PATH CODING ON GEOMETRIC PLANAR GRAPH FOR 2D / 3D VISUAL DATA PARTITIONING

Weihang Liao [*], *Gene Cheung* [*], *Wei Hu* [#]

[*] National Institute of Informatics, [#] Peking University

ABSTRACT

New visual media types like light field images and point clouds are often irregularly sampled data in 2D or 3D space. While coding of irregularly sampled data has enjoyed recent progress due to the advent of graph-based coding tools like graph transforms and wavelets, the absence of efficiently coded side information (SI) limits the adaptivity and hence the coding efficiency of these tools. In this paper, we present a general methodology to code a path through a geometric planar graph—a generalization of a contour in a 2D image—to partition irregular samples in 2D / 3D space. The encoded partition boundary can subsequently be used to assign appropriate weights of edges connecting samples across the boundary for more efficient graph-based coding.

Specifically, for the 2D case, we first construct a graph based on a Voronoi map computed from the irregularly sampled locations. We show that the Voronoi map boundaries represent the best local unbiased estimator of edge directions in the original continuous 2D signal. For the 3D case, we project a local window of 3D points onto a best-fitted plane, then construct a planar graph based on a Voronoi map as done in the 2D case. The local window is then shifted for the next iteration in the direction of the coded path. For a given constructed graph, knowing the maximum degree of each node, we design an alphabet to designate outgoing edges and assign a probability for each using linear regression of past path segment and Von Mises distribution with locally optimized parameters. Given assigned probabilities, arithmetic coding is used to encode a sequence of symbols in the alphabet into a bitstream. Experimental results show that our proposed method outperforms state-of-the-art contour coding on 2D grid, and uniform probability assignment in the 3D case.

Index Terms— Path coding, graph cut, image compression

1. INTRODUCTION

The advent of new image capturing devices like light field cameras and active depth sensors means that the acquired visual data are no longer restricted to pixels on a regular 2D grid in a traditional image, but irregularly placed samples in 2D or 3D space. One example is light field data where the demosaicking step is postponed to the decoder to reduce data volume required for compression, so that available pixels for coding in constructed sub-aperture images are sparse and irregularly distributed [1]. Another example is color attributes (RGB components) on a 3D point cloud, which are irregularly placed samples in 3D space.

With recent advance in *graph signal processing* (GSP) [2], using graph-based coding tools like graph Fourier transforms [3–5] and wavelets [6, 7], the aforementioned irregular samples can now be coded as sparse coefficients of frequency components in appropriately defined graphs that reflect *geometric* structure of the irregularly sample kernels—*i.e.* Euclidean distances between pairs of samples. However, with the absence of efficiently coded *side information* (SI) that designates the *photometric* structure [8]—*e.g.* the boundaries between foreground and background pixels in an irregularly sampled depth image—the adaptivity and thus coding efficiency of these graph-based coding tools are still limited.

Towards the goal of efficient SI coding that reflects the underlying photometric structure for improved graph-based compression of irregularly placed samples in 2D / 3D space, in this paper we propose a general methodology to encode a path through a geometric planar graph. Generalizing contour coding in 2D images [9, 10] to irregular sample kernels, the coded path partitions samples in 2D / 3D space into two disjoint subsets, so that edges connecting samples across the partition boundaries can be assigned more appropriate edge weights for graph-based compression. Specifically, for the 2D case we first construct a graph based on a Voronoi map computed from the irregularly placed samples [11]. We show that the Voronoi map boundaries represent the best local unbiased estimator of edge directions in the pre-sampled original continuous 2D signal. For the 3D case, we project a local window of 3D points onto a best-fitted plane, then construct a planar graph based on a Voronoi map as done in the 2D case. The local window is then shifted for the next iteration in the direction of the coded path.

