

Efficient Path Generation with Reduced Coordinates

Renjie Chen¹ Craig Gotsman² Kai Hormann^{3,4}

¹Max-Planck Institute for Informatics, Saarbrücken, Germany

²New Jersey Institute of Technology, Newark, NJ, USA

³Università della Svizzera italiana, Lugano, Switzerland

⁴Nanyang Technological University, Singapore

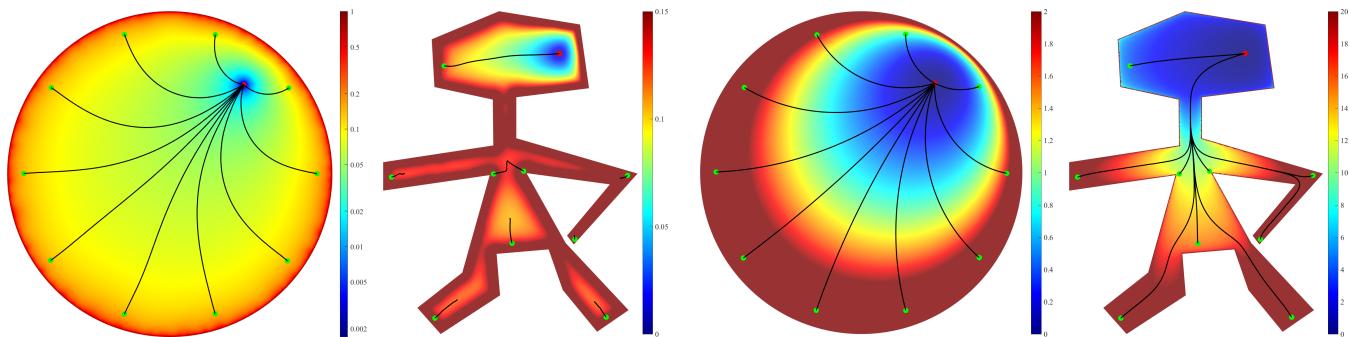


Figure 1: A smooth path from a source point z (green) to a target point y (red) inside a planar domain can be generated by following the negative gradient of a shape-aware distance function $d(y, z)$ that measures some distance between the Poisson kernels at y and z . While the L^2 -distance (left) works well for simple shapes, for complex domains it has local minima where the gradient-descent path terminates. In contrast, the f -divergence (right) is guaranteed to be void of local minima and can be implemented efficiently using reduced coordinates.

Abstract

Path generation is an important problem in many fields, especially robotics. One way to create a path between a source point z and a target point y inside a complex planar domain Ω is to define a non-negative distance function $d(y, z)$, such that following the negative gradient of d (by z) traces out such a path. This presents two challenges: (1) The mathematical challenge of defining d , such that $d(y, z)$ has a single minimum at $z = y$ for any fixed y , because the gradient-descent path may otherwise terminate at a local minimum before reaching y ; (2) The computational challenge of defining d , such that it can be computed efficiently. Using the concepts of harmonic measure and f -divergence, we show how to assign a set of reduced coordinates to each point in Ω and to define a family of distance functions based on these coordinates, such that both the mathematical and the computational challenge are met. Since in practice, especially in robotics applications, the path is often restricted to follow the edges of a discrete network defined on a finite set of sites sampled from Ω , any method that works well in the continuous setting must be discretized appropriately to preserve the important properties of the continuous case. We show how to define a network connecting a finite set of sites, such that a greedy routing algorithm, which is the discrete equivalent of continuous gradient descent, based on our reduced coordinates is guaranteed to generate a path in the network between any two sites. In many cases, this network is close to a planar graph, especially if the set of sites is dense. Guaranteeing the existence of a greedy route between any two points in the graph is a significant advantage in practical applications, avoiding the complexity of other path-planning methods, such as the shortest-path and A* algorithms. While the paths generated by our algorithm are not the shortest possible, in practice we found that they are close to that.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Geometric algorithms G.2.2 [Computer Graphics]: Graph Theory—Network problems

1. Introduction

Path generation in complex planar domains is an important and challenging problem in robotic navigation. The objective is for the

robot to move from one point (the source) in the domain to another (the target) along a short and smooth path that stays within the domain and avoids obstacles. Practical implementations discretize the domain as a dense graph and “route” along the edges of this graph,

ideally generating such paths. Common algorithms are the classical Dijkstra algorithm for shortest-path computation and the more sophisticated A* [HNR68] and its variants. The interested reader is referred to Souissi et al. [SBD^{*}13] for a recent survey of these methods. The simple shortest-path algorithm, while generating an optimal path, can be notoriously slow, and in principle may search the entire graph in order to find the path. Using heuristic functions to provide a lower-bound estimate of the remaining distance in the A* algorithm may speed this up by avoiding traversal of irrelevant portions of the graph, but, depending on the accuracy of the heuristic function, may still incur significant overhead. In terms of complexity, the best situation to hope for is the existence of a distance function, which can guide the robot from the source to the target by systematically minimizing this distance. Ideally, a path is generated by *gradient descent* on this function, where the robot moves from a node in the graph to the neighboring node which reduces the distance function the most, until the target is reached. This method, sometimes called *greedy routing*, is an extremely simple algorithm. It requires no priority queue and incurs no extra overhead of exploring other nodes of the graph, as opposed to A* and its variants. The small price to pay is that the paths are not the shortest possible. But shortest paths may not be desirable anyway at times, since they have sharp corners. However, greedy routing assumes quite optimistically that the distance function has no local minima, otherwise the procedure could get stuck there (namely, reach a vertex from which all neighbors increase the distance to the target). Indeed, if the shortest-path distance from the target to every other node in the graph was known in advance, following a gradient-descent path on this distance would immediately (by definition) trace out the shortest path. As this distance is obviously not known in advance, a second best is to use a different distance function, which is easy to compute, void of local minima and generates paths not too much longer than the shortest path. Defining and implementing such distance functions is the topic of this paper.

The idea of path-planning based on gradient descent is not new. A well-known family of path generation algorithms, inspired by the physics of electrical force fields, is based on *potential functions*. These were first proposed by Khatib [Kha86] and developed by a number of authors [KB91, KK92, RK92, CG93] soon after. As described above, the main idea is, given the target point, to construct a scalar function on the domain, such that a path to the target point from any other source point may be obtained by following the negative gradient of the function. For this to work as planned, the scalar function must have a global minimum at the target and be void of local minima elsewhere in the domain. Other critical points, such as saddles, are undesirable but not fatal, since a negative gradient can still be detected by probing around the saddle. Designing and computing potential functions for complex planar domains, possibly with holes (representing obstacles), has been a topic of intense activity for decades. Perhaps the most elegant type of potential function is the harmonic function [ABR01], which has very useful mathematical properties, most notably the guaranteed absence of local minima, in both the continuous and discrete settings. Alas, the main problem preventing widespread use of this potential function is the high complexity of computing the function, essentially the solution of a very large system of linear equations on a discretization of the domain every time the target point is changed, and

the fact that very high precision numerical methods are required, as the functions are almost constant, especially in regions distant from the target. A recent paper by Chen et al. [CGH17] addresses the first of these issues. They describe a new family of continuous distance functions, which, while quite distinct from the harmonic potential function, also possess no local minima and generate exactly the same gradient descent paths as the harmonic potential. The present paper expands on these functions, in particular for the discrete setting. Since the discrete version of these distance functions are not completely void of local minima, we show how to remedy that. More importantly, we show how to compute variants of these distance functions, based on the concept of *reduced coordinates*, extremely efficiently.

Chen et al. [CGH17] use the concept of harmonic measure [GM05] and its conformal invariance to define a family of shape-aware distance functions $d_f(y, z)$ on a bounded simply-connected planar domain Ω . These *f-divergence distances* (see Section 3) are defined as the *f-divergence* [Csi67, KL51] of the two Poisson kernels of Ω at y and z , where f is a strictly convex real function. Figure 1 shows examples of gradient-descent paths for two bounded simply-connected domains, using the *f*-divergence (right) and the L^2 -distance (left) between the Poisson kernels. While the latter can have local minima where the paths may terminate, the paths derived from the *f*-divergence always reach the target. Chen et al. [CGH17] prove that $d_f(y, z)$ is symmetric, subharmonic, has a global minimum at $z = y$ and no local minima, and that the paths are *invariant* to f . This is because only the *magnitude* of $\nabla_z d_f(y, z)$ depends on f , *but not its direction*. Furthermore, the *f*-divergence distance paths are identical to those generated by following the negative gradient of the harmonic potential function.

