Space Filling Curves for 3D Sensor Networks with Complex Topology

Mayank Goswami* Siming Li† Junwei Zhang† Emil Saucan‡ David Xianfeng Gu† Jie Gao†

Abstract

Several aspects of managing a sensor network (e.g., motion planning for data mules, serial data fusion and inference) benefit once the network is linearized to a path. The linearization is often achieved by constructing a space filling curve in the domain. However, existing methods cannot handle networks distributed on surfaces of complex topology.

This paper presents a novel method for generating space filling curves for 3D sensor networks that are distributed densely on some two-dimensional geometric surface. Our algorithm is completely distributed and constructs a path which gets uniformly, progressively denser as it becomes longer. We analyze the algorithm mathematically and prove that the curve we obtain is dense. Our method is based on the Hodge decomposition theorem and uses holomorphic differentials on Riemann surfaces. The underlying high genus surface is conformally mapped to a union of flat tori and then a proportionally-dense space filling curve on this union is constructed. The pullback of this curve to the original network gives us the desired curve.

1 Introduction

In this paper we consider sensors deployed in 3D space such that the sensors are located densely on some underlying 2-dimensional geometric surface of possibly complex topology. This assumption models many practical scenarios in sensor deployment — sensors are often attached to the surfaces of terrains, exterior/interior of buildings [13], or other architectural structures, for easy installation and energy supplies, etc. In some other cases, the applications require sensors to be installed to monitor complex 3D structures, such as underground tunnels [14, 29] or pipes [22]. Therefore the sensors are located sparsely in 3D space but densely on a 2-dimensional surface (the "boundary" of some 3D objects) of possibly complex topology.

We are interested in innovative ways of managing such sensor networks, in the regime of using mobile entities to aid such management. Such mobile agents are often termed 'data mules', since one of the major applications is to use a mobile node to collect data from static sensors [1, 10, 17, 23, 27]. Static data sinks suffer from the well known problem of 'energy hole', as sensors near the sink are used more often and may run out of battery sooner than others. Mobile data sinks could get around this problem.

Our focus is to plan data mule along a path that uniformly traverses the entire field for a sensor network. This represents a periodic solution by which all sensors have a fair chance of being served by the data mule. It is better for scenarios when sensors have the same data generation rate, or when the sensor network requires a patrolling team to continuously monitor its general functioning and health. It can also be used for linear, logical operations in sensor networks such as data fusion [20] or sequential inference. Thus, the main questions is: given a unit-disk-graph G on n nodes (sensors) densely placed on a smooth two-dimensional manifold, construct a path γ that 1) passes through each node, and 2) for any integer $\ell = \Omega(\text{diameter}(G))$ and any subset A of nodes, the proportion of nodes among the first ℓ nodes in γ that lie in A is equal to the relative size of A.

The above problem is hard in general; however, if the sensors are densely distributed, one can attack a continuous version of it. The continuous version basically asks to construct a space filling curve on a two dimensional manifold M, which gets progressively dense (for any $\ell = \Omega(\text{diameter}(M))$ and any submanifold $A \subset M$, the proportion of γ_ℓ (the first ℓ length of γ) that lies inside A is equal to the relative area of A). The hope is that by "thickening" a solution to the continuous version, we might get a reasonable solution to the discrete version.

The representative work in this direction is by Ban *et al.* [4], which generalizes the idea of a space filling curve, often defined for a square, for a general 2D domain with holes. A space filling curve is a single curve that recursively 'fills up' the square, when the number of iterations goes to infinity [21]. For a sensor network with fixed density, a space filling curve nicely tours around all sensors with total travel length comparable to the traveling salesman solution. When the sensor network has holes, however, the space filling curve is broken and loses its nice properties. For a domain with a single hole, Ban *et al.* [4] proposed to map it to a torus such that a space filling curve can be easily found — by essentially following a line bouncing back and forth between the inner and outer boundaries. When there are two or more holes, all but one holes are mapped to 'slits', and the path bounces on these slits too. It is proved that this curve is dense, i.e., any point of the domain will be covered by the path of sufficiently long; and the curve has bounded density, i.e., it does not path through any point too many times.

However, one major limitation of the mechanism in Ban *et al.* [4] is its applicability to 2D domains with holes

*Max Planck Institute for Informatics, Saarbrücken, Germany gmayank@mpi-inf.mpg.de

†Computer Science Department, Stony Brook University, NY, USA {silli,junwezhang,gu,jgao}@cs.stonybrook.edu

‡Technion, Israel Institute of Technology; Max Planck Institute for Mathematics, Leipzig, Germany semil11@gmail.com

only. Terrains with holes can be handled via an additional mapping to 2D but general surfaces with high genus (multiple handles) cannot be handled. For underground deployment of sensors for monitoring tunnels, handles appear very often and we need a different scheme for generating the space filling curves.

Our contribution. The main result in this paper is a new linearization scheme for general sensor networks on 2D surfaces. We generate space filling curves with the same nice properties as those in [4]. In particular, the curves have 1) dense, progressive coverage – that as the curve gets longer, the distance from any point to the curve decreases quickly; 2) efficient coverage – a point is not visited more than a constant number of times.