For a given graph, knowing the maximum degree of each node, we design an alphabet to designate outgoing edges, and, using a linear regression model and Von Mises distribution (with locally optimized parameters), assign probabilities to each edge. Finally, arithmetic coding [12] is used to encode a sequence of symbols representing the path into a bitstream. Experimental results show that our proposed method outperforms state-of-the-art contour coding on 2D grid, and uniform probability assignment in the 3D case. *To the best of our knowledge, we are the first in the literature to efficiently encode a discrete path through a geometric planar graph.*

2. RELATED WORK

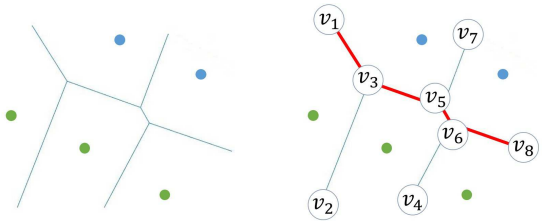
The coding of a path through a 2D grid of pixels (image)—also called a *contour*—has been well studied in previous works. Most of the contour coding works [9, 13–15] first map the contour into a chain code, like the 4-connected Freeman chain code [16], which uses the alphabet $\{0, 1, 2, 3\}$ to denote the four possible absolute directions from current node to the following node. The Differential Chain Code (DCC) [14] uses a three-symbol alphabet instead $\{0, 1, 2\}$ corresponding to direction `left`, `straight` and `right` relative to the previously coded direction. Then, an entropy encoding engine such as Huffman [17] or arithmetic coding [18] is used to code the symbols of chain code. In this paper, we extend our previous contour coding work [9] to irregularly sampled kernels, where the notion of a contour on a 2D grid is generalized to a path on a geometric planar graph. To the best of our knowledge, we are the first in the literature to tackle this problem.

3. GRAPH CONSTRUCTION

We first describe how we construct a graph from irregular samples in 2D and 3D space in order, so that an appropriately chosen path through the graph would translate to a partitioning of samples into two disjoint subsets.

3.1. Graph Construction for 2D Data

For 2D data, we assume that the irregularly sampled locations are known both at the encoder and decoder, and only the sample values are unknown and need to be encoded; this is the case for light field sub-aperture image coding without demosaicking in [1]. Given irregular sample locations on a 2D plane, we first build a *Voronoi map* [11]: partitioning of a finite 2D plane of interest into different *Voronoi cells* (one cell for each sample), where points in a cell are 2D points that are closest in Euclidean distance to the cell's sample location. See Fig. 1(a) for an illustration of Voronoi cells in 2D.



(a) Voronoi map (b) A contour acroos data

Fig. 1. (a) shows Voronoi cells for five 2D irregularly sampled data points. (b) shows a path through a graph (constructed using Vonronoi cell boundaries) that partitions the samples into two disjoint subsets.

We now interpret the Voronoi map as a graph $\mathcal{G}$, where cell boundaries are edges $\mathcal{E}$ and intersections of cell boundaries are nodes $\mathcal{V}$. Fig. 1(b) shows an example of a constructed graph from Voronoi cells in Fig. 1(a). The edges constructed from cell boundaries are the *best local unbiased direction estimators* given available samples. In other words, if two neighboring samples belong to two different partitions (*e.g.*, foreground and background), the true partition boundary in continuous space is best estimated as the line with equal distance to the two samples—the Voronoi cell boundary. Practically, this means that the geometric locations of the endpoints of constructed edges provide the most reliable information in predicting the next edge in a path on the graph.

For each node in the graph with a 2D coordinate, we assign a label $v \in \{1, \ldots, |\mathcal{V}|\}$ in a raster scan order: samples are sorted in ascending order in their x-coordinates, and for samples with the same x-coordinates, they are sorted in ascending order in their y-coordinates. The same graph construction procedure can be replicated at the decoder.

Since each edge in $\mathcal{E}$ separates two neighboring samples in two adjacent cells, a path on graph $\mathcal{G}$ represents a *graph cut* that partitions the original 2D data into two disjoint subsets. See Fig. 1(b) for an illustration.

3.2. Graph Construction for 3D Data

In general, a path in a graph connecting 3D points does not represent a partition of points into two subsets. Here, we assume more specifically that the 3D point cloud are samples of a 2D surface in 3D space, *e.g.*, surface of a human body. For a local subset of points then, we approximate their representation as a planar graph, and a cut through the approximated planar graph would divide the points into two disjoint subsets.

