on all five 7-bit datasets. Each plot is a graph of Y PSNR as a function of the total color (Y + U + V) bits per input voxel. Here, for entropy coding, we use adaptive Run-Length Golomb-Rice (RLGR) coding ([36]) separately on each color component (Y, U, V) on each level. Though this is the preferred method of entropy coding for RAHT ([30]), as of yet there has been no study of entropy coding for BV with $p \geq 2$, so potentially the results for BV may improve (e.g., by estimating the variance of each coefficient and using it as context for the entropy coder). Regardless, the plots show that the improvement of BV over RAHT in distortion for a given bit rate can be close to 2 dB, especially at lower bit rates. Datasets with smoother color textures (*loot* and *queen*) show larger gains than datasets with medium color textures (*soldier*). Datasets with high-frequency color texture (*shiva* and *longdress*) demonstrate little coding gain of BV over RAHT, despite still having a significant energy compaction gain. The reason probably has to do with a mismatched entropy code, but requires more investigation.

VI. Conclusion

In Part I of this paper, we introduce an approach to compress the attributes of the points of a point cloud using volumetric functions. Volumetric functions are functions defined everywhere in space, not just on the discrete set of points of a point cloud. As in typical regression problems (e.g., linear regression), it is possible to extend the domain of definition of a signal from a discrete set of points to an entire Euclidean space by fitting a continuous function to the data. It is in this spirit that we propose to represent the attributes of the points of a point cloud by volumetric functions. Because a volumetric function is defined everywhere in space, even between points, attributes can be easily defined on geometry that is continuously interpolated between points. (Such geometry and its compression will be discussed in Part II of this paper.) However, even if the values of the attributes only on a discrete set of points are desired, compression of the attributes can be performed efficiently by representing the volumetric function as a linear combination of basis functions, and quantizing and compressing the basis coefficients.

We choose to represent volumetric functions in a B-spline basis of order $p \geq 1$. Though other choices could be made, the B-spline basis maps naturally to a multi-resolution, wavelet framework. Detail is thus easily preserved where necessary, and smooth regions are processed efficiently. Our multi-resolution approach fits perfectly with the octree approach for geometry processing. In our approach, the octree can be considered a *non-zero tree*, which is in some sense the complement of the zero-tree used in image processing [37], in that it compactly codes the locations of non-zero (rather than zero) wavelet coefficients of the volumetric functions.

We show that one of the most successful and practical transforms previously used for point cloud attribute compression, namely the Region-Adaptive Haar Transform (RAHT), can be regarded as a continuous B-spline wavelet transform of order $p = 1$, and thereby can be generalized to higher orders, which we relate to Bézier Volumes (BV), for higher performance.

Experimental results show that our volumetric approach (BV) with $p = 2$ outperforms RAHT with $p = 1$ by up to several dB in terms of transform coding gain (a measure of energy compaction) and up to a couple of dB in terms of actual coding gain in a compression system.

Acknowledgment

The authors would like to thank their colleagues at 8i, Charles Loop, Robert Higgs, and Gianluca Cernigliaro, for inspiring discussions leading to the formulation of this work. This work was done while the authors were with 8i.com in Wellington, NZ and Seattle, WA, USA.

References

[1] M. Krivokuća, P. A. Chou, and M. Koroteev, "A volumetric approach to point cloud compression, Part II: Geometry compression," *IEEE Trans. Image Process.*, vol. 29, pp. 2217–2229, 2020.

[2] S. Schwarz *et al.*, "Emerging MPEG standards for point cloud compression," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 9, no. 1, pp. 133–148, Mar. 2019.

[3] X. Gu, S. Gortler, and H. Hoppe, "Geometry images," *ACM Trans. Graph.*, vol. 21, no. 3, pp. 355–361, Jul. 2002. [Online]. Available: http://doi.acm.org/10.1145/566654.566589

[4] H. Briceño, P. Sander, L. McMillan, S. Gortler, and H. Hoppe, "Geometry videos: A new representation for 3D animations," in *Proc. Symp. Comput. Animation*, 2003, pp. 136–146.