Moving from a 2D domain to a general 2D surface in 3D really makes the problem harder. Note that before this work, there was no algorithm to construct the desired space-filling curve on a surface, even in the continuous setting. We remark that the problem is much easier if one drops the progressive density requirement. For that one can cut the 2D surface into small patches each mapped to a 2D domain such that previous methods can be applied.

To summarize, our contribution is not only on the algorithmic and application aspects, but also presents a theoretically proven, continuous-setting algorithm, that may be of independent interest. The algorithms presented here can be easily made to work in a distributed setting for a sensor network.

2 Related Work

In this section we survey other ideas that generate a path to visit all sensor nodes.

Space filling curves. Space filling curves have been used for linear/serial fusion [20] in sensor networks when the sensors are deployed uniformly in a square. There has been a heuristic algorithm that generalizes a Hilbert curve for an ellipse [11]. Technically the curves generated by Ban *et al.* [4] and the one in this paper are not going to completely fill up the surface (since topologically a curve is different from a surface) – but both the curves get infinitesimally closer to any given point.

Finding a tour. On a sensor network deployed in space, generating a tour of the graph, depending on the requirement, maps to either the Hamiltonian cycle/path problem or the traveling salesman tour. The former requires each vertex be visited exactly once and only the edges of the graph can be used. The second tries to minimizes the total travel distance instead. Both problems are NP-hard [9]. Euclidean TSP has good approximation schemes [3, 19] but these solutions suffer from two potential problems: 1) lack of progressive density; 2) cannot support multiple data mules easily.

Random walk. A practically appealing solution for visiting nodes in a network is by random walk. The downside is that we encounter the coupon collector problem. Initially a random walk visits a new node with high probability. After a random walk has visited a large fraction of nodes, it is highly likely that the next random node encountered has been visited before. Thus it takes a long time to aimlessly walk in the network and hope to find the last few unvisited nodes. Theoretically for a random walk to cover a grid-like network, the number of steps is quadratic in the size of the network [15]. For a random walk of linear number of steps, there are a lot of duplicate visits as well as a large number of nodes that are not visited at all. In the case of multiple random walks, since there is little coordination between the random walks, they may visit the same nodes and duplicate their efforts.

3 Theory of constructing space filling curves

For ease of exposition, we start by summarizing our method in this section. We then describe the theory behind our constructed curve, and end this section with the proof that the curve is dense. For the sake of completeness, we have provided all the theoretical material necessary for understanding our construction.

3.1 Informal discussion of techniques

Let us consider the mathematical problem of constructing a dense curve with the desired property of *proportional density* on a two dimensional manifold S. We first treat the surface as a one dimensional complex manifold, also called a Riemann surface. This basically means that locally our surface looks like an open set in the complex plane, and the transition maps from one such local "chart" to another are holomorphic.

With this point of view, we consider a *holomorphic differential* on our Riemann surface S. A holomorphic differential is basically an assignment of a complex-valued holomorphic function on each chart of the surface, that transforms line elements in the correct way; in complex coordinates z and $\bar{z}$, it is a tensor of type $(1, 0)$.

Using properties of certain special kinds of holomorphic differentials called Strebel differentials, we partition our surface into pieces, each of which is a flat torus with some holes removed. Each such piece is mapped to a parallelogram with slits (the boundaries of the holes map to the slits). In other words, we view the surface S as a union of parallelograms with slits, with slits being glued together in a certain way. This change of coordinates is mathematically termed a "branched covering".

In these coordinates, our curve is just a straight line on the cover. The slope of this line is either irrational, or chosen randomly, depending on the position of the slits and the sides of the parallelograms. Using several important and recent results in *Teichmüller theory*, we can prove density.

Note that although we partition the surface into pieces, we do not cover one piece first and then move on to the next. Instead our curve comes back into each piece infinitely often, increasing the density proportionally to the length.

3.2 Theoretic Background

Conformal Atlas Suppose $(S, \mathbf{g})$ is a surface with a Riemannian metric $\mathbf{g}$. Given any point $p \in S$, there is a neighborhood $U(p)$ where one can find the *isothermal coordinates* (i.e., local coordinates where the metric is conformal to the Euclidean metric) (x, y) on $U(p)$, such that

$$\mathbf{g} = e^{2\lambda(x,y)}(dx^2 + dy^2),$$

where the scalar function $\lambda : U(p) \to \mathbb{R}$ is the conformal factor function. The atlas consisting of isothermal coordinates is called a *conformal atlas*. In the following discussion, we always assume the local parameters are isothermal.