Specifically, for a given set of irregularly located samples in 3D space, we first select a subset of points and project them onto a best-fitted 2D plane. We then compute a Voronoi map and corresponding graph as done previously in the 2D case. We assign labels $v \in \{1, \ldots, |\mathcal{V}|\}$ to nodes in a raster scan order similar to the 2D case: in ascending order in x-, y- and z-coordinates. Like the 2D case, the same graph construction procedure can be replicated at the decoder.

3.3. Chain Code Representation of a Path in Graph

We can represent a path in a graph $\mathcal{G}$ as a sequence of symbols chosen from a finite alphabet as follows. We first specify a starting node i in $\mathcal{G}$ by encoding its label. Denote the degree of a node i by D_i and the set of neighboring nodes by $\mathcal{N}_i$, where $|\mathcal{N}_i| = D_i$. Generalizing previously proposed chain code [16] that specifies *absolute* directions, we can specify the first edge stemming from node i as follows: i) we first sort neighbors $\mathcal{N}_i$ in ascending order of their labels, then ii) encode a symbol a_i chosen from alphabet $\mathcal{A}_i = \{1, \ldots, D_i\}$, specifying that the a_i-th neighbor in the sorted order is the destination node.

For each subsequent edge (j, k) along the path following edge (i, j), generalizing previously proposed differential chain code (DCC) [14] that specifies *relative* directions, we can perform a similar encoding procedure as above: i) first sort neighbors $\mathcal{N}_j \setminus \{i\}$ in ascending order of their labels, then ii) encode a symbol a_j chosen from alphabet $\mathcal{A}_j = \{1, \ldots, D_j - 1\}$, specifying that the a_j-th neighbor in the sorted order is the destination node.

Like our previous work on contour coding on 2D images [9, 10], the coding efficiency of this chain of symbols (assuming arithmetic coding [12] is used as the entropy coding engine) depends heavily on how probabilities are assigned to symbols a_i in alphabet $\mathcal{A}_i$ for each hop. We discuss our proposed context model for this purpose next.

4. PATH CODING ON GEOMETRIC GRAPH

To assign probabilities for symbols in alphabet $\mathcal{A}_i$ for efficient arithmetic coding [12], we assume that each node in graph $\mathcal{G}$ carries useful geometric information (hence the name geometric graph), and a local neighborhood of nodes reside on a line approximately, so that a linear regression model built from a local set of samples can be used to estimate probabilities. We describe our scheme in details below.

4.1. Linear Prediction

For either 2D or 3D case, given a set of K coordinates from K previous coded nodes on the path, $\mathbf{p}_i$, $i = 1, \ldots, K$, where each coordinate $\mathbf{p}_i \in \mathbb{R}^r$, $r \in \{2, 3\}$, we seek to derive a best-fitted line:

$$\mathbf{p} = \mathbf{s} + t\mathbf{v} \tag{1}$$

where $\mathbf{s}$ is a point on the line, $\mathbf{v}$ is a unit vector, t is a scaling parameter, $-\infty < t < \infty$, to construct the line. Because the mean of the K points must reside on the fitted line, we compute $\mathbf{s}$ to be the mean of the K coordinates $\mathbf{p}_i$. See Fig. 2 for illustration.

To find the optimal $\mathbf{v}$, we first compute mean-removed vectors $\mathbf{x}_i = \mathbf{p}_i - \mathbf{s}$, then maximize the sum of inner-products between each

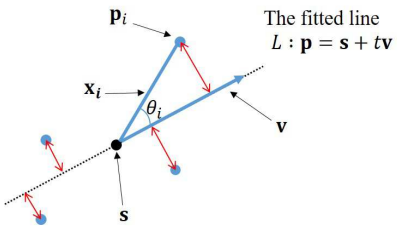


Fig. 2. Linear regression given K nodes. We seek to minimize the projected distance from each point to the fitted line, which equals to maximization of the inner-product.

$\mathbf{x}_i$ and $\mathbf{v}$, resulting in the following objective:

$$\max_{\mathbf{v} \mid \|\mathbf{v}\|_2^2=1} \sum_{i}^{K} (\mathbf{x}_i^\top \mathbf{v})^2 = \max_{\mathbf{v} \mid \|\mathbf{v}\|_2^2=1} \|\mathbf{X}\mathbf{v}\|_2^2 \tag{2}$$

where $\mathbf{X}$ is a $K \times r$ matrix composed of $\mathbf{x}_i^T$ as rows.