[5] A. Collet *et al.*, "High-quality streamable free-viewpoint video," *ACM Trans. Graph.*, vol. 34, no. 4, p. 69, Jul. 2015. [Online]. Available: http://doi.acm.org/10.1145/2766945

[6] C. L. Jackins and S. L. Tanimoto, "Oct-trees and their use in representing three-dimensional objects," *Comput. Graph. Image Process.*, vol. 14, no. 3, pp. 249–270, Nov. 1980. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0146664X80900556

[7] D. Meagher, "Geometric modeling using octree encoding," *Comput. Graph. Image Process.*, vol. 19, no. 2, pp. 129–147, Jun. 1982. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0146664X82901046

[8] H. P. Moravec, "Sensor fusion in certainty grids for mobile robots," *AI Mag.*, vol. 9, no. 2, pp. 61–74, 1988. [Online]. Available: http://www.aaai.org/ojs/index.php/aimagazine/article/view/676

[9] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *IEEE Computer*, vol. 22, no. 6, pp. 46–57, Jun. 1989. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/2.30720

[10] K. Pathak, A. Birk, J. Poppinga, and S. Schwertfeger, "3D forward sensor modeling and application to occupancy grid based sensor fusion," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, Oct./Nov. 2007, pp. 2059–2064.

[11] R. Schnabel and R. Klein, "Octree-based point-cloud compression," in *Proc. Eurographics Symp. Point-Based Graph.*, vol. 6, Jul. 2006, pp. 111–120.

[12] Y. Huang, J. Peng, C.-C. J. Kuo, and M. Gopi, "A generic scheme for progressive point cloud coding," *IEEE Trans. Visualization Comput. Graph.*, vol. 14, no. 2, pp. 440–453, Mar./Apr. 2008.

[13] J. Kammerl, N. Blodow, R. B. Rusu, S. Gedikli, M. Beetz, and E. Steinbach, "Real-time compression of point cloud streams," in *IEEE Int. Conf. Robot. Automat.*, Saint Paul, MN, USA, May 2012, pp. 778–785,

[14] R. B. Rusu and S. Cousins, "3D is here: Point cloud library (PCL)," in *Proc. IEEE Int. Conf. Robot. Automat.*, May 2011, pp. 1–4.

[15] C. Loop, C. Zhang, and Z. Zhang, "Real-time high-resolution sparse voxelization with application to image-based modeling," in *Proc. 5th High-Perform. Graph. Conf.*, New York, NY, USA, Jul. 2013, pp. 73–79.

[16] G. M. Morton, "A computer oriented geodetic data base; and a new technique in file sequencing," IBM, Ottawa, ON, Canada, Tech. Rep., 1966. [Online]. Available: https://domino.research.ibm.com/library/cyberdig.nsf/0/0dabf9473b9c86d48525779800566a39?OpenDocument

[17] S. Lasserre and D. Flynn, *Neighbour-Dependent Entropy Coding of Occupancy Patterns in TMC3*, document m42238, ISO/IEC JTC1/SC29/WG11 MPEG, Gwangju, South Korea, Jan. 2018.

[18] R. L. de Queiroz, D. C. Garcia, P. A. Chou, and D. A. Florencio, "Distance-based probability model for Octree coding," *IEEE Signal Process. Lett.*, vol. 25, no. 6, pp. 739–742, Jun. 2018.

[19] D. C. Garcia and R. L. de Queiroz, "Intra-frame context-based Octree coding for point-cloud geometry," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2018, pp. 1807–1811.

[20] E. Pavez, P. A. Chou, R. L. de Queiroz, and A. Ortega. (2016). "Dynamic polygon clouds: Representation and compression for VR/AR." [Online]. Available: https://arxiv.org/abs/1610.00402

[21] E. Pavez and P. A. Chou, "Dynamic polygon cloud compression," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Mar. 2017, pp. 2936–2940.

[22] C. Zhang, D. Florêncio, and C. Loop, "Point cloud attribute compression with graph transform," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2014, pp. 2066–2070.

[23] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 83–98, May 2013.

[24] D. Thanou, P. A. Chou, and P. Frossard, "Graph-based motion estimation and compensation for dynamic 3D point cloud compression," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2015, pp. 3235–3239

[25] D. Thanou, P. A. Chou, and P. Frossard, "Graph-based compression of dynamic 3D point cloud sequences," *IEEE Trans. Image Process.*, vol. 25, no. 4, Apr. 2016, pp. 1765–1778.

[26] R. A. Cohen, D. Tian, and A. Vetro, "Attribute compression for sparse point clouds using graph transforms," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 1374–1378.

[27] P. A. Chou and R. L. de Queiroz, "Gaussian process transforms," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 1524–1528.

[28] R. L. de Queiroz and P. A. Chou, "Transform coding for point clouds using a Gaussian process model," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3507–3517, Jul. 2017.

[29] R. L. de Queiroz, and P. A. Chou, "Compression of 3D point clouds using a region-adaptive hierarchical transform," *IEEE Trans. Image Process.*, vol. 25, no. 8, pp. 3947–3956, Aug. 2016.

