GAUSSIAN PROCESS TRANSFORMS

Philip A. Chou[⋆] *Ricardo L. de Queiroz*[†]

[⋆] Microsoft Research, Redmond, WA, USA (pachou@microsoft.com)
[†] Computer Science Department, Universidade de Brasilia, Brasilia, Brazil (queiroz@ieee.org)

ABSTRACT

We introduce the Gaussian Process Transform (GPT), an orthogonal transform for signals defined on a finite but otherwise arbitrary set of points in a Euclidean domain. The GPT is obtained as the Karhunen-Loève Transform (KLT) of the marginalization of a Gaussian Process defined on the domain. Compared to the Graph Transform (GT), which is the KLT of a Gauss Markov Random Field over the same set of points whose neighborhood structure is inherited from the Euclidean domain, the GPT retains up to 6 dB higher energy in the same number of coefficients or the same energy in up to 20 times fewer coefficients, and has several times higher coding gain.

***Index Terms*—** Graph signal processing, graph transform, Karhunen-Loève transform, Gaussian Process, energy compaction, coding gain, voxels, point clouds.

1. INTRODUCTION

Graph Signal Processing is an emerging field within Signal Processing. In Graph Signal Processing, the tools that are used for processing signals defined on a regular Euclidian domain are extended to processing signals defined on a graph. This extension opens signal processing to applications where the signals are defined on an irregular domain, which is naturally modeled by a graph, including social, energy, transportation, sensor and neuronal networks [1].

Many of the applications for Graph Signal Processing are for signals defined on a graph whose nodes can be considered to be embedded in a Euclidean domain. Common examples include measurements from sensor networks as well as irregular samplings of 2D and 3D signals. An example considered in this paper is the 3D voxelized point cloud shown in Fig. 1.

One of the central tools in Graph Signal Processing is the Graph Transform (GT). The Graph Transform is defined as a linear transform from signal space into the eigenspace of the graph Laplacian. The Graph Transform plays the role of the Fourier Transform in regular signal domains. Hence, the Graph Transform and its inverse are used in many signal processing operations, such as analysis, filtering, synthesis, denoising, energy compaction, and compression. The Graph Transform may be regarded as the Karhunen Loève Transform (KLT) for a stochastic signal whose covariance matrix is given by the inverse of the Graph Laplacian. It is well known that for given signal statistics the KLT is the optimal transform in terms of energy compaction and coding gain.

Fig. 1. Voxelized point cloud.

In this paper, we show that for signals defined on graphs embedded in Euclidean domains, the Graph Transform may be far from optimal in terms of energy compaction and coding gain. The simple reason is that the Graph Laplacian does not necessarily well model the statistics of the signal in this case. The signal may be better modeled as an irregular sampling, or restriction, of a stochastic signal or process whose domain of definition is the containing Euclidean domain.

For graphs embedded in Euclidean domains, we introduce the Gaussian Process Transform (GPT), which like the GT is an orthogonal linear transform of signals defined on a finite set of nodes. Also like the GT, the GPT can be regarded as a Karhunen-Loève Transform. However, the covariance matrix underlying the KLT that is the GPT is different than the covariance matrix underlying the KLT that is the GT.

We show on datasets of voxelized point clouds that the GPT has energy compaction that is up to 6 dB more efficient than the GT, reduces the number of coefficients needed for the same amount of energy compaction by up to a factor of 20, and has several times higher coding gain.

The paper is organized as follows. Section 2 couches the Graph Transform as the KLT for a Gauss Markov Random Field (GMRF). Section 3 presents two models for graphs of GMRFs embedded in a regular Euclidean domain: a popular model based on inverse distance, and a new model based on auto-regressive fitting. Section 4 introduces the Gaussian Process Transform as the KLT of a marginalized GMRF. Section 5 shows experimental results, and Section 6 concludes.

2. THE GRAPH TRANSFORM AND THE GMRF

We begin with the definition of the Graph Transform [1]. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be an *undirected* graph, where $\mathcal{V} = \{v_1, \ldots, v_n\}$ is a finite set of nodes and $\mathcal{E} = \{(v_i, v_j)\}$ is a set of edges between nodes in $\mathcal{V}$. (A graph is undirected, also called bi-directed, if $(v_i, v_j) \in \mathcal{E}$ whenever $(v_j, v_i) \in \mathcal{E}$ [2].) Let $(\mathcal{G}, W)$ be a *weighted* undirected graph, where $W = [w_{ij}]$ is a symmetric non-negative $n \times n$ matrix of weights such that

$$w_{ij} > 0 \text{ if } (v_i, v_j) \in \mathcal{E} \text{ and } w_{ij} = 0 \text{ otherwise.} \quad (1)$$