De Rham Cohomology De Rham cohomology theory is based on the existence of differential forms with certain prescribed properties. Suppose $f : S \to \mathbb{R}$; then its differential is given by

$$df(x, y) = \frac{\partial f(x, y)}{\partial x}dx + \frac{\partial f(x, y)}{\partial y}dy,$$

Suppose ω is a differential 1-form on the surface, which has local representation as $\omega(x, y) = f(x, y)dx+g(x, y)dy$. The *exterior differential operator* d acts on ω as,

$$d\omega(x, y) = (\frac{\partial g}{\partial x} - \frac{\partial f}{\partial y})dx \wedge dy.$$

If $d\omega = 0$, then ω is called a *closed 1-form*. If there exists a function $h : S \to \mathbb{R}$ such that $\omega = dh$, then ω is called an *exact 1-form*. Exact 1-forms are closed. The first De Rham cohomology group of the surface is the group of all non-exact closed 1-forms.

Hodge Decomposition The Hodge star operator on differential forms is defined as

$$^*\omega = {}^*(f(x, y)dx+g(x, y)dy) = (-g(x, y)dx+f(x, y)dy).$$

A differential 1-form is called a *harmonic*, if $d\omega = 0, d^*\omega = 0$. *The Hodge decomposition theorem* states that each cohomology class has a unique harmonic form. The group consisting of all the harmonic 1-forms is denoted as $H^1_\Delta(S, \mathbb{R})$; it is isomorphic to $H^1(S, \mathbb{R})$.

Holomorphic Differentials Let $\{(U_\alpha, z_\alpha\}$ be the conformal atla ygs, where the complex parameter $z_\alpha = x_\alpha + \sqrt{-1}y_\alpha$. Suppose (U_β, z_β) is another chart, the parameter transition function is $z_\beta(z_\alpha)$ is *holomorphic*, namely, it satisfies the following Cauchy-Riemann equations:

$$\begin{cases} \frac{\partial x_\beta}{\partial x_\alpha} &= \frac{\partial y_\beta}{\partial y_\alpha} \\ \frac{\partial x_\beta}{\partial y_\alpha} &= -\frac{\partial y_\beta}{\partial x_\alpha} \end{cases}$$

Let Ω be a complex differential form with local representation $\Omega(z_\alpha) = f(z_\alpha)dz_\alpha$, where $f(z_\alpha)$ is holomorphic. A holomorphic 1-form can be decomposed to a pair of conjugate harmonic real differential 1-forms, $\Omega = \omega + \sqrt{-1}^*\omega$, where ω is harmonic. All the holomorphic differentials form a group $\Omega(S)$, which is isomorphic to $H^1_\Delta(S, \mathbb{R})$.

Branched covering Let X, Y be compact connected topological spaces. A continuous mapping $f : X \to Y$ is called a branched covering if it is a local homeomorphism everywhere except a finite number of "branch" points. In the complex setting, this would mean that a branched covering is, locally at a point p, upto composition by biholomorphic maps, of the form $z \to z^{e_p}$, where $e_p > 1$ for finitely many branch points, and $e_p = 1$ everywhere else.

Trajectory Structure and Strebel differentials Given a holomorphic 1-form Ω on a genus g surface, there are $2g - 2$ zero points. At each point $p \in S$, the tangent direction $d\gamma \in TM_p$ is called a *horizontal direction*, if $\Omega(d\gamma)$ is real. A curve $\gamma \subset S$ is called a *horizontal trajectory* of Ω, if at each point $p \in \gamma$, $d\gamma$ is along the horizontal direction. The horizontal trajectories through zeros of Ω are called *critical trajectories*. Similar to holomorphic 1-forms, one can consider quadratic differentials, which are tensors of type $(2, 0)$ in holomorphic coordinates. For quadratic differentials we define a direction to be horizontal if the differential is positive along it, and vertical if it is negative.

If the graph of vertical critical trajectories is compact, the quadratic differential is called Strebel. In the group of quadratic differentials, Strebel differentials are dense [5]. We will use a holomorphic-1 form whose square is Strebel. This will imply that the horizontal trajectories are closed curves.

3.3 Dense curve construction in continuous setting

We describe first the branched covering we use to construct our curve, and then prove the density.

Branched covering from a Strebel differential Given a Strebel differential Ω, the critical horizontal trajectories segment the surface to g connected components, denoted as $\{\Gamma_1, \Gamma_2, \cdots, \Gamma_g\}$. Each connected component Γ_k is of genus one with boundaries,

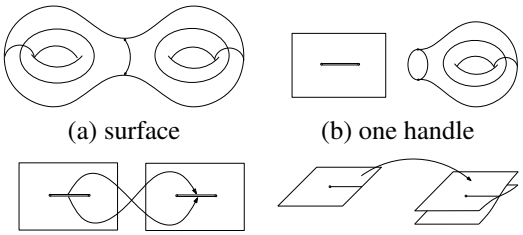
$$\partial\Gamma_k = b^1_k + b^2_k + \cdots + b^{n_k}_k.$$

The Strebel differential Ω induces a flat metric on each Γ_k, the integration of Ω on each boundary loop b^i_k maps the boundary loop to a straight line slit. Namely, then integration of Ω on each Γ_k maps Γ_k to a flat torus with straight line slits.