In practice, we usually consider a discretization of the domain Ω , that is, a triangulation T with k vertices, where k can be on the order of hundreds of thousands, and all computations are done on this triangulation. Given a source point $z \in T$ and a target point $y \in T$, path generation using the harmonic potential function requires solving a large $k \times k$ system of sparse linear equations that stem from the standard Finite Element Method (FEM) discretization of Laplace's equation. This is usually done by computing the Cholesky decomposition of the system matrix in a preprocessing step and then at query time, given a target vertex y , using back-substitution to compute the distances from *all* $x \in T$ to y at once in $\mathcal{O}(k \log k)$ time, because the Cholesky factor has $\mathcal{O}(k \log k)$ non-zero entries. In contrast, for *f*-divergence distance paths, the Poisson kernels are determined at *all* the vertices of T in a preprocessing step, resulting in k *coordinate vectors* (one per vertex). These vectors contain one value for each boundary point of T and typically have length $\mathcal{O}(\sqrt{k})$. The path from z to y can then be found by computing only the distances from the path vertices and their neighbors to y , where each distance calculation requires $\mathcal{O}(\sqrt{k})$ operations [CGH17].

While a cost of $\mathcal{O}(\sqrt{k})$ per distance calculation does not seem too expensive, in practice it may still be too much for real-time performance. Furthermore, the coordinate vectors generated by the preprocessing procedure must be stored for each of the k vertices of the triangulation T , implying an $\mathcal{O}(k\sqrt{k})$ storage requirement, which could be prohibitive. This paper addresses both issues.

2. Contribution

This paper makes two main contributions. The first contribution is a more practical approach to path generation based on f -divergence distances. The key idea is to replace each of the k coordinate vectors with a very small set of $n \ll \mathcal{O}(\sqrt{k})$ reduced coordinates (see Section 4), without losing the important property that the distance function has no local minima (see Section 5). This reduces the cost of the distance calculations from $\mathcal{O}(\sqrt{k})$ to $\mathcal{O}(n)$. The small price to pay is that the generated gradient-descent paths may not be as natural as before, and we lose the properties of symmetry of the distance function and independence of the gradient direction on f .

The second contribution shows how to use reduced f -divergence distances in a purely discrete setting. While reducing the size of the coordinate vector from $\mathcal{O}(\sqrt{k})$ to n makes for an efficient computation of the distance function d_f , the actual path is generated using the triangulation T , implying a storage requirement of $\mathcal{O}(kn)$. In practice it would often be much more efficient to generate paths on a sparse network of $m \ll k$ sites in Ω . This requires building a suitable network of edges between the sites, one that supports *greedy routing*. In other words, for any site x , there must always be a neighbor of x with a smaller distance to the target y , which is the discrete analog of the continuous property of not having local minima. In Section 7 we describe an algorithm for building such a graph.

To illustrate the advantage of both contributions, consider a typical triangulation T of Ω with $k = 200,000$ vertices and 700 boundary vertices. In this case, the algorithm of Chen et al. [CGH17] requires storing 1.4×10^8 real values, and each distance calculation calls for 700 basic calculations. In contrast, using $n = 30$ reduced coordinates and a network with $m = 300$ sites requires storing only 9,000 real values and each distance calculation comes at the cost of just 30 basic calculations.

3. The f -divergence Distance

The f -divergence distance proposed in [CGH17] is based on the concept of f -divergence, which was first introduced by Kullback and Leibler [KL51] and later generalized by Csiszár [Csi67], for measuring the difference between two probability distributions.

Definition 1 Let f be a strictly convex function such that $f(1) = 0$ and $p, q: E \rightarrow [0, 1]$ be two real functions on some domain E such that $\int_E p(t) dt = \int_E q(t) dt = 1$. The f -divergence from q to p is

$$d_f(p, q) = \int_E p(t) f\left(\frac{q(t)}{p(t)}\right) dt.$$

It is well known that $d_f(p, q) \geq 0$ and

$$d_f(p, q) = 0 \iff p = q,$$

but d_f is not necessarily a metric. Many instances of f have been proposed over the years, each suitable for some specific application, mostly in probability theory, statistics, and information theory. The interested reader is referred to [LV06] for a survey of the possibilities.

In this context, it is noteworthy to recall the concept of the dual function

$$f^*(x) = xf(1/x).$$

For example, if $f(x) = -\log x$, then $f^*(x) = x \log x$, and if $f(x) = |1-x|$, then $f^*(x) = f(x)$. It is well known that

1. f is (strictly) convex if and only if f^* is (strictly) convex;
2. $d_f(p, q) = d_{f^*}(q, p)$;
3. $f(1) \leq d_f(p, q) \leq f(0) + f^*(0)$.

The f -divergence distance further relies on the *Poisson kernel* of Ω at $y \in \Omega$,

$$P(t, y) = \frac{\partial D}{\partial n}(t, y), \quad t \in \partial\Omega,$$

which is the derivative of the Green's function $D(x, y)$ for the Laplace operator with Dirichlet boundary conditions at the boundary point t , in the direction of the unit outer normal n at t [GM05]. The f -divergence distance is then defined as the f -divergence of two Poisson kernels.

Definition 2 Let f be a strictly convex function such that $f(1) = 0$ and $y, z \in \Omega$. The f -divergence distance from z to y is

$$d_f(y, z) = \oint_{\partial\Omega} P(t, y) f\left(\frac{P(t, z)}{P(t, y)}\right) dt. \quad (1)$$

Although the f -divergence of two probability functions is not necessarily symmetric, the special nature of the f -divergence distance implies that it is symmetric. In fact, f and f^* generate *identical* divergence distances, hence

$$d_f(y, z) = d_{f^*}(y, z) = d_f(z, y).$$

While the f -divergence distance is symmetric, it is not a metric in general, since it may fail to satisfy the triangle inequality.

For path-generation purposes, the gradient of the f -divergence distance plays a key role. Chen et al. [CGH17] show that for bounded and simply-connected domains the gradient field vanishes only at the target,

$$\nabla_z d_f(y, z) = 0 \iff z = y, \quad (2)$$

and that its direction is independent of f , that is,

$$\arg(\nabla_z d_f(y, z)) = \arg(\nabla_z d_g(y, z)),$$

for any two strictly convex functions f and g .

4. Reduced Coordinates

Since the Poisson kernel satisfies

$$\oint_{\partial\Omega} P(t, z) dt = 1, \quad \oint_{\partial\Omega} P(t, z) t dt = z,$$

it can be viewed as a continuous “coordinate vector” for $z \in \Omega$, which can be discretized as follows. Let $t_1, \dots, t_n \in \partial\Omega$ be a sequence of n points along the domain boundary. We then partition $\partial\Omega$ into the n segments $E_j = [t_j, t_{j+1}]$, $j = 1, \dots, n$, where t_{n+1} is identified with t_1 and define the *reduced coordinates* $(\phi_1(z), \dots, \phi_n(z))$ of $z \in \Omega$ as

$$\phi_j(z) = \int_{E_j} P(t, z) dt, \quad j = 1, \dots, n. \quad (3)$$

The quantity $\phi_j(z)$ is also called the *harmonic measure* of E_j at z [GM05]. Using reduced coordinates, we can now discretize the

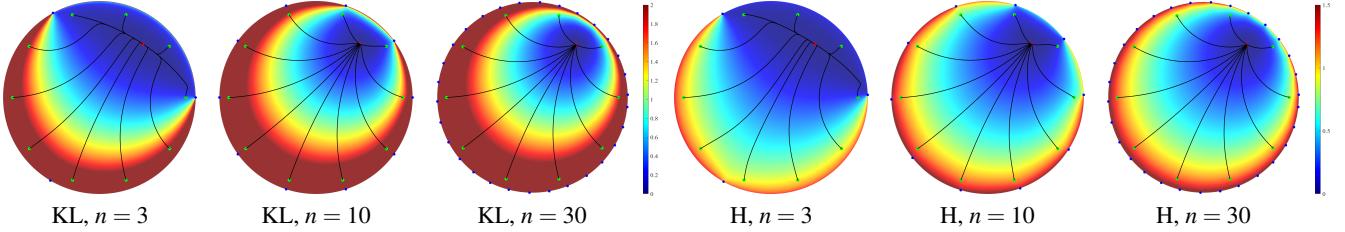


Figure 2: Paths from several source points (green) to a common target point (red), generated using reduced KL (top) and reduced H (bottom) divergence distance with respect to a partition of the boundary defined by the n boundary points (blue) for different n . Note that the KL paths are in general not identical to the H paths, but both converge to the original f -divergence paths (based on the continuous coordinates) in Figure 1 (right) as n increases.

f -divergence distance in (1), and define the *reduced f -divergence distance* from z to y as

$$d_f(y, z) = \sum_{j=1}^n \phi_j(y) f\left(\frac{\phi_j(z)}{\phi_j(y)}\right). \quad (4)$$

5. The Divergence Gradient Theorem

Our main theoretical result is that the reduced f -divergence distance can be used for generating gradient-descent paths.