W defines the *neighborhood structure* of $\mathcal{V}$ in the graph. Let $D = [d_{ij}]$ be a diagonal matrix such that $d_{ii} = w_{ii} + \sum_j w_{ij}$ for all i. The *Graph Laplacian* of the weighted graph $(\mathcal{G}, W)$ is defined

$$L = D - W. \quad (2)$$

Let $L = \Psi^T \Lambda \Psi$ be the eigen-decomposition of L, where Λ is the diagonal matrix of eigenvalues and Ψ is the matrix whose columns are the corresponding right eigenvectors. The *Graph Transform* (or Graph Fourier Transform) is the linear transform from $\mathbb{R}^n$ to $\mathbb{R}^n$ represented by the matrix Ψ^T.

We next remark that the Graph Transform is the KLT of a corresponding Gauss Markov Random Field. A GMRF is a collection of Gaussian random variables whose joint distribution has a covariance structure given by a weighted undirected graph. Specifically, a random vector $\mathbf{x} = (X(v_1), \cdots, X(v_N))^T$ is called a GMRF with respect to the undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with mean vector μ and a symmetric positive definite precision matrix $Q = [q_{ij}]$ if and only if its density has the form [3]:

$$p(\mathbf{x}) = (2\pi)^{-\frac{n}{2}} |Q|^{\frac{1}{2}} \exp\Big(-\frac{1}{2}(\mathbf{x} - \mu)^T Q (\mathbf{x} - \mu) \Big), \quad (3)$$

$$\text{and } q_{ij} \neq 0 \Leftrightarrow (v_i, v_j) \in \mathcal{E} \text{ for all } i \neq j. \quad (4)$$

From the above definition, it is clear that a GMRF $\mathbf{x}$ is a multivariate Gaussian distribution with mean vector μ whose covariance matrix R is the inverse of precision matrix Q [4].

It is shown in [5, sec 2.1] that there is a one-to-one mapping from the set of symmetric non-negative weight matrices W satisfying (1) to the set of symmetric positive semi-definite precision matrices Q satisfying (4), through the mapping

$$q_{ij} = -w_{ij}, \text{for all } i \neq j \quad (5)$$

$$q_{ii} = \sum_{j=1}^{n} w_{ij}, \text{for all } i \quad (6)$$

and its inverse

$$w_{ij} = -q_{ij}, \text{for all } i \neq j \quad (7)$$

$$w_{ii} = \sum_{j=1}^{n} q_{ij}, \text{for all } i. \quad (8)$$

It can be shown that Q is positive semi-definite if W is non-negative, and furthermore that Q is positive definite if W has at least one self-loop (i.e., $w_{ii} > 0$ for some i) in every connected component [5, 6]. In this paper, for simplicity we deal with only the case where Q is positive definite. The case where Q is singular requires more care but results in qualitatively similar conclusions. For details see [5].

Thus it can be seen that every weighted graph $(\mathcal{G}, W)$ corresponds uniquely to a GMRF with zero mean and precision matrix Q given by (5)-(6). Moreover, it is easy to verify from (2) and (5)-(6) that

$$Q = L \quad (9)$$

and therefore $R = Q^{-1} = \Psi \Lambda^{-1} \Psi^T$. Hence Λ^{-1} is the diagonal matrix of eigenvalues of Σ and Ψ is the matrix whose columns are the corresponding eigenvectors. Thus the Graph Transform Ψ^T is the KLT of the GMRF.

3. GMRFS EMBEDDED IN A EUCLIDEAN DOMAIN

For a wide class of applications of Graph Signal Processing, the nodes $\mathbf{v}_1, \ldots, \mathbf{v}_n$ of $\mathcal{V}$ are embedded in a Euclidean domain $\mathbb{R}^N$. (Here we use boldface for the nodes to indicate that they are vectors in $\mathbb{R}^N$.) It is common in this case for the neighborhood structure of $\mathcal{V}$ to be inherited from the neighborhood structure of the containing domain. In this section, we deal with the simplified case where the containing domain is the integer lattice $\mathbb{Z}^N \subset \mathbb{R}^N$, and we examine two models for determining the neighborhood structure of $\mathbb{Z}^N$. The neighborhood structure of $\mathbb{Z}^N$ is given by a symmetric set of weights between pairs of elements of $\mathbb{Z}^N$.