The mapping from the surface to the flat tori are diffeomorphic except at the zero points. Locally, the mapping at the zero points is similar to the complex power map $z \mapsto z^2$. Therefore, the zero points are the branch points.

The curve we use Suppose each flat torus is $\mathbb{R}^2/\Gamma_k$, $k = 1, 2, \cdots, g$. Here Γ_k represent lattice groups. Then we can find a line ℓ on the plane, such that ℓ does not go through any points in the union of lattices $\cup_k\Gamma_k$, since this union is countable. In particular if the lattice points are all rational then a line with irrational slope would do; otherwise we chose a random line. Denote the slope of ℓ as k.

On the"welded flat tori", start from one point draw a line γ with slope k. Then γ goes across the handles via the slits; when it hits a slit it moves from one handle to another and



(a) surface (b) one handle

(c) glue pattern (d) branch point

Figure 1: Branch covering map.

continues with the same slope k. We take care to chose k in such a way that this line does not pass through the endpoint of any slit. Again, this is easy to maintain since we only have finitely many of these endpoints.

Theorem 1. *Let γ be the curve constructed as above. Then γ is dense and does not go through any point more than once.*

Proof. Density would imply aperiodicity of γ, which in turn would imply that γ does not pass through any point twice. This is because on each torus, γ is a line with slope k, and if it visited a point twice it must necessarily become periodic. Thus it suffices to prove that γ is dense.

Density of such a curve follows from results in Teichmüller theory [18], and we just sketch the proof. Essentially, as long as the direction k does not contain a "saddle connection", which is a trajectory connecting two zeroes of the holomorphic differential, it will be dense. In our case, if the slit coordinates are rational, we choose an irrational k; otherwise we choose a random k. In both cases, with probability 1 we will neither hit the lattice points Γ_k nor the end points of the slits. This guarantees density. □

Note that although proving progressive density requires more sophisticated math techniques, our simulations in [32] show that the curve we construct does indeed satisfy this property.

4 Algorithm

In this section we present a centralized algorithm for the input of a sensor network densely deployed on a surface. Note that this can be generalized to a completely distributed algorithm too; we choose not to present the distributed algorithm for the sake of simplicity in this extended abstract, and refer to [32] for details.

We assume that the sensors are densely deployed on some underlying surface S such that locally the sensors lie on a flat plane. Thus we can apply existing algorithms to first come up with a triangulation of the sensors that approximate the underlying surface S [7, 8].

The surface is approximated by a triangular mesh $M = (V, E, F)$, where V, E, F denotes the vertex, edge and face sets respectively. We use $v_i \in V$ to represent a vertex, $[v_i, v_j]$ an oriented edge from v_i to v_j, $[v_i, v_j, v_k]$ an oriented face where v_i, v_j and v_k are sorted counter-clock-wisely. We assume the mesh is closed with genus g.

All the mathematics objects we described in Section 3 (homology groups, cohomology groups, harmonic and holomorphic 1-forms, branched covering maps) have straightforward discrete analogs. We omit the definitions (they can be found in [32]). In the following, we are talking about the discrete versions of such objects.

The algorithm pipeline is as follows:

1. Compute the basis of the first homology group $H_1(M, \mathbb{Z})$.
2. Calculate the dual basis of the first cohomology group $H^1(M, \mathbb{R})$.
3. Obtain the basis of the harmonic 1-form group
4. Achieve the basis of the holomorphic 1-form group
5. Integrate a holomorphic 1-form to get the required branched covering map.
6. Use the curve described in the previous section.

Steps $1-3$ have been carried out in literature (even in the distributed setting) before, and we give their details in the appendix. Here we elaborate on the final and novel Steps 4 and 5.

Branch Covering Map We first compute the cut graph[1] G of the mesh. Then we slice the mesh along the cut group to obtain a *fundamental domain* M/G. Choose one holomorphic 1-form $\omega + \sqrt{-1}^*\omega$ and integrate the holomorphic 1-form on the fundamental domain to get a branched covering map. Fix a vertex $v_0 \in M/G$ as the base vertex, for any vertex $v_i \in M/G$,

$$\varphi(v_i) = \int_{v_0}^{v_i} \omega + \sqrt{-1}^*\omega,$$

the integration path $\gamma \subset M/G$ can be chosen arbitrarily, which consists a sequence of consecutive oriented edges, connecting v_0 to v_i, denoted as

$$\gamma = e_0 + e_1 + \cdots e_k,$$

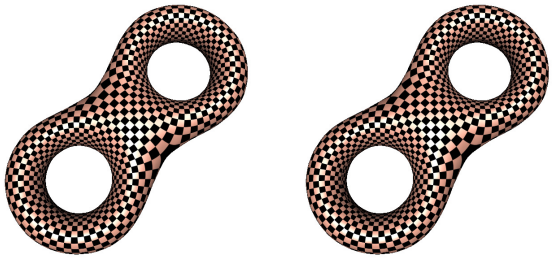
such that target vertex of e_i equals to the source vertex of e_{i+1}, the source of e_0 is v_0, the target of e_k is v_i.

$$\int_\gamma \omega = \sum_{i=0}^{k} \omega(e_i).$$

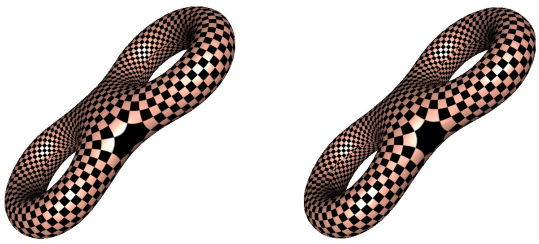
The branching points of φ are the zero points of the holomorphic 1-form. The slits are the horizontal trajectories connecting the zeros of the holomorphic 1-form.