Since $\|\mathbf{X}\mathbf{v}\|_2^2 = \mathbf{v}^\top \mathbf{X}^\top \mathbf{X}\mathbf{v}$, given constraint $\|\mathbf{v}\|_2^2 = 1$, (2) is the *Rayleigh quotient* for matrix $\mathbf{X}^\top \mathbf{X}$, and the solution is the eigenvector corresponding to the largest eigenvalue $\lambda_{\max}$. Note that $\mathbf{X}^\top \mathbf{X}$ is of dimension $r \times r$, and since r is at most 3, eigen-decomposition of a small matrix is not computation-expensive.

4.2. Probability Assignment

At a given node i there are $D_i - 1$ outgoing edges for the next hop, with corresponding designation coordinates $\mathbf{q}_j$, $j \in \{1, \ldots, D_i - 1\}$. For each candidate coordinate $\mathbf{q}_j$, we include coordinates of the previous $K - 1$ nodes in the path to construct matrix $\mathbf{X}$, then compute (2) for a vector $\mathbf{u}_j$. We can then compute the angles α_j between $\mathbf{u}_j$ and $\mathbf{v}$ computed from the coordinates of the previous K nodes in the path.

Intuitively, a smaller angle α_j means a higher probability should be assigned to coordinate $\mathbf{q}_j$. For this assignment, we utilize the von Mises probability distribution, which maps an angle α, $-\pi \leq \alpha \leq -\pi$, to a positive real number:

$$f(\alpha \mid \mu, \kappa) = \frac{e^{\kappa \cos(\alpha)}}{2\pi I_0(\kappa)} \tag{3}$$

where $I_0(.)$ is the modified Bessel function of order 0, and $\frac{1}{\kappa}$ is the variance of the distribution. A larger κ would induce a sharper distribution, and thus can be interpreted as a measure of confidence in the probability estimation.

Eigen-decomposition of data matrix $\mathbf{X}^\top \mathbf{X}$ can be interpreted as *principal component analysis* (PCA), and thus the smallest eigenvalue $\lambda_{\min}$ of matrix $\mathbf{X}^\top \mathbf{X}$ can be interpreted as the variance from the main component, *i.e.*, the reliability of the fitted line. Thus we can adaptively define κ as a function of $\lambda_{\min}$. Specifically, if $\lambda_{\min}$ is small, then the linear model fits the data $\mathbf{x}_i$ well and we can be confident about its estimated direction $\mathbf{v}$. κ should be set large in this case. Mathematically, we define κ as follows:

$$\kappa = c \cdot e^{-\frac{\lambda_{\min}}{\sigma^2}} \tag{4}$$

where c and σ are parameters. In our experiments, we optimize c and σ using a set of training data.

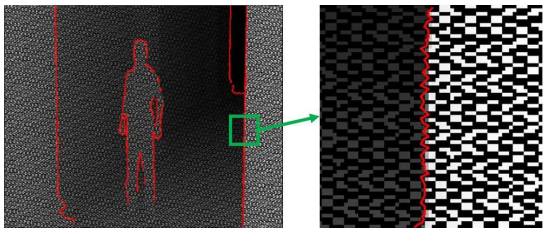


Fig. 3. Example depth map for sequence *Dancer* (1280×720). The white and gray blocks are pixels with different gray levels, and the black component means there are no observed pixels.

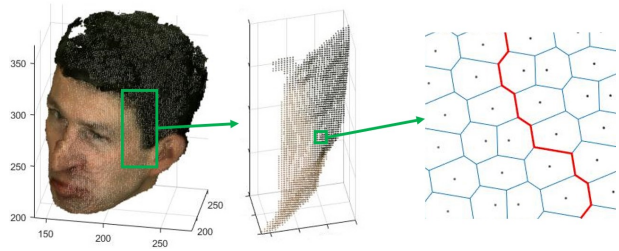


Fig. 4. Subset of point cloud data *Andrew*. We use a sliding window to traverse the point cloud; for each window, we project the points to a 2D plane and construct a Voronoi map. The red line is the path on the Voronoi map which corresponds to a boundary on the 2D surface in 3D space.