3.1. The Inverse Distance Model

Let $d(\mathbf{v}, \mathbf{v}')$ denote the Euclidean distance between points in $\mathbb{R}^N$, which defines a distance $d(\mathbf{v}_i, \mathbf{v}_j)$ between nodes in a set. In the Inverse Distance model, the weights are defined

$$w_{ij} = \begin{cases} 1/d(\mathbf{v}_i, \mathbf{v}_j) & \text{for } 0 < d(\mathbf{v}_i, \mathbf{v}_j) \leq d_{\max} \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

for some distance cutoff $d_{\max}$. When applied to nodes on the integer lattice $\mathbb{Z}^N$, typical cutoffs satisfy $d^2_{\max} = 1, \ldots, N$, leading to local neighborhoods as illustrated in Fig. 2. We refer to these neighborhood models as ID$d^2_{\max}$: ID1,...,IDN.

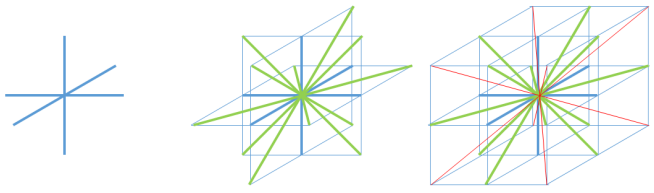

Fig. 2. Local neighborhoods in $\mathbb{Z}^3$ for $d^2_{\max} = 1, 2, 3$. Blue are $d^2 = 1$ edges; green are $d^2 = 2$ edges; red are $d^3 = 3$ edges. Respective local neighborhoods sizes are 6, 18, 26.

3.2. The Auto-Regressive Model

On the integer lattice $\mathbb{Z}^N$, let $X(\mathbf{v})$ be the stationary zero-mean Gaussian random process satisfying the auto-regressive equation

$$X(\mathbf{v}) = U(\mathbf{v}) - \sum_{\mathbf{d}\in\mathcal{N}} a_{\mathbf{d}} X(\mathbf{v}-\mathbf{d}), \tag{11}$$

where $U(\mathbf{v})$ is an iid $N(0,\sigma_u^2)$ Gaussian white noise, $a_{\mathbf{d}}$ is a coefficient for offset $\mathbf{d}$, and $\mathcal{N} = \{\mathbf{d} : 0 < |\mathbf{d}| \le d_{\max}\}$ is a set of offsets in a neighborhood around the current point $\mathbf{v}$.

With the proper boundary conditions, the above equation can be used to generate $X(\cdot)$ from $U(\cdot)$. In particular,

$$\mathbf{x} = A^{-1}\mathbf{u}, \tag{12}$$

where $\mathbf{x} = [X(\mathbf{v}) : \mathbf{v} \in \mathbb{Z}_m^N]$ and $\mathbf{u} = [U(\mathbf{v}) : \mathbf{v} \in \mathbb{Z}_m^N]$ are vectors truncated to blocks of size $\ell = m^N$ and $A = [a_{ij}]$ is an $\ell \times \ell$ matrix whose (i,j)th entry equals 1 if $i = j$, equals $a_{\mathbf{d}}$ if $\mathbf{v}_i - \mathbf{d} = \mathbf{v}_j$ for $\mathbf{d} \in \mathcal{N}$, and equals 0 otherwise.

Since the probability density of $\mathbf{u}$ is

$$p(\mathbf{u}) = \frac{1}{(2\pi\sigma_u^2)^{\ell/2}} \exp\left(-\frac{1}{2\sigma_u^2}\mathbf{u}^T\mathbf{u}\right), \tag{13}$$

the probability density of $\mathbf{x}$ can be written

$$\begin{aligned} p(\mathbf{x}) &= \frac{|A|}{(2\pi\sigma_u^2)^{\ell/2}} \exp\left(-\frac{1}{2\sigma_u^2}\mathbf{x}^T A^T A\mathbf{x}\right) && (14)\\ &= \frac{1}{(2\pi)^{\ell/2}|R|^{1/2}} \exp\left(-\frac{1}{2}\mathbf{x}^T R^{-1}\mathbf{x}\right), && (15)\end{aligned}$$

where $R = \sigma_u^2(A^TA)^{-1}$ is the covariance matrix of $\mathbf{x}$. It can be seen the coefficients q_{ij} in the precision matrix $Q = R^{-1} = \sigma_u^{-2}(A^TA)$ of $\mathbf{x}$ are the coefficients of the autocorrelation of the coefficients $\{a_{\mathbf{d}}\}$ in A (scaled by σ_u^{-2}). Indeed since a_{ij} can be non-zero only if $|\mathbf{v}_i - \mathbf{v}_j| \le d_{\max}$, q_{ij} can be non-zero only if $|\mathbf{v}_i - \mathbf{v}_j| \le 2d_{\max}$. Solving for W using (7)-(8) gives a local neighborhood structure on $\mathbb{Z}^N$. We refer to these neighborhood models as AR$d_{\max}^2$.