As shown in Fig.1, there are $2g-2$ zero points of the holomorphic 1-form. The horizontal trajectories through the zeros segment the surface into handles as shown in Frame (a).

[1] Intuitively, this is the set of edges whose removal transforms the surface into a disk.

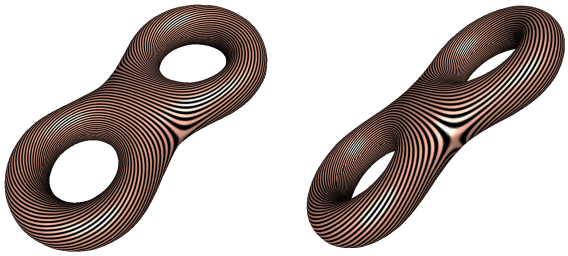


(a) front view (b) back view



(c) first zero point (d) second zero point

Figure 2: Holomorphic 1-form and zero points.



(a) dense curve (b) zero point

Figure 3: Dense curve on the surface.

Each handle is conformally mapped onto a flat torus with a slit, the end points of the slit are the zero points, as shown in Frame (b). The flat tori are glued together through slits, the top (bottom) edge of the slit on one torus is glued to the bottom (top) edge of the slit on the other torus, as shown in Frame (c). In the neighborhood of each zero point, the mapping is a branch covering similar to $z \mapsto z^2$, as illustrated in Frame (d).

The curve On the glued tori, start from any point and draw a line γ with slope k. Then γ goes across the handles via the slits; when it hits a slit it moves from one handle to another and continues with the same slope k. We take care to chose k in such a way that this line does not pass through the endpoint of any slit. This is easy to maintain since we only have finitely many of these endpoints.

5 Conclusion

We show in this paper a new construction for computing a dense curve on a 3D sensor network when the sensors are densely on a 2D manifold. The algorithm substantially generalizes over the prior work by Ban *et al.* [4] while keeping essentially the same nice properties. As future work we would like to see how to generalize the idea to truly 3D networks (volumetric 3D sensor networks).

Acknowledgment

Jie Gao would like to acknowledge NSF support through DMS-1418255, DMS-1221339, CNS-1217823 and Air Force Research AFOSR FA9550-14-1-0193.

References

[1] G. Anastasi, M. Conti, and M. Di Francesco. Data collection in sensor networks with data mules: An integrated simulation analysis. In *Computers and Communications, 2008. ISCC 2008. IEEE Symposium on*, pages 1096–1102, July 2008.

[2] D. L. Applegate, R. E. Bixby, V. Chvatal, and W. J. Cook. *The Traveling Salesman Problem: A Computational Study (Princeton Series in Applied Mathematics)*. Princeton University Press, 2007.

[3] S. Arora. Polynomial time approximation schemes for euclidean traveling salesman and other geometric problems. *J. ACM*, 45:753–782, September 1998.

[4] X. Ban, M. Goswami, W. Zeng, X. D. Gu, and J. Gao. Topology dependent space filling curves for sensor networks and applications. In *Proc. of 32nd Annual IEEE Conference on Computer Communications (INFOCOM'13)*, April 2013.

[5] A. Douady and J. Hubbard. On the density of strebel differentials. *Inventiones mathematicae*, 30(2):175–179, 1975.

[6] E. Ekici, Y. Gu, and D. Bozdag. Mobility-based communication in wireless sensor networks. *Communications Magazine, IEEE*, 44(7):56–62, July 2006.

[7] S. Funke and N. Milosavljević. Network sketching or: "how much geometry hides in connectivity? - part II". In *SODA '07: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 958–967, 2007.

[8] J. Gao, L. J. Guibas, J. Hershberger, L. Zhang, and A. Zhu. Geometric spanner for routing in mobile networks. In *Proceedings of the 2nd ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'01)*, pages 45–55, 2001.

[9] M. R. Garey and D. S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1990.

[10] D. Jea, A. Somasundara, and M. Srivastava. Multiple controlled mobile elements (data mules) for data collection in sensor networks. In *Proceedings of the First IEEE International Conference on Distributed Computing in Sensor Systems*, DCOSS'05, pages 244–257, 2005.

[11] M. Kamat, A. Ismail, and S. Olariu. Modified hilbert space-filling curve for ellipsoidal coverage in wireless ad hoc sensor networks. In *Signal Processing and Communications, 2007. ICSPC 2007. IEEE International Conference on*, pages 1407 –1410, nov. 2007.