5. EXPERIMENTATION

5.1. Experimental Setup

For the 2D visual data simulation, we use an irregularly sampled depth map. The original depth map is from a color-plus-depth video sequence from the MCL-3D database [19]. Then we randomly sample it to generate a sparsely distributed depth map, as shown in Fig. 3.

For 3D visual data simulation, we use a point cloud data provided by MPEG [20]. In our implementation, we select a subset (4000 points) of the original point cloud as shown in Fig. 4. We use a shifting window with size of $4 \times 4 \times 4$ to traverse the point cloud. Notice that a larger window size means more 3D points would be mapped to a 2D plane, resulting in a worse planar approximation. For each iteration, we set the center of next window to the end point of coded path in current window. Thus we ensure some overlap between consecutive windows, which guarantee that we have a sufficiently long coded path segment for linear regression.

For the coding parameters, we use 5 previous coded nodes for linear prediction, and set $c = 0.5$, $\sigma = 0.05$ in equation (4).

5.2. Experimental Results

To the best of our knowledge, there is no existing work on path coding in planar graphs. Hence for the 2D case, we compare our method with one previous work employing AEC [9] and DCC utilized in MPEG-4 standard [21], where the path is expanded to a full 2D grid contour representation. We also compare with a uniform probability distribution—where the same probability is assigned to each outgoing edge—to demonstrate how important proper probability assignment is to coding efficiency. The coding results are shown in Tab. 1.

Table 1. Symbol count and bit consumption of different methods for 2D case

	Total Num. of Symb.	Total Bits	Bits per Symb.
DCC [21]	1608	1929	1.20
AEC [9]	1608	951	0.59
Uniform	989	1507	1.52
Proposal	989	896	0.91

Table 2. Symbol count and bit consumption of different methods for 3D case

	Total Num. of Symb.	Total Bits	Bits per Symb.
Uniform	195	203	1.04
Proposal	195	171	0.87
DGU	292	301	1.03
DGP	292	285	0.98

We observe that our proposed method outperforms [21], [9] and uniform distribution in terms of total bit count, although our method has a higher bits / symbol than [9]. This is because Voronoi map is built from an irregularly sampled kernel, thus a path through it is harder to predict. Using this graph, however, we can represent the same contour using fewer symbols than a full 2D grid, resulting in a smaller total bit consumption.

For the 3D case, we add another comparison besides uniform distribution. We first build a triangular mesh for the point cloud data by ball-pivoting algorithm [22], then draw a dual graph from the mesh, similarly done in previous mesh segmentation works [23, 24]. The results are shown in Tab. 2, where DGU and DGP mean dual graph with uniform probability distribution and linear prediction model, respectively.

We observe that for the 3D case, our method outperforms the uniform distribution case. For the path on dual graph, the result with prediction is close to the uniform distribution. This is because the mesh is quite irregular in triangle construction, resulting in elongated triangles, and edges in the dual graph are not representative of the directions of the segmented signal. This experimentally validates our claim that the boundaries of Voronoi cells are the best local unbiased direction estimators, leading to better path coding performance.

6. CONCLUSION

We proposed an efficient method to code a path through a geometric planar graph, which is a generalization of contour coding on a 2D full grid image. The path can be used to partition irregularly sampled data in both 2D and 3D case. The key idea is to build a Voronoi map based on the irregularly sampled kernel, whose boundaries represent the best local unbiased estimators of the directions of the segmented signal. A path prediction scheme based on linear regression and Von Mises distribution with locally optimized parameters is proposed. Experimental results show that our proposed scheme outperforms state-of-the-art contour coding schemes on 2D grid, and naïve schemes in 3D point clouds.

7. REFERENCES

[1] Y.-H. Chao, G. Cheung, and A. Ortega, "Pre-demosiac light field compression using graph lifting transform," in *IEEE International Conference on Image Processing*, Beijing, China, September 2017.

[2] 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," in *IEEE Signal Processing Magazine*, vol. 30, no.3, May 2013, pp. 83–98.

[3] W. Hu, G. Cheung, A. Ortega, and O. Au, "Multi-resolution graph Fourier transform for compression of piecewise smooth images," in *IEEE Transactions on Image Processing*, vol. 24, no.1, January 2015, pp. 419–433.