The coefficients $\{a_{\mathbf{d}}\}$ that best fit a set of data $\mathbf{x}$ can be determined in principle by maximizing the log likelihood $\log p(\mathbf{x})$ over the variables $\{a_{\mathbf{d}}\}$. However, this is an "extremely non-linear" problem that is difficult to solve [7]. A close approximation is to maximize the pseudo log likelihood

$$-\frac{1}{2}\mathbf{x}^T R^{-1}\mathbf{x}, \tag{16}$$

or equivalently, minimize the squared prediction error $||\mathbf{u}||^2$ over the variables $\{a_{\mathbf{d}}\}$, where the prediction of $X(\mathbf{v})$ is $-\sum_{\mathbf{d}\in\mathcal{N}} a_{\mathbf{d}} X(\mathbf{v}-\mathbf{d})$ and therefore the prediction error is

$$U(\mathbf{v}) = X(\mathbf{v}) + \sum_{\mathbf{d}\in\mathcal{N}} a_{\mathbf{d}} X(\mathbf{v}-\mathbf{d}). \tag{17}$$

The mean squared prediction error can be minimized using the principle of orthogonality in Hilbert spaces, so that the error $U(\mathbf{v})$ in the approximation of $X(\mathbf{v})$ by its projection onto a span of basis vectors $\{X(\mathbf{v}-\mathbf{d})\}$ must be orthogonal to each of the basis vectors. That is, for all $\mathbf{d} \in \mathcal{N}$,

$$\begin{aligned} 0 &= \langle U(\mathbf{v}), X(\mathbf{v}-\mathbf{d})\rangle \\ &= \langle X(\mathbf{v}), X(\mathbf{v}-\mathbf{d})\rangle + \\ &\quad \sum_{\mathbf{d}'\in\mathcal{N}} a_{\mathbf{d}'}\langle X(\mathbf{v}-\mathbf{d}'), X(\mathbf{v}-\mathbf{d})\rangle. \end{aligned} \tag{18}$$

Using $\langle X(\mathbf{v}-\mathbf{d}'), X(\mathbf{v}-\mathbf{d})\rangle = R_{xx}(\mathbf{d}-\mathbf{d}')$, where the covariance function $R_{xx}(\mathbf{d}-\mathbf{d}')$ is determined empirically from the data $\mathbf{x}$, the variables $\{a_{\mathbf{d}}\}$ can be solved from the $|\mathcal{N}|$ normal equations,

$$R_{xx}(\mathbf{d}) = -\sum_{\mathbf{d}'\in\mathcal{N}} a_{\mathbf{d}'} R_{xx}(\mathbf{d}-\mathbf{d}') \tag{19}$$

for all $\mathbf{d} \in \mathcal{N}$.

4. THE GAUSSIAN PROCESS TRANSFORM

In the previous section we considered how to find a neighborhood structure for the integer lattice $\mathbb{Z}^N$. For nodes that lie in a subset $\mathcal{V} \subset \mathbb{Z}^N$, it is common practice for the neighborhood structure of $\mathcal{V}$ to be inherited from the neighborhood structure of $\mathbb{Z}^N$. For example, as shown in Fig. 3, nodes in a subset of an image are neighbors in the subset if and only if they are neighbors in the image. To be precise, if $\mathbf{v}_i$ and $\mathbf{v}_j$ are both in $\mathcal{V}$, and w_{ij} is the weight between them in $\mathbb{Z}^N$, then w_{ij} is also the weight between them in $\mathcal{V}$. This induces a weighted graph $((\mathcal{V},\mathcal{E}),W)$ and a corresponding GMRF on $\mathcal{V}$. The KLT of this GMRF is the Graph Transform (GT).