[12] A. Kansal, A. A. Somasundara, D. D. Jea, M. B. Srivastava, and D. Estrin. Intelligent fluid infrastructure for embedded networks. In *Proceedings of the 2Nd International Conference on Mobile Systems, Applications, and Services*, MobiSys '04, pages 111–124, 2004.

[13] M. Klann. Mobile response. chapter Tactical Navigation Support for Firefighters: The LifeNet Ad-Hoc Sensor-Network and Wearable System, pages 41–56. Springer-Verlag, Berlin, Heidelberg, 2009.

[14] M. Li and Y. Liu. Underground coal mine monitoring with wireless sensor networks. *ACM Trans. Sen. Netw.*, 5:10:1–10:29, April 2009.

[15] L. Lovasz. Random walks on graphs: A survey. *Bolyai Soc. Math. Stud.*, 2, 1996.

[16] M. Ma and Y. Yang. Sencar: An energy-efficient data gathering mechanism for large-scale multihop sensor networks. *Parallel and Distributed Systems, IEEE Transactions on*, 18(10):1476–1488, Oct 2007.

[17] D. D. L. Mascarenas, E. Flynn, K. Lin, K. Farinholt, G. Park, R. Gupta, M. Todd, and C. Farrar. Demonstration of a roving-host wireless sensor network for rapid assessment monitoring of structural health.

[18] T. S. Masur H. Rational billiards and flat structures. *Hasselblatt B., Katok, A. (eds.) Handbook of Dynamical Systems*, 1A:1015–1089, 2002.

[19] J. S. B. Mitchell. Guillotine subdivisions approximate polygonal subdivisions: A simple polynomial-time approximation scheme for geometric tsp, k-mst, and related problems. *SIAM J. Comput.*, 28:1298–1309, March 1999.

[20] S. Patil and S. R. Das. Serial data fusion using space-filling curves in wireless sensor networks. In *Proceedings of IEEE International Conference on Sensor and Ad Hoc Communications and Networks (SECON'04)*, pages 182–190, 2004.

[21] H. Sagan. *Space-Filling Curves*. Springer-Verlag, New York, 1994.

[22] R. Sokullu, M. A. Akkas, and F. Demirel. Data gathering system for watering and gas pipelines using wireless sensor networks. In *ICNS 2011, The Seventh International Conference on Networking and Services*, pages 190–195, May 2011.

[23] A. Somasundara, A. Kansal, D. Jea, D. Estrin, and M. Srivastava. Controllably mobile infrastructure for low energy embedded networks. *Mobile Computing, IEEE Transactions on*, 5(8):958–973, Aug 2006.

[24] A. Somasundara, A. Ramamoorthy, and M. Srivastava. Mobile element scheduling for efficient data collection in wireless sensor networks with dynamic deadlines. In *Real-Time Systems Symposium, 2004. Proceedings. 25th IEEE International*, pages 296–305, Dec 2004.

[25] A. A. Somasundara, A. Ramamoorthy, and M. B. Srivastava. Mobile element scheduling with dynamic deadlines. *IEEE Transactions on Mobile Computing*, 6(4):395–410, 2007.

[26] P. Toth and D. Vigo, editors. *The Vehicle Routing Problem*. 2002.

[27] I. Vasilescu, K. Kotay, D. Rus, M. Dunbabin, and P. Corke. Data collection, storage, and retrieval with an underwater sensor network. In *Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems*, SenSys '05, pages 154–165, 2005.

[28] Y. Wang, J. Gao, and J. S. B. Mitchell. Boundary recognition in sensor networks by topological methods. In *Proc. of the ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, pages 122–133, September 2006.

[29] D. Wu, L. Bao, and R. Li. A holistic approach to wireless sensor network routing in underground tunnel environments. *Comput. Commun.*, 33:1566–1573, August 2010.

[30] G. Xing, T. Wang, Z. Xie, and W. Jia. Rendezvous planning in mobility-assisted wireless sensor networks. In *Real-Time Systems Symposium, 2007. RTSS 2007. 28th IEEE International*, pages 311–320, Dec 2007.

[31] W. Zhao and M. Ammar. Message ferrying: proactive routing in highly-partitioned wireless ad hoc networks. In *Distributed Computing Systems, 2003. FTDCS 2003. Proceedings. The Ninth IEEE Workshop on Future Trends of*, pages 308–314, May 2003.

[32] M. Goswami, S. Li, J. Zhang, E. Saucan, X. D. Gu, and J. Gao Space Filling Curves for 3D Sensor Networks with Complex Topology. In*Technical Report*, http://www3.cs.stonybrook.edu/ jgao/paper/densecurve3d.pdf.

Appendix–The Algorithm Details

As mentioned, we choose not to present the distributed algorithm for the sake of simplicity in this extended abstract, and refer the reader to [32] for details.