Theorem 1 (Divergence Gradient Theorem) Let Ω be a bounded and simply-connected domain, f be a strictly convex function, and $n \geq 3$. Then the reduced f -divergence distance (4) has property (2).

We prove Theorem 1 for the special case of the unit disk with the target point at the origin in Appendix A (see Theorem A.1) and now show how to generalize it to arbitrary bounded and simply-connected domains. The key is the conformal invariance of the f -divergence distance, which in turn is implied by the well-known conformal invariance of the harmonic measure of $E \subset \partial\Omega$ at z ,

$$\phi^\Omega(E, z) = \int_E P^\Omega(t, z) dt.$$

Theorem 2 [GM05] Let Ω_1 and Ω_2 be two bounded and simply-connected domains and $C: \Omega_1 \rightarrow \Omega_2$ be a conformal map between them such that $C(\partial\Omega_1) = \partial\Omega_2$. Then,

$$\phi^{\Omega_2}(C(E), C(z)) = \phi^{\Omega_1}(E, z).$$

Theorem 3 Let $C: \Omega_1 \rightarrow \Omega_2$ be a conformal map and $d_f^{\Omega_1}: \Omega_1 \times \Omega_1 \rightarrow \mathbb{R}$ and $d_f^{\Omega_2}: \Omega_2 \times \Omega_2 \rightarrow \mathbb{R}$ be the f -divergence distances for Ω_1 and Ω_2 , respectively. Then,

$$d_f^{\Omega_2}(C(y), C(z)) = d_f^{\Omega_1}(y, z).$$

Proof By the definition of d_f in (4) and Theorem 2, we have

$$\begin{aligned} d_f^{\Omega_2}(C(y), C(z)) &= \sum_{j=1}^n \phi^{\Omega_2}(C(E_j), C(y)) f\left(\frac{\phi^{\Omega_2}(C(E_j), C(z))}{\phi^{\Omega_2}(C(E_j), C(y))}\right) \\ &= \sum_{j=1}^n \phi^{\Omega_1}(E_j, y) f\left(\frac{\phi^{\Omega_1}(E_j, z)}{\phi^{\Omega_1}(E_j, y)}\right) \\ &= d_f^{\Omega_1}(y, z) \end{aligned}$$

□

Now we are ready to prove Theorem 1 for any bounded and simply-connected domain by conformally mapping it to the unit disk.

Proof of Theorem 1 For a given target point $y \in \Omega$, the Riemann Mapping Theorem [GM05] implies that there exists a conformal map $C: \Omega \rightarrow D$, where D is the unit disk, such that $\partial\Omega$ is mapped to ∂D and $C(y) = 0$. By Theorem 3,

$$d_f^\Omega(y, z) = d_f^D(0, C(z)).$$

The gradients of the two distance functions with respect to their second argument are related by the chain rule for holomorphic functions. Dropping the first (fixed) argument, we get

$$\nabla d_f^\Omega(z) = \nabla d_f^D(C(z)) \frac{\partial C}{\partial z}(z).$$

Since the derivative of a conformal map never vanishes, according to Theorem A.1, we have

$$\begin{aligned} \nabla d_f^\Omega(z) = 0 &\iff \nabla d_f^D(C(z)) = 0 \iff C(z) = 0 \\ &\iff z = y. \end{aligned}$$

□

5.1. Examples

Figures 2 and 3 show what happens to the paths in Figure 1 (right) when the reduced f -divergence distance (4) is used instead of the original (continuous) f -divergence distance (1), for two choices of f , the Kullback–Leibler (KL) divergence $f(x) = -\log x$ and the Hellinger (H) divergence $f(x) = 2(1 - \sqrt{x})$. The coordinates are reduced by uniformly partitioning the boundary of the domain. For polygonal boundaries (as in Figure 3), it seems natural to partition according to the polygon edges, namely at least one reduced coordinate per edge. Long edges are further partitioned uniformly until the partition length is less than some threshold. Since reduced coordinates lose the property of being invariant to f , different paths are obtained for reduced KL and H distances. They are typically not as natural as the original paths, especially when n is small. Obviously, for very large n , the reduced paths approach the original paths. The same behavior can be observed for multiply-connected domains, as shown in Figure 4. Note that the distance function has saddle points in this case (typically one per “hole”), but no local minima. In all these examples, we discretize the domains with dense triangulations of 200,000 vertices, and the number of boundary vertices are listed in the second column of Table 1.

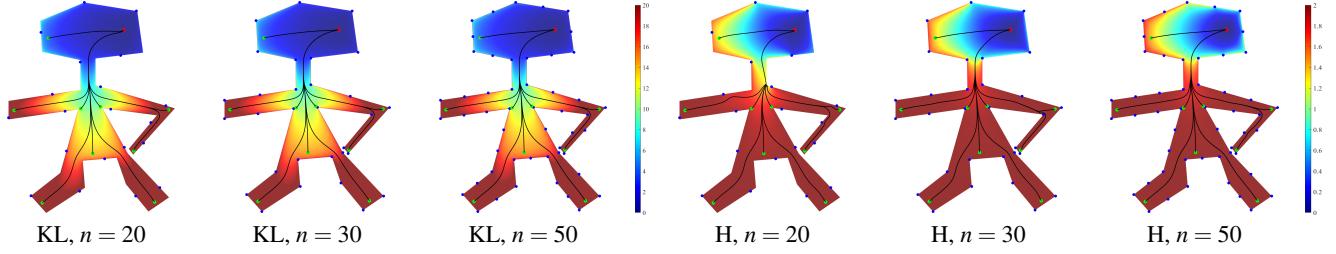


Figure 3: Same as Figure 2, for a non-convex polygonal domain. Convergence to the original f -divergence paths in Figure 1 (right) is more rapid for the KL divergence than for the H divergence.

Domain	Chen et al. [CGH17]				f -divergence distance with n reduced coordinates											
	n	preprocessing	online	memory	n	preprocessing	online	memory	n	preprocessing	online	memory				
Fig. 2	1,200	14.6	2.38	1,827.9	10	0.903	0.023	15.2	30	0.959	0.061	45.7				
Fig. 3	3,805	178.5	6.94	5,814.0	20	0.768	0.057	30.6	30	0.895	0.079	45.7				
Fig. 4 top	3,581	117.4	6.84	5,466.8	10	0.810	0.022	15.2	30	0.862	0.074	45.8				
Fig. 4 bottom	5,604	303.2	10.6	8,551.7	20	0.710	0.044	30.5	50	0.971	0.113	76.3				

Table 1: Runtime comparison of Chen et al.’s [CGH17] routing method using a full set of coordinates and our method, using reduced coordinates: times (in seconds) for preprocessing and online computation of the distance from one vertex to all other vertices, and memory cost (in megabytes).

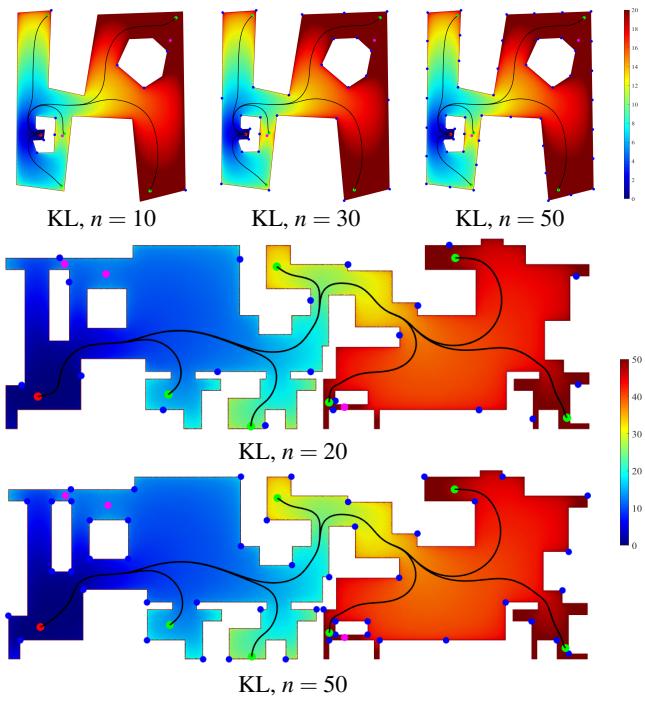


Figure 4: Same as Figure 2, for multiply-connected polygonal domains. The saddle points of the distance function are marked in magenta.