[30] G. Sandri, R. L. de Queiroz, and P. A. Chou, "Comments on compression of 3D point clouds using a region-adaptive hierarchical transform," *IEEE Trans. Image Process.*, vol. 25, no. 8, pp. 3947–3956, May 2018. [Online]. Available: https://arxiv.org/abs/1805.09146v1

[31] P. Billingsley, *Probability and Measure*, 2nd ed. Hoboken, NJ, USA: Wiley, 1986.

[32] R. M. Gray, *Probability, Random Processes, and Ergodic Properties*, New York, NY, USA: Springer-Verlag, 1988.

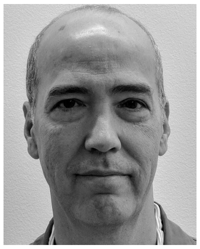
[33] A. J. Weir, *Lebesgue Integration and Measure*. Cambridge, U.K.: Cambridge Univ. Press, 1973.

[34] D. G. Luenberger, *Optimization by Vector Space Methods*. Hoboken, NJ, USA: Wiley, 1969.

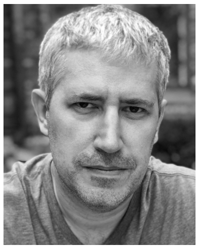
[35] E. d'Eon, B. Harrison, T. Myers, and P. A. Chou, *8i Voxelized Full Bodies—A Voxelized Point Cloud Dataset*, documents M74006 & m40059, ISO/IEC JTC1/SC29/WG1 & WG11 JPEG & MPEG, Geneva, Switzerland, Jan. 2017.

[36] H. S. Malvar, "Adaptive run-length/Golomb-Rice encoding of quantized generalized Gaussian sources with unknown statistics," in *Proc. Data Compress. Conf. (DCC)*, Mar. 2006, pp. 23–32.

[37] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. Signal Process.*, vol. 41, no. 12, pp. 3445–3462, Dec. 1993.



Philip A. Chou received the B.S.E. degree from Princeton University, the M.S. degree from the University of California at Berkeley, Berkeley, CA, USA, and the Ph.D. degree from Stanford University. He has been a Research Staff member with AT&T Bell Laboratories, Xerox Palo Alto Research Center, Microsoft, and Google. He was with startups Telesensory Systems, Speech Plus, VXtreme, and 8i. He has been an Affiliate Faculty member with Stanford University, the University of Washington, and The Chinese University of Hong Kong. He is a coeditor of a book on multimedia communication. He received best paper awards from the IEEE TRANSACTIONS ON SIGNAL PROCESSING and the IEEE TRANSACTIONS ON MULTIMEDIA and from several conferences. He has been on the organizing committees of numerous conferences and workshops, including ICASSP and ICIP. He has served on the Board of Governors and Technical Committees of the IEEE Signal Processing Society. He has served on the IEEE Fellow evaluation committees for the IEEE Signal Processing and Computer Societies. He has been an Associate Editor and/or a Guest Editor of the IEEE TRANSACTIONS ON INFORMATION THEORY, the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON MULTIMEDIA, and the *IEEE Signal Processing Magazine*.



Maxim Koroteev received the degree (Hons.) in applied mathematics and physics from the Moscow Institute of Physics and Technology (MIPT) and the Ph.D. degree. After his Ph.D. degree, he worked for around 15 years in various academic and industrial research institutions. In 2016, he joined 8i as a Researcher in computer vision. He was a data scientist for a company designing mobile autonomous vehicles. He is currently with Imatrex Inc., working on image reconstruction for a new generation CT scanner. His interests vary from statistical physics and bioinformatics to computer vision and robotics.

Maja Krivokuća received the B.E. degree (Hons.) in computer systems engineering and the Ph.D. degree from The University of Auckland, New Zealand, in 2010 and 2015, respectively. Her Ph.D. research was on developing a new and improved algorithm for the progressive compression of 3D mesh geometry using redundant dictionaries and sparse representation techniques. Since 2015, she has been with Mitsubishi Electric Research Laboratories, Cambridge, MA, USA, researching new methods for compressing large-scale 3D point clouds obtained from Mobile Mapping Systems, and with 8i, Wellington, New Zealand, researching and developing new shape compression algorithms for 3D point clouds in virtual/augmented reality applications. She has also been actively involved in contributing to the emerging MPEG Point Cloud Compression standards, since 2017. She is currently with INRIA, Rennes, France, working on research related to light field technology as part of the SIROCCO research team and the Computational Light Fields Imaging Project. Her main research interests include the areas of data compression and information theory, as well as, signal and image processing.