We assume that the sensors are densely deployed on some underlying surface S such that locally the sensors lie on a flat plane. Thus we can apply existing algorithms to first come up with a triangulation of the sensors that approximate the underlying surface S [7, 8].

The surface is approximated by a triangular mesh $M = (V, E, F)$, where V, E, F denotes the vertex, edge and face sets respectively. We use $v_i \in V$ to represent a vertex, $[v_i, v_j]$ an oriented edge from v_i to v_j, $[v_i, v_j, v_k]$ an oriented face where v_i, v_j and v_k are sorted counter-clock-wisely. We assume the mesh is closed with genus g.

The algorithm pipeline is as follows:

1. Compute the basis of the first homology group $H_1(M, \mathbb{Z})$.
2. Calculate the dual basis of the first cohomology group $H^1(M, \mathbb{R})$.
3. Obtain the basis of the harmonic 1-form group.
4. Achieve the basis of the holomorphic 1-form group.
5. Integrate a holomorphic 1-form to get the required branched covering map.
6. Use the curve described in the Section 3.

Homology Group First, the dual mesh $\bar{M} = (\bar{V}, \bar{E}, \bar{F})$ of the input mesh M is constructed. Each vertex $v_i \in V$, face $f_j \in F$ and edge $e_k \in E$ corresponds to a face $\bar{v}_i \in \bar{F}$, a vertex $\bar{f}_j \in \bar{V}$ and and edge $\bar{e}_k \in \bar{E}$ on the dual mesh respectively.

Second, a spanning tree $\bar{T}$ of the dual mesh $\bar{M}$ is computed. The *cut graph* $G \subset M$ is the union of edges, whose dual edges are not in the spanning tree:

$$G = \{e \in E | \bar{e} \notin \bar{T}\}.$$

Intuitively, the mesh $M \setminus G$ with the cut graph removed is a topological disk. Third, a spanning tree T of the cut graph G is calculated. The complement of T in G is a union of edges:

$$G/T = \{e_1, e_2, \cdots, e_{2g}\},$$

Each edge e_i when included in the spanning tree T (thus $T \cup e_i$) gives rise to a unique loop $\gamma_i \subset T \cup e_i$. These loops

$$\{\gamma_1, \gamma_2, \cdots, \gamma_{2g}\}$$

form the basis of the first homology group $H_1(M, \mathbb{Z})$.

Cohomology Group The differential forms are approximated by discrete forms. A discrete 0-form is a function defined on vertices, $f : V \to \mathbb{R}$; a discrete 1-form is a function defined on the oriented edges, $\omega : E \to \mathbb{R}$, $\omega([v_i, v_j]) = -\omega([v_j, v_i])$; a discrete 2-form is defined on the oriented faces $\tau : F \to \mathbb{R}$. Discrete exterior differential operator d is dual to the boundary operator ∂, for example

$$\begin{aligned} d\omega([v_i, v_j, v_k]) &= \omega(\partial[v_i, v_j, v_k]) \\ &= \omega([v_i, v_j]) + \omega([v_j, v_k]) + \omega([v_k, v_i]). \end{aligned}$$

Given the first homology group $H_1(M, \mathbb{Z})$ basis $\{\gamma_1, \cdots, \gamma_{2g}\}$, the dual cohomology group basis can be obtained as follows. For each base loop γ_k, slice the mesh M to get an open mesh M_k. The boundary of M_k has two connected components, denoted them as

$$\partial M_k = \gamma_k^+ \cup \gamma_k^-.$$

Construct a function $f_k : M_k \to \mathbb{R}$,

$$f_k(v_i) = \begin{cases} 1 & v_i \in \gamma_k^+ \\ 0 & v_i \in \gamma_k^- \\ \text{rand} & v_i \notin \partial M_k \end{cases}$$

Then the 1-form df_k, $df_k([v_i, v_j]) = f_k(v_j) - f_k(v_i)$ is 0 on all boundary edges. Therefore, one can define ω_k on the original closed mesh M. Suppose e is not on the loop γ_k, then it has a unique corresponding edge $\tilde{e}$ on M_k, define $\omega_k(e) = df_k(\tilde{e})$. If $e \subset \gamma_k$, then let $\omega_k(e) = 0$. ω_k is a closed 1-form, and not exact. These non-exact closed 1-forms

$$\{\omega_1, \omega_2, \cdots, \omega_{2g}\}$$

form the basis of $H^1(M, \mathbb{R})$.