6. Computing Reduced Coordinates

In practice, if T is a dense triangulation of the domain Ω with k vertices x_1, \dots, x_k , and t_1, \dots, t_n is a sequence of points along the boundary of T , then the reduced coordinates $\phi_j(x_i)$, $j = 1, \dots, n$ of

all vertices can be computed as follows. Let A be the sparse, symmetric, and positive definite $k \times k$ matrix with the so-called cotangent weights [PP93] corresponding to edges of the triangulation, that is, the standard FEM discretization of the Laplace operator for T . Note that all weights are positive, if a constrained Delaunay triangulation of the domain [DBCvKO08, She96] is used. Further let $\Phi_j = (\phi_j(x_1), \dots, \phi_j(x_k))$ be the vector of all j -th reduced coordinates and $b_j = (b_{j,1}, \dots, b_{j,k})$ be the “binary indicator” of the segment $[t_j, t_{j+1}]$ of points along the boundary of T between t_j and t_{j+1} ,

$$b_{j,i} = \begin{cases} 1, & \text{if } x_i \in [t_j, t_{j+1}], \\ 0, & \text{otherwise.} \end{cases} \quad (5)$$

We then get all reduced coordinates by solving the n linear systems

$$A\Phi_j = b_j, \quad j = 1, \dots, n, \quad (6)$$

which can be done efficiently by pre-factoring A and performing back-substitution for the different b_j [PTVF07]. Note that the reduced coordinates are computed once for all vertices in a preprocessing stage. These are then used again and again for each “online” routing query (consisting of a pair of source and target points).

7. Discrete Routing Graph

We now show how to make path generation with reduced f -divergence distances even more practical. To this end, let S be a finite and sparse set of m sites in the domain Ω , and G be a graph on S , and consider paths running along the edges of G . To be able to mimic the gradient-descent idea in this discrete setting, we need a graph G that supports greedy routing to any $y \in S$,

$$\forall s \in S \setminus \{y\} : \exists r \in N_G(s) : d_f(y, r) < d_f(y, s), \quad (7)$$

where $N_G(s)$ is the set of neighbors of s in G .

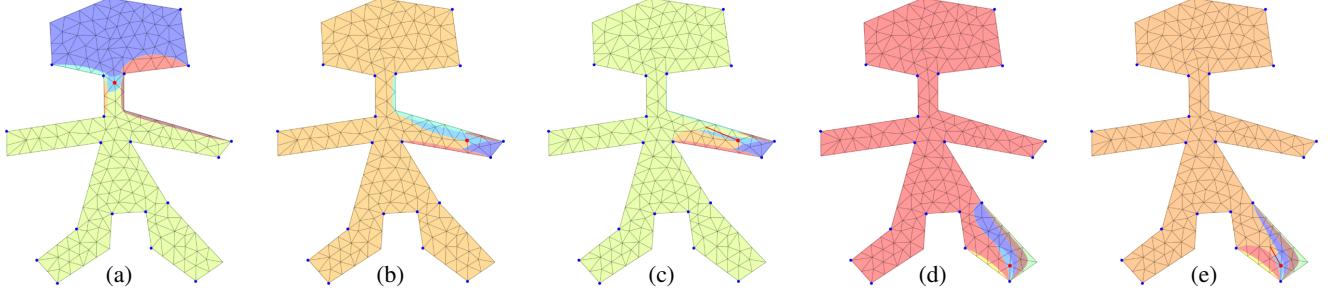


Figure 5: Construction of a greedy routing graph for the reduced KL divergence distance with $n = 20$ reduced coordinates and $m = 200$ sites sampled from a simply-connected domain. For the initial Euclidean Delaunay triangulation of the sites (a), it may happen that the local Voronoi cell (light blue) of a site (red) contains other sites (b,d). In this case, we add an edge (red) between these two sites (c,e), until the local Voronoi cells contain no other sites. Note how this affects the local Voronoi cells of the neighboring sites.

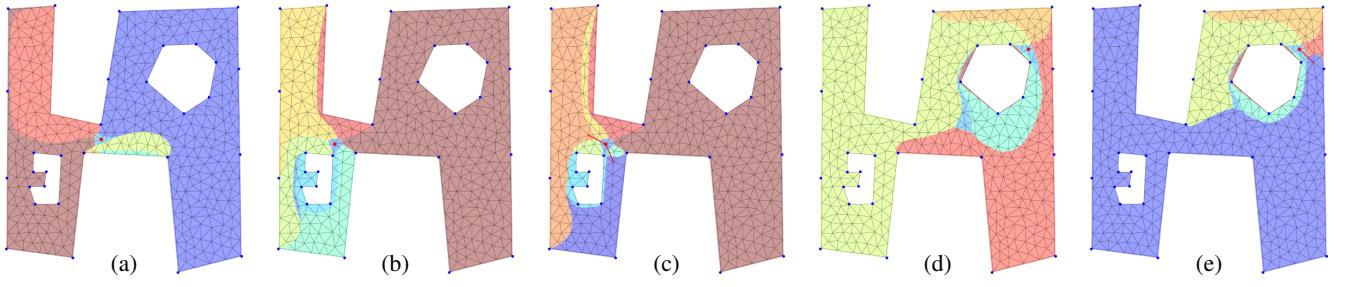


Figure 6: Same as Figure 5, for a multiply-connected domain with $n = 30$ reduced coordinates and $m = 400$ sites.

To achieve this, and inspired by [BCGGW11], we follow the logic of Bose and Morin [BM04], who show that the *Delaunay triangulation* [dBCvKO08] of S supports greedy routing on the convex hull of S , using the Euclidean distance between points in the plane. The reason for the Delaunay triangulation to have this property is because it is the dual of the Euclidean *Voronoi diagram* of S . That is, two sites are connected by an edge if and only if their two corresponding Voronoi cells share a common edge. The proof that greedy routing works relies on the fact that the Euclidean distance is a metric and its Voronoi cells are convex polygons. Since the Voronoi diagram of S using a reduced f -divergence distance is more complicated, the condition must be modified, requiring the concept of a *local Voronoi cell* of a site in a graph.

Definition 3 The local Voronoi cell of $s \in S$ relative to the graph G is the set of all points in Ω for which s is closer than any of the neighbors of s in G ,

$$LV_G(s) = \{z \in \Omega : d_f(z, s) < d_f(z, r) \forall r \in N_G(s)\}.$$

The *greedy routing property* now guarantees that (7) is satisfied.

Definition 4 The graph G has the *greedy routing property*, if for every site $s \in S$, $LV_G(s)$ does not contain any site other than s ,

$$\forall s, t \in S : t \in LV_G(s) \iff t = s.$$

Constructing a graph G on S having the greedy routing property is not as straightforward as it seems. It is not sufficient to merely take the dual of the Voronoi diagram of S , because the local Voronoi cells of the reduced f -divergence distance may have irregular structure. They need not even be connected and can have so-called or-

phan cells, and the reduced f -divergence distance may be asymmetric. This is especially true when the number n of coordinates is small. On the other hand, the clique graph, which connects all sites with each other, obviously has the greedy routing property, but this is a gross overkill, as we would like to have a graph that is as sparse as possible, with edges as short as possible. A planar graph would be the most desirable.

In order to build a greedy routing graph G , given S and d_f , we propose an incremental algorithm. Starting with the Euclidean Delaunay triangulation, constrained to a polygonal approximation of the domain, we *augment* this initial graph with additional edges until it becomes greedy. Given a site $s \in S$, edges are added to G between s and other sites, until $LV_G(s)$ contains only s . By definition, each addition of an edge shrinks $LV_G(s)$. Obviously, this procedure eventually terminates when the worst case of s being connected to all other sites is obtained. A good heuristic is to add edges between s and other sites in order of increasing Euclidean distance to s . We call the resulting greedy graph the *augmented Delaunay triangulation*. In practice, no Voronoi diagrams are computed, and the only data structure required to support the graph construction algorithm is the $m \times m$ matrix of pairwise reduced f -divergence distances between the m sites, sampled from the dense underlying triangulation that we use to precompute the coordinates. Note that this matrix is not symmetric if d_f is not symmetric.