In this paper we introduce an alternative approach to defining a transform for signals living on $\mathcal{V}$. The concept is simple. By regarding $[X(\mathbf{v}) : \mathbf{v} \in \mathbb{Z}^N]$ as a stationary Gaussian random process, indeed as a GMRF with respect to the same neighborhood structure on $\mathbb{Z}^N$ determined in Section 3, we find the marginal distribution of the random vector $[X(\mathbf{v}) : \mathbf{v} \in \mathcal{V}]$. The KLT with respect to the correlation matrix of this random vector is a transform we call the Gaussian Process Transform (GPT). Both the GT and the GPT are orthogonal linear transforms for signals living on $\mathcal{V}$.

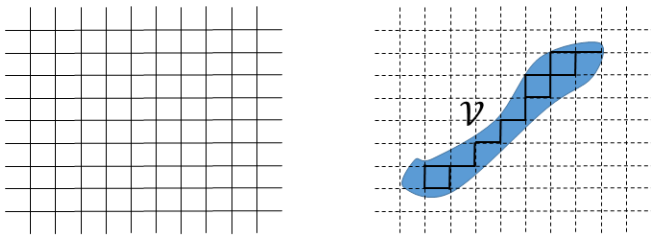

Fig. 3. Left: Neighborhood structure in $\mathbb{Z}^2$ for $d = 1$. Right: Inherited neighborhood structure in $\mathcal{V} \subset \mathbb{Z}^2$.

The computational difference between the two transforms can be seen clearly by partitioning the random vector $\mathbf{x} = [X(\mathbf{v}) : \mathbf{v} \in \mathbb{Z}^N]$ into two pieces, $\mathbf{x}_1 = [X(\mathbf{v}) : \mathbf{v} \in \mathcal{V}]$ and $\mathbf{x}_2 = [X(\mathbf{v}) : \mathbf{v} \in \mathbb{Z}^N \setminus \mathcal{V}]$, and likewise partitioning its covariance and precision matrices,

$$\begin{bmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{bmatrix} = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix}^{-1} \tag{20}$$

$$= \begin{bmatrix} S_{11}^{-1} & -Q_{11}^{-1}Q_{12}S_{22}^{-1} \\ -Q_{22}^{-1}Q_{21}S_{11}^{-1} & S_{22}^{-1} \end{bmatrix}, \tag{21}$$

the latter following from the block matrix inverse formula with $S_{11} = Q_{11} - Q_{12}Q_{22}^{-1}Q_{21}$ and $S_{22} = Q_{22} - Q_{21}Q_{11}^{-1}Q_{12}$ being the Schur complements of Q_{11} and Q_{22} [8]. Thus

$$R_{11}^{-1} = S_{11} = Q_{11} - Q_{12}Q_{22}^{-1}Q_{21}, \tag{22}$$

which is also known as the Kron reduction [5, 6]. The eigenvectors of Q_{11} form the Graph Transform, while the eigenvectors of R_{11}, or equivalently the eigenvectors of R_{11}^{-1}, form the Gaussian Process Transform. Note that Q_{11} is generally sparse while both R_{11} and R_{11}^{-1} are generally dense. The GT is based on considering $\mathcal{V}$ as the nodes of a graph, while the GPT is based on considering the variables on $\mathcal{V}$ as samples of a process on a Euclidean domain.

5. EXPERIMENTAL RESULTS

We compare GPT and GT on the voxelized point cloud dataset shown in Fig. 1. This dataset comprises 207242 occupied voxels within a cubic volume 512 voxels per side. Each occupied voxel has a color in YUV space. We remove the mean of Y and treat it as a signal X living on the set of occupied voxels $\mathcal{V} \subset \mathbb{Z}^3$. We measure the autocovariance function $R_{xx}(\mathbf{d})$ of X empirically by assuming $R_{xx}(\mathbf{d}) = R_{xx}(d)$, where $d = |\mathbf{d}|$, and for each value of d averaging the product $X(\mathbf{v}_i)X(\mathbf{v}_j)$ over all pairs $(\mathbf{v}_i, \mathbf{v}_j)$ for which both $\mathbf{v}_i$ and $\mathbf{v}_j$ are in $\mathcal{V}$, and $|\mathbf{v}_i - \mathbf{v}_j| = d$.