Harmonic Differential Group According to Hodge theory, each cohomological class has a unique harmonic 1-form. Given a cohomology group basis $\{\omega_1, \omega_2, \cdots, \omega_{2g}\}$, for each closed 1-form ω_k, there is a function $h_k : V \to \mathbb{R}$, such that $\omega_k + dh_k$ is harmonic. By definition, the 1-form is curl free

$$d(\omega_k + dh_k) = d\omega_k + d^2 h_k = 0.$$

The divergence is

$$^*d^*(\omega_k + dh_k) = 0,$$

this induces the linear system, for each vertex $v_i \in V$,

$$\sum_{[v_i, v_j] \in E} w_{ij} \left(h_k(v_j) - h_k(v_i) + \omega_k([v_i, v_j])\right) = 0,$$

where w_{ij} is the cotangent edge weight. Suppose edge $[v_i, v_j]$ is shared by two faces $[v_i, v_j, v_k]$ and $[v_j, v_i, v_l]$, then

$$w_{ij} = \cot \theta_k^{ij} + \cot \theta_l^{ji},$$

where θ_k^{ij} is the corner angle at vertex v_k in triangle $[v_i, v_j, v_k]$. The coefficient matrix of the linear system is

positive definite, the solution exists and is unique. The 1-forms

$$\{\omega_1 + dh_1, \omega_2 + dh_2, \cdots, \omega_{2g} + dh_{2g}\}$$

form the basis of the harmonic 1-form group.

Holomorphic Differential Group Suppose the harmonic 1-form group basis is given, still denoted as $\{\omega_1, \omega_2, \cdots, \omega_{2g}\}$. Let ω be a harmonic 1-form, its conjugate 1-form ${}^*\omega$ is harmonic as well, therefore it can be represented as linear combination of $\{\omega_k\}$,

$${}^*\omega = \lambda_1\omega_1 + \lambda_2\omega_2 + \cdots + \lambda_{2g}\omega_{2g}. \tag{1}$$

The coefficients can be obtained by solving the following the linear system

$$\int_M {}^*\omega \wedge \omega_k = \sum_{i=1}^{2g} \lambda_i \int_M \omega_i \wedge \omega_k, k = 1, 2, \cdots, 2g.$$

On one triangle $[v_i, v_j, v_k]$ embed on the plane $\mathbb{R}^2$, the closed 1-form ω_k can be represented as a constant 1-form $\omega_k = a_k dx + b_k dy$, such that

$$\omega_k([v_i, v_j]) = \int_{[v_i, v_j]} a_k dx + b_k dy,$$

same equations hold for other edges $[v_j, v_k]$ and $[v_k, v_i]$. The wedge product on the face is given by

$$\omega_i \wedge \omega_j = (a_i dx + b_i dy) \wedge (a_j dx + b_j dy) = \begin{vmatrix} a_i & b_i \\ a_j & b_j \end{vmatrix} dx \wedge dy.$$

Therefore

$$\int_{[v_i, v_j, v_k]} \omega_i \wedge \omega_j = (a_i b_j - a_j b_i)\ Area([v_i, v_j, v_k]).$$

and

$$\int_M \omega_i \wedge \omega_j = \sum_{[v_i, v_j, v_k]} \int_{[v_i, v_j, v_k]} \omega_i \wedge \omega_j.$$

Locally, the Hodge operator is given by

$${}^*\omega_k = {}^*(a_k dx + b_k dy) = a_k dy - b_k dx.$$

So the coefficients in the linear equation 1 can be easily computed. By solving the linear system, the conjugate harmonic 1-form ${}^*\omega$ is obtained.

The harmonic 1-form basis $\{\omega_k\}$, paired with its conjugate harmonic 1-form $\{{}^*\omega\}$ form the holomorphic 1-form basis

$$\{\omega_1 + \sqrt{-1}{}^*\omega_1, \omega_2 + \sqrt{-1}{}^*\omega_2, \cdots, \omega_{2g} + \sqrt{-1}{}^*\omega_{2g}\}$$

Branch Covering Map Compute the cut graph G of the mesh, slice the mesh along the cut group to obtain a *fundamental domain* M/G. Choose one holomorphic 1-form $\omega + \sqrt{-1}{}^*\omega$ and integrate the holomorphic 1-form on the fundamental domain to get the branch covering map. Fix a vertex $v_0 \in M/G$ as the base vertex, for any vertex $v_i \in M/G$,

$$\varphi(v_i) = \int_{v_0}^{v_i} \omega + \sqrt{-1}{}^*\omega,$$

the integration path $\gamma \subset M/G$ can be chosen arbitrarily, which consists a sequence of consecutive oriented edges, connecting v_0 to v_i, denoted as

$$\gamma = e_0 + e_1 + \cdots e_k,$$

such that target vertex of e_i equals to the source vertex of e_{i+1}, the source of e_0 is v_0, the target of e_k is v_i.

$$\int_\gamma \omega = \sum_{i=0}^{k} \omega(e_i).$$

The branching points of φ are the zero points of the holomorphic 1-form. The slits are the horizontal trajectories connecting the zeros of the holomorphic 1-form.

As shown in Fig.1, there are $2g-2$ zero points of the holomorphic 1-form. The horizontal trajectories through the zeros segment the surface into handles as shown in Frame (a). Each handle is conformally mapped onto a flat torus with a slit, the end points of the slit are the zero points, as shown in Frame (b). The flat tori are glued together through slits, the top (bottom) edge of the slit on one torus is glued to the bottom (top) edge of the slit on the other torus, as shown in Frame (c). In the neighborhood of each zero point, the mapping is a branch covering similar to $z \mapsto z^2$, as illustrated in Frame (d).