8. Experimental Results

We implemented our method in Matlab R2018a on a PC with Intel Xeon (6 cores) CPU E5-2643 v4 3.40Ghz and 32GB RAM. For

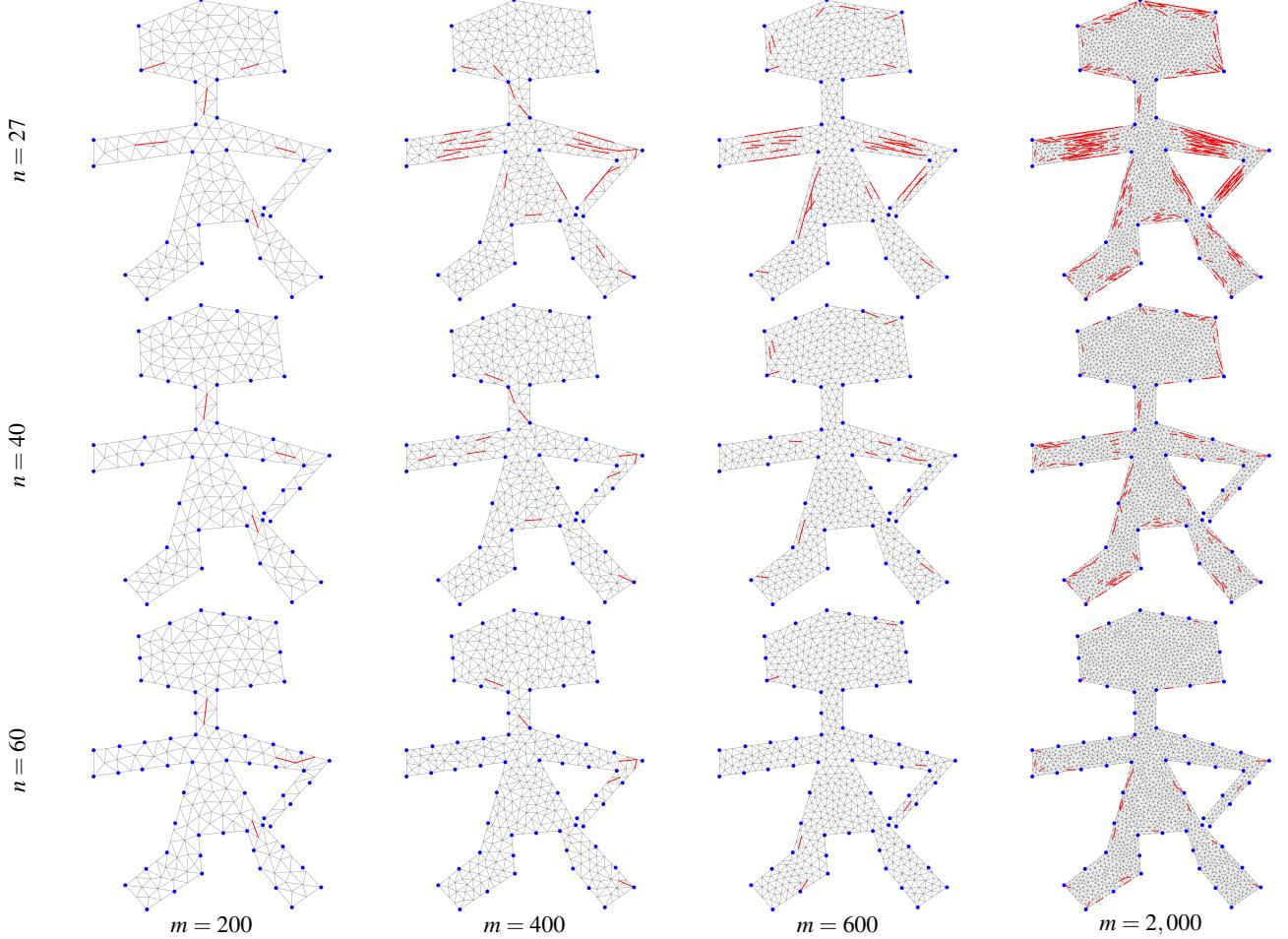


Figure 7: Discrete greedy routing graph using the KL divergence distance with n reduced coordinates and m sites on a simply connected domain for different values of n and m . The greedy routing property is obtained after augmenting the initial constrained Delaunay triangulation (black) with additional edges (red). Note how the number of augmented edges decreases with increasing n .

Domain	n	$m = 200$		$m = 400$		$m = 600$		$m = 2,000$	
		distances	augmenting	distances	augmenting	distances	augmenting	distances	augmenting
Fig. 7	27	0.033	0.0149	0.090	0.0084	0.191	0.0138	1.344	0.1247
	40	0.059	0.0142	0.111	0.0082	0.243	0.0109	1.732	0.0837
	60	0.078	0.0118	0.147	0.0089	0.319	0.0092	2.472	0.0757
Fig. 8	40	0.050	0.0152	0.103	0.0063	0.234	0.0117	1.707	0.0808
	60	0.081	0.0165	0.177	0.0056	0.329	0.0122	2.675	0.0783

Table 2: Runtime for generating discrete routing graphs. For each configuration of m and n , we list the times (in seconds) for computing the pairwise reduced f -divergence distances between the m sites and for augmenting the constrained Delaunay triangulation.

all the examples throughout the paper, the underlying constrained Delaunay triangulation T of the domain on which the n coordinates are computed contains $k = 200,000$ vertices. Table 1 lists the runtimes for computing n reduced coordinates and f -divergence distances (from one vertex to all other vertices of T), and the memory footprint for storing the reduced coordinates, in comparison to Chen et al. [CGH17], where n is the number of boundary vertices of T . We observe that our method is faster both in preprocessing and online distance computation, and the memory requirement is

much lower. Note that the system matrices involved in the two approaches are identical, yet the preprocessing takes significant more time for Chen et al. [CGH17], since they need to solve for significantly more right hand sides in order to get a good approximation of the harmonic measure.

Figures 5 and 6 show parts of the greedy routing graphs generated by our algorithm on a few sites sampled in two different domains with 3,805 and 3,581 boundary vertices, respectively. The

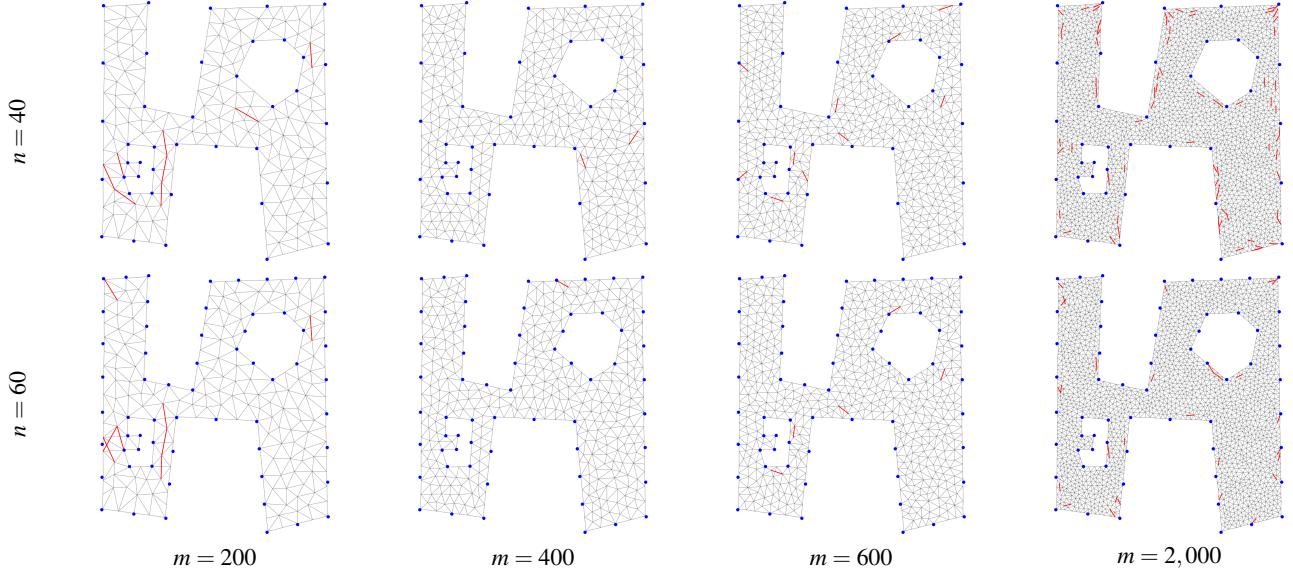


Figure 8: Same as Figure 7, for a multiply-connected domain.