We partition the voxel cube of size 512^3 into 64^3 blocks each of size $\ell = 8^3$. For this dataset, only 3025 blocks are occupied, yielding an average of 69 occupied voxels per occupied block. In each occupied block, we treat the occupied voxels as a set of nodes $\mathcal{V} \subseteq \mathbb{Z}_8^3$. We determine six different neighborhood structures on $\mathbb{Z}_8^3$, using both Inverse Distance (ID) and Auto-Regressive (AR) models and $d_{\max}^2 = 1, 2, 3$: ID1, ID2, ID3, AR1, AR2, AR3. Each neighborhood structure induces possible precision matrices Q_{11} and $S_{11} = Q_{11} - Q_{12}Q_{22}^{-1}Q_{21}$ for the signal $[X(\mathbf{v}) : \mathbf{v} \in \mathcal{V}]$. Eigenvectors of the former are the GT, while eigenvectors of the latter are the GPT. The dot product of each eigenvector with the signal is a coefficient whose "frequency" equals the eigenvalue associated with the eigenvector. The coefficients from all the occupied blocks are sorted by increasing frequency.

Fig. 4 shows the percentage of energy of the signal captured in the first (low-frequency) fraction of coefficients, under various conditions (model ID or AR and size $d_{\max}$). Higher $d_{\max}$ is not necessarily better, probably because of boundary effects in our small 8^3 block. However, GPT has far greater energy compaction than GT, regardless of condition. The first 1% of the GPT coefficients (under AR1 or AR2 conditions) captures over 6 dB more energy than the first 1% of the GT coefficients (under the ID3 condition — the most popular for 3D voxels [9]). Alternatively, 93% of the signal energy is captured by fewer GPT coefficients than 5% of the GT coefficients. Table 1 shows the transform coding gain (ratio of arithmetic to geometric mean of the energies in each band [10]) for each condition. The GPT has a coding gain several times that of the GT, regardless of condition.

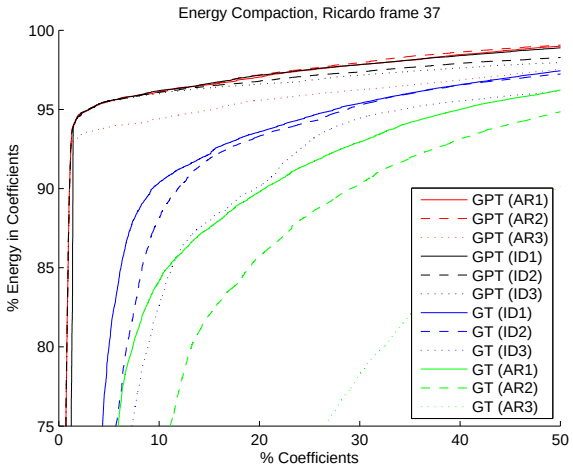

Fig. 4. Energy compaction.

Table 1. Transform coding gain.

Model	Inverse Distance			Auto-Regressive		
$d_{\max}^2$	1	2	3	1	2	3
GT	15.1	14.8	13.0	9.7	6.7	3.6
GPT	**71.4**	**64.4**	**64.5**	**72.7**	**73.1**	**51.7**

6. CONCLUSION

This paper presents clear evidence from the point of view of energy compaction and transform coding gain that for signals defined on points embedded in a Euclidean domain, modeling the signal as samples of a stochastic process may be far better than modeling it as a signal on a graph whose neighborhood structure is inherited from the domain. This has implications for transform coding of many natural signals. In future work, we will examine how this new perspective can improve the coding efficiency of voxelized point clouds. Based on the transform coding gains seen in this paper, we expect a multifold reduction in bit rate.

7. REFERENCES

[1] 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.

[2] J. Edmonds and E. L. Johnson, "Matching: A well-solved class of integer linear programs," in *in: Combinatorial structures and their applications (Gordon and Breach)*, 1970.

[3] H. Rue and L. Held, *Gaussian Markov Random Fields: Theory and Applications*, Chapman and Hall/CRC, 2005.

[4] Y. Dodge, *The Oxford Dictionary of Statistical Terms*, Oxford University Press, 2003.

[5] C. Zhang, D. Florencio, and P.A. Chou, "Graph signal processing - a probabilistic framework," Tech. Rep. MSR-TR-2015-31, April 2015.

[6] F. Dörfler and F. Bullo, "Kron reduction of graphs with applications to electrical networks," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 60, no. 1, pp. 150–163, 2013.

[7] J.D. Markel and A.H. Gray, Jr., *Linear Prediction of Speech*, Springer Verlag, Berlin, 1976.

[8] T. Kailath, *Linear Systems*, Prentice-Hall, Englewood Cliffs, N.J., 1980.

[9] C. Zhang, D. Florêncio, and C. Loops, "Point cloud attribute compression with graph transform"," in *Proc. IEEE Int. Conference on Image Processing*, Paris, France, Sept 2014.

[10] A. Gersho and R.M. Gray, *Vector Quantization and Signal Compression*, Kluwer, 1992.