[4] Y.-H. Chao, A. Ortega, W. Hu, and G. Cheung, "Edge-adaptive depth map coding with lifting transform on graphs," in *31st Picture Coding Symposium*, Cairns, Australia, May 2015.

[5] W. Hu, G. Cheung, and A. Ortega, "Intra-prediction and generalized graph Fourier transform for image coding," in *IEEE Signal Processing Letters*, vol. 22, no.11, November 2015, pp. 1913–1917.

[6] S. K. Narang and A. Ortega, "Multi-dimensional separable critically sampled wavelet filterbanks on arbitrary graphs," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, Kyoto, Japan, March 2012.

[7] ——, "Compact support biorthogonal wavelet filterbanks for arbitrary undirected graphs," in *IEEE Transactions on Signal Processing*, vol. 61, no.19, 2013, pp. 4673–4685.

[8] P. Milanfar, "A tour of modern image filtering," in *IEEE Signal Processing Magazine*, vol. 30, no.1, January 2013, pp. 106–128.

[9] I. Daribo, D. Florencio, and G. Cheung, "Arbitrarily shaped motion prediction for depth video compression using arithmetic edge coding," *IEEE Transactions on Image Processing*, vol. 23, no. 11, pp. 4696–4708, 2014.

[10] A. Zheng, G. Cheung, and D. Florencio, "Context tree based image contour coding using a geometric prior," in *IEEE Transactions on Image Processing*, vol. 26, no.2, February 2017, pp. 574–589.

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

[12] I. Witten, R. Neal, and J. Cleary, "Arithmetic coding for data compression," in *Communications of the ACM*, vol. 30, no.6, June 1987, pp. 520–540.

[13] C.-C. Lu and J. G. Dunham, "Highly efficient coding schemes for contour lines based on chain code representations," *IEEE Transactions on Communications*, vol. 39, no. 10, pp. 1511–1514, 1991.

[14] Y. K. Liu and B. Zalik, "An efficient chain code with huffman coding," *Pattern Recognition*, vol. 38, no. 4, pp. 553–557, 2005.

[15] C. L. B. Jordan, S. Bhattacharjee, F. Bossen, F. Jordan, and T. Ebrahimi, "Shape representation and coding of visual objets in multimedia applicationsan overview," in *Annales des télécommunications*, vol. 53, no. 5-6. Springer, 1998, pp. 164–178.

[16] H. Freeman, "On the encoding of arbitrary geometric configurations," *IRE Transactions on Electronic Computers*, no. 2, pp. 260–268, 1961.

[17] D. A. Huffman, "A method for the construction of minimum-redundancy codes," *Proceedings of the IRE*, vol. 40, no. 9, pp. 1098–1101, 1952.

[18] I. H. Witten, R. M. Neal, and J. G. Cleary, "Arithmetic coding for data compression," *Communications of the ACM*, vol. 30, no. 6, pp. 520–540, 1987.

[19] R. Song, H. Ko, and C. Kuo, "Mcl-3d: A database for stereoscopic image quality assessment using 2d-image-plus-depth source," *arXiv preprint arXiv:1405.1403*, 2014.

[20] C. Loop, Q. Cai, S. O. Escolano, and P. Chou, "Microsoft voxelized upper bodies-a voxelized point cloud dataset," *ISO/IEC JTC1/SC29 Joint WG 11/WG1 (MPEG/JPEG) input document m38673/M72012*, 2016.

[21] A. K. Katsaggelos, L. P. Kondi, F. W. Meier, J. Ostermann, and G. M. Schuster, "Mpeg-4 and rate-distortion-based shape-coding techniques," *Proceedings of the IEEE*, vol. 86, no. 6, pp. 1126–1154, 1998.

[22] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin, "The ball-pivoting algorithm for surface reconstruction," *IEEE Transactions on Visualization and Computer Graphics*, vol. 5, no. 4, pp. 349–359, 1999.

[23] A. Shamir, "Segmentation and shape extraction of 3d boundary meshes." in *Eurographics (STARs)*, 2006, pp. 137–149.

[24] C. Gotsman, "On graph partitioning, spectral analysis, and digital mesh processing," in *Shape Modeling International, 2003*, Seoul, South Korea, May 2003.