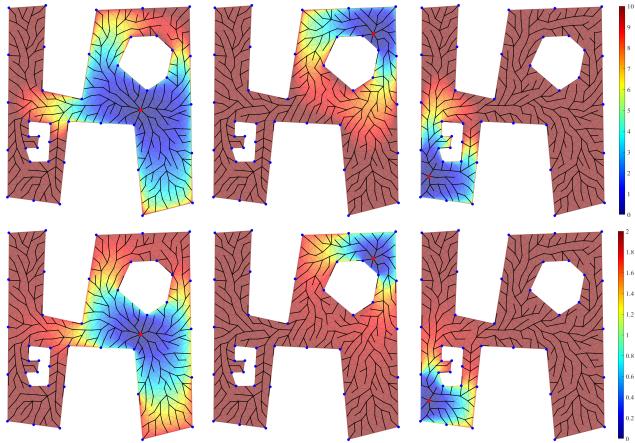


Figure 9: Routing path tree (thick black edges) for three different target sites (red) for the multiply-connected domain and graph of Figure 8 for $n = 40$ reduced coordinates and $m = 400$ sites using the KL (top) and the H (bottom) divergence distance. The domain is color-coded according to the f -divergence distance of the reduced coordinates to the target site. Note the subtle differences between the two rows.

black edges are the initial constrained Delaunay triangulation, and the red edges are those augmented by our algorithm for three select sites to obtain the greedy routing property.

Figures 7 and 8 show the greedy augmented Delaunay triangulation on three increasing sets of sites with increasing sets of reduced coordinates using a similar coloring of the edges. As before, the boundary polygon was naturally partitioned to its edges, and then each edge further partitioned to segments until the length of each segment is below a predefined threshold. The results demonstrate

how the Delaunay triangulation is already very close to greedy, requiring only a slight augmentation, for large values of n . Table 2 lists the runtimes for generating the greedy routing graphs of the augmented Delaunay triangulations. Note that all this is done in a preprocessing stage. Figure 9 shows routing trees for three target vertices on the resulting graph for two different distance functions on a multiply-connected shape.

The idea of using distance functions for path planning in robotic applications promises to remain a popular methodology. Indeed, a recent paper of Chen et al. [CLL*16] uses diffusion distances for this objective. Similarly to our divergence distances, diffusion distances may be computed efficiently using a small set of “embedding coordinates” derived from the eigenvectors of the graph Laplacian having the largest eigenvalues, computed in a preprocessing stage. However, this distance function is not void of local minima, so the “best-first-search” algorithm is used to generate paths on a discrete network, incurring overhead by examining many neighboring nodes in the vicinity of the path and maintaining a priority queue. Note that if the distance function is void of local minima, best-first-search reduces to gradient descent and a priority queue is not required. In principle, we could have adopted a similar approach and applied best-first-search instead of gradient descent, avoiding the need to augment the graph. However, as our experimental results indicate, the augmentation required is minimal for sparse triangulation and most of the time unnecessary for sufficiently densely triangulated domains, due to the fact that the continuous version of our reduced divergence distance is greedy, as compared to the continuous diffusion distance, which does not have this desirable property. Thus we believe it is preferable to pay the small price of augmenting the graph when necessary in the preprocessing stage and benefiting significantly from a very simple routing algorithm in the online query stage. Figure 10 compares our method with harmonic potential, A* search [HNR68] (with the Euclidean distance as a heuristic function), and diffusion search [CLL*16] (based on 18

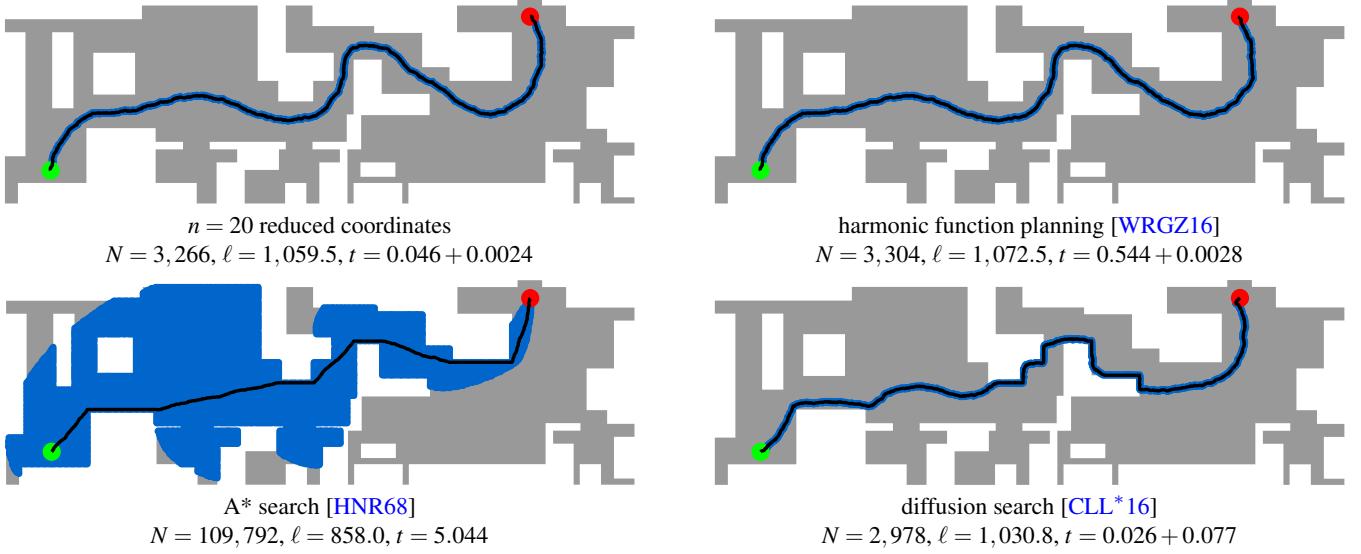


Figure 10: Comparison of different path planning algorithms. The black polyline marks the path between the source (green) and target (red) found by each algorithm. To determine this path, each algorithm needs to compute the distance function for the N vertices in the blue region around the path. We report N , the length ℓ of each path, and the planning time t (in seconds). Note that t does not include the preprocessing time (0.737 s for our method and 93 s for diffusion search), but for harmonic function planning we include the linear solve time of 0.544 s, as it has to be computed online after the routing target is selected, and for our method and diffusion search we include the 0.046 s and 0.026 s that it takes, respectively, to compute the distance from the target to every other vertex.

eigenvectors) for the laboratory map from Wray et al. [WRGZ16]. For harmonic function planning, we directly compute the potential function by a sparse linear solver instead of its logarithm as in [WRGZ16], as it is more efficient and provides sufficient accuracy for greedy routing in this case. For diffusion search, we used a time step of 200 times the width of the domain, because the factor of 50 suggested in [CLL*16] gives significantly worse values of N and t .

9. Conclusion and Discussion

We have described a practical method for generating paths between pairs of points in planar domains by following the negative gradient of a suitable distance function. The key is to use a discrete version of the f -divergence distance, based on reduced coordinate vectors. Our reduced coordinate vectors are the harmonic measures of a partition of the boundary into n segments, which are just the inner products of the Poisson kernel with a “box” indicator function for each segment. We speculate that the Divergence Gradient Theorem 1 also holds for other sets of coordinates, derived from an inner product of the Poisson kernel with more sophisticated basis functions, such as the piecewise linear “tent” function over two adjacent boundary segments. This is the method used to construct *harmonic barycentric coordinates* on a polygonal domain [JMD*07]. Another possible choice are Gaussian basis functions over the boundary, as long as they are not too narrow or too wide. These variants of reduced coordinates may be computed similarly to the reduced coordinates used in this paper merely by changing the right hand sides b_j of the linear equations in (6) to something more sophisticated than the binary vector in (5). Figure 11 shows the gradient-descent

trees generated by these coordinates using the KL divergence distance function.

The proof of our Divergence Gradient Theorem 1 holds only for bounded and simply-connected domains, as it relies on conformal reduction to the canonical case of a unit disk with the target at the origin. We do not have a proof for the absence of local minima for multiply-connected domains, where the holes in the domain correspond to obstacles in a real-world scenario, but speculate that the theorem still holds. This is supported by all our experiments, as well as a recent result on the f -divergence distance for the χ^2 -divergence $f(x) = x^2 - 1$ in the continuous setting [CGH18]. Note that for multiply-connected domains, the distance function may contain saddle points, where the continuous gradient vanishes, but as these are not local minima, they are not fatal for path generation.

Our experimental results also indicate that as the number of reduced coordinates increases, the augmentation of the Delaunay triangulation of the sites decreases. We wonder if there exists a condition, possibly dependent on n , the number of coordinates, and m , the number of sites, which guarantees that the Delaunay triangulation is greedy in its own right. Alternatively, under which conditions does there exist a planar graph, possibly a non-Delaunay triangulation, of the sites, which is greedy? This may well be the dual of the Voronoi diagram for d_f .

It is worth noting that the harmonic potential with pole at the target point $y \in \Omega$, which has traditionally been used for path planning, can also be expressed as a boundary integral once the Poisson kernel of the domain has been computed:

$$D(y, z) = -\log|z - y| + \oint_{\partial\Omega} P(t, z) \log|t - y| dt.$$

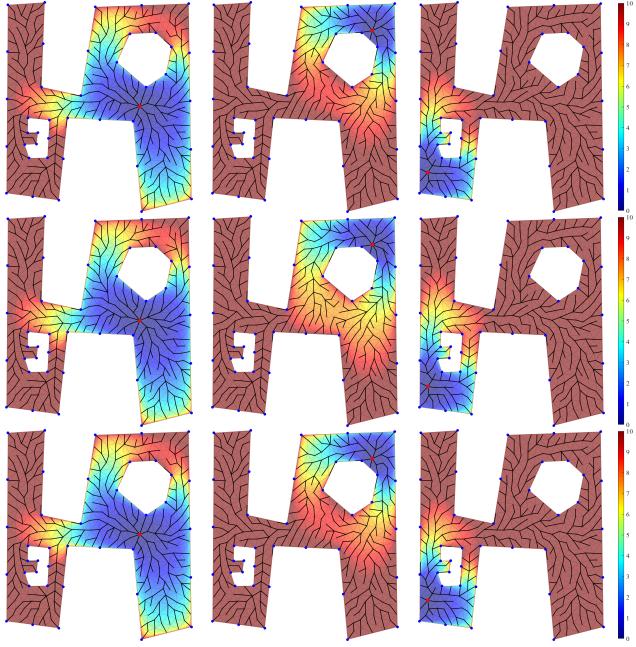


Figure 11: Same as in Figure 9, for $n = 40$ reduced coordinates and $m = 400$ sites using the KL divergence distance and reduced coordinates based on box (top), tent (middle), and Gaussian (bottom) basis functions over the boundary.

So theoretically, we could compute this potential quite efficiently at any given point $z \in \Omega$ as a boundary integral (once the Poisson kernels are computed in a preprocessing stage), instead of solving a large system of linear equations. Thus the natural question arises: Why is it better to compute the divergence distance as a boundary integral and not the harmonic potential?

One reason is that, in the discrete case, the harmonic potential computed as a boundary integral is not discrete harmonic (namely, does not satisfy the discrete Laplace equation), thus may contain local minima. This is in contrast to explicitly solving the discrete Laplace equation. So, to use it in practice, we will have to apply graph augmentation, which may be more excessive than in the case of the augmentation of the graph by divergence distance.

More importantly, the harmonic potential is much less attractive because we cannot apply coordinate reduction to reduce the complexity of the boundary integral, which we can do for the divergence distance, thanks to Theorem 1.

References

- [ABR01] AXLER S., BOURDON P., RAMEY W.: *Harmonic Function Theory*, 2nd ed., vol. 137 of *Graduate Texts in Mathematics*. Springer, New York, 2001. 2
- [BCGGW11] BEN-CHEN M., GORTLER S. J., GOTSMAN C., WORMSER C.: Distributed computation of virtual coordinates for greedy routing in sensor networks. *Discrete Appl. Math.* 159, 7 (Apr. 2011), 544–560. 6
- [BM04] BOSE P., MORIN P.: Online routing in triangulations. *SIAM J. Comput.* 33, 4 (2004), 937–951. 6
- [CG93] CONNOLLY C. I., GRUPEN R. A.: The applications of harmonic functions to robotics. *J. Field Robotics* 10, 7 (Oct. 1993), 931–946. 2
- [CGH17] CHEN R., GOTSMAN C., HORMANN K.: Path-planning with divergence-based distance functions. arXiv:1708.05855 [cs.RO], 2017. <https://arxiv.org/pdf/1708.05855.pdf>. 2, 3, 5, 7
- [CGH18] CHEN R., GOTSMAN C., HORMANN K.: On divergence-based distance functions for multiply-connected domains. arXiv:1801.07099 [math.CV], 2018. <https://arxiv.org/pdf/1801.07099.pdf>. 9
- [CLL*16] CHEN Y. F., LIU S.-Y., LIU M., MILLER J., HOW J. P.: Motion planning with diffusion maps. In *Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems* (Daejeon, Oct. 2016), pp. 1423–1430. 8, 9
- [Csi67] CSISZÁR I.: Information-type measures of difference of probability distributions and indirect observations. *Stud. Sci. Math. Hung.* 2 (1967), 299–318. 2, 3
- [dBCvKO08] DE BERG M., CHEONG O., VAN KREVELD M., OVERMARS M.: *Computational Geometry: Algorithms and Applications*, 3rd ed. Springer, Berlin, 2008. 5, 6
- [GM05] GARNETT J. B., MARSHALL D. E.: *Harmonic Measure*, vol. 2 of *New Mathematical Monographs*. Cambridge University Press, New York, 2005. 2, 3, 4
- [HNR68] HART P. E., NILSSON N. J., RAPHAEL B.: A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans. Syst. Sci. Cybern.* 4, 2 (July 1968), 100–107. 2, 8, 9
- [JMD*07] JOSHI P., MEYER M., DEROSE T., GREEN B., SANOCKI T.: Harmonic coordinates for character articulation. *ACM Trans. Graph.* 26, 3 (July 2007), Article 71, 9 pages. 9
- [KB91] KOREN Y., BORENSTEIN J.: Potential field methods and their inherent limitations for mobile robot navigation. In *Proceedings of the 1991 IEEE International Conference on Robotics and Automation* (Sacramento, Apr. 1991), pp. 1398–1404. 2
- [Kha86] KHATIB O.: Real-time obstacle avoidance for manipulators and mobile robots. *Int. J. Robotics Res.* 5, 1 (1986), 90–98. 2
- [KK92] KIM J.-O., KHOSLA P. K.: Real-time obstacle avoidance using harmonic potential functions. *IEEE Trans. Robotics Autom.* 8, 3 (June 1992), 338–349. 2
- [KL51] KULLBACK S., LEIBLER R. A.: On information and sufficiency. *Ann. Math. Stat.* 22, 1 (Mar. 1951), 79–86. 2, 3
- [LV06] LIESE F., VAJDA I.: On divergences and informations in statistics and information theory. *IEEE Trans. Inf. Theory* 52, 10 (Oct. 2006), 4394–4412. 3
- [PP93] PINKALL U., POLTHIER K.: Computing discrete minimal surfaces and their conjugates. *Exp. Math.* 2, 1 (1993), 15–36. 5
- [PTVF07] PRESS W. H., TEUKOLSKY S. A., VETTERLING W. T., FLANNERY B. P.: *Numerical Recipes in C: The Art of Scientific Computing*, 3rd ed. Cambridge University Press, New York, 2007. 5
- [RK92] RIMON E., KODITSCHEK D. E.: Exact robot navigation using artificial potential functions. *IEEE Trans. Robotics Autom.* 8, 5 (Oct. 1992), 501–518. 2
- [SBD*13] SOUSSI O., BENATITALLAH R., DUVIVIER D., ARTIBA A., BELANGER N., FEYZEAU P.: Path planning: A 2013 survey. In *Proceedings of the 2013 IEEE International Conference on Industrial Engineering and Systems Management* (Rabat, Oct. 2013), pp. 849–856. 2
- [She96] SHEWCHUK J. R.: Triangle: Engineering a 2D quality mesh generator and Delaunay triangulator. In *Applied Computational Geometry. Towards Geometric Engineering*, Lin M. C., Manocha D., (Eds.), vol. 1148 of *Lecture Notes in Computer Science*. Springer, Berlin, 1996, pp. 203–222. 5
- [WRGZ16] WRAY K. H., RUIKEN D., GRUPEN R. A., ZILBERSTEIN S.: Log-space harmonic function path planning. In *Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems* (Daejeon, Oct. 2016), pp. 1511–1516. 9

Appendix A:

In this appendix, we prove Theorem 1 for the special case of the unit disk domain with the target point at the origin.

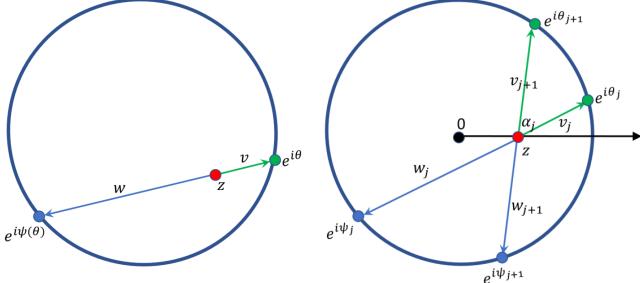


Figure 12: Notation used in Sections A.1 (left) and A.2 (right).

A.1. Some circle geometry

In what follows we use complex number algebra in the plane. Let z be some point in the unit disk D . As shown in Figure 12 (left), we denote by $\psi(\theta) \in (-\pi, \pi]$ the *antipode* of $\theta \in (-\pi, \pi]$ relative to z , that is, $e^{i\psi(\theta)}$ is the intersection of the chord through $e^{i\theta}$ and z with the unit circle. We further let $v = e^{i\theta} - z$ and $w = e^{i\psi(\theta)} - z$.

Lemma A.1 For any $\theta \in (-\pi, \pi]$,

$$\frac{1}{\bar{v}} = -\frac{w}{1 - |z|^2}.$$

Proof By the intersecting chords theorem, $|v||w| = 1 - |z|^2$. Since v and w are collinear, this implies $-\bar{v}w = |v||w| = 1 - |z|^2$. \square

Lemma A.2 For any $\theta \in (-\pi, \pi]$,

$$\psi(\theta) = -i \log \left(-\frac{v}{\bar{v}} e^{-i\theta} \right).$$

Proof By the definition of $\psi(\theta)$, we have $\psi(\theta) = -i \log(w + z)$, and it follows from Lemma A.1 that $w + z = -ve^{-i\theta}/\bar{v}$. \square

Note that for $z = 0$, we have $\psi(\theta) = \pi + \theta$, as expected.

A.2. Reduced coordinates

Given a partition $-\pi < \theta_1 < \dots < \theta_n \leq \pi$ of the unit circle, the j -th reduced coordinate of $z \in D$ is the harmonic measure of the arc (θ_j, θ_{j+1}) at z , that is,

$$\phi_j(z) = \int_{\theta_j}^{\theta_{j+1}} P^D(\theta, z) d\theta = \frac{1}{2\pi} (2\alpha_j(z) - (\theta_{j+1} - \theta_j)), \quad (8)$$

where $P^D(\theta, z)$ is the Poisson kernel of the unit disk,

$$P^D(\theta, z) = \frac{1}{2\pi} \frac{1 - |z|^2}{|z - e^{i\theta}|^2},$$

and $\alpha_j(z)$ denotes the angle the arc forms with z , as shown in Figure 12 (right). Note that since $e^{i\theta}$ is a 2π -periodic function, we may use $\theta_{n+1} = \theta_1 + 2\pi$, so that $\theta_{j+1} - \theta_j$ always gives the positive length of the arc, and then $\phi_j(z) \geq 0$ and $\sum_j \phi_j(z) = 1$.

Lemma A.3 The harmonic measure in (8) can also be expressed as

$$\phi_j(z) = \frac{1}{2\pi} (\psi_{j+1} - \psi_j), \quad (9)$$

where $\psi_j = \psi(\theta_j)$ is the antipode of θ_j relative to z .

Proof By the intersecting chords theorem, $\theta_{j+1} - \theta_j + \psi_{j+1} - \psi_j = 2\alpha_j(z)$, and the statement then follows directly from (8). \square

Lemma A.4 The gradient (by z) of $\phi_j(z)$ is

$$\nabla \phi_j(z) = \frac{i}{\pi(1 - |z|^2)} (e^{i\psi_{j+1}} - e^{i\psi_j}).$$

Proof By Lemma A.3 and viewing ψ_j as a function of z , we get

$$\nabla \phi_j(z) = \frac{1}{2\pi} \nabla (\psi_{j+1}(z) - \psi_j(z)).$$

Using Lemma A.2 and the complex form of the gradient then gives

$$\nabla \psi_j(z) = -i \nabla \log \left(\frac{-v_j}{\bar{v}_j} \right) = -2i \left(\frac{\partial}{\partial z} \log \left(\frac{-v_j}{\bar{v}_j} \right) \right) = -\frac{2i}{\bar{v}_j},$$

so

$$\nabla \phi_j(z) = -\frac{i}{\pi} \left(\frac{1}{\bar{v}_{j+1}} - \frac{1}{\bar{v}_j} \right).$$

Applying Lemma A.1, we finally get

$$\nabla \phi_j(z) = \frac{i}{\pi(1 - |z|^2)} (w_{j+1} - w_j) = \frac{i}{\pi(1 - |z|^2)} (e^{i\psi_{j+1}} - e^{i\psi_j}). \quad \square$$

A.3. Reduced f -divergence distance

Given a partition $-\pi < \theta_1 < \dots < \theta_n \leq \pi$ of the unit circle and a strictly convex function f , the reduced f -divergence distance in (4) from $z \in D$ to $y = 0$ can be written as

$$d_f(z) = d_f(0, z) = \sum_{j=1}^n \phi_j(0) f \left(\frac{\phi_j(z)}{\phi_j(0)} \right), \quad (10)$$

with $\phi_j(z)$ in (9). By the chain rule and Lemma A.4,

$$\begin{aligned} \nabla d_f(z) &= \sum_{j=1}^n \phi_j(0) \nabla f \left(\frac{\phi_j(z)}{\phi_j(0)} \right) = \sum_{j=1}^n f' \left(\frac{\phi_j(z)}{\phi_j(0)} \right) \nabla \phi_j(z) \\ &= \frac{i}{\pi(1 - |z|^2)} \sum_{j=1}^n f' \left(\frac{\psi_j - \psi_{j+1}}{\theta_j - \theta_{j+1}} \right) (e^{i\psi_{j+1}} - e^{i\psi_j}). \end{aligned}$$

Without loss of generality, let us now assume that z is on the positive x -axis, so that $z = \bar{z} = |z|$. Any other case can be reduced to this setting by a simple rotation of the plane.

Lemma A.5 The function $g(\psi) = f'(1/\theta'(\psi))$ is strictly increasing for $\psi \in (0, \pi)$ and strictly decreasing for $\psi \in (-\pi, 0)$, if $z \neq 0$, and constant, if $z = 0$.

Proof It follows from Lemma A.2 that

$$\psi(\theta) = -i \log \left(-\frac{e^{i\theta} - z}{e^{-i\theta} - z} e^{-i\theta} \right),$$

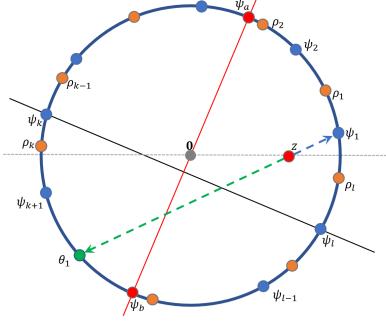


Figure 13: Notation used in the proof of Theorem A.1.

hence

$$\begin{aligned}\psi'(\theta) &= \frac{e^{i\theta}}{e^{i\theta}-z} + \frac{e^{-i\theta}}{e^{-i\theta}-z} - 1 = \frac{z}{e^{i\theta}-z} + \frac{z}{e^{-i\theta}-z} + 1 \\ &= 2z \operatorname{Re}\left(\frac{1}{e^{i\theta}-z}\right) + 1\end{aligned}$$

and

$$\begin{aligned}\psi''(\theta) &= -\frac{iz e^{i\theta}}{(e^{i\theta}-z)^2} + \frac{iz e^{-i\theta}}{(e^{-i\theta}-z)^2} = 2z \operatorname{Im}\left(\frac{e^{i\theta}}{(e^{i\theta}-z)^2}\right) \\ &= \frac{-2z(1-z^2)}{|e^{i\theta}-z|^4} \sin \theta \begin{cases} = 0, & \text{if } z = 0, \\ < 0, & \text{if } z \neq 0 \text{ and } \theta \in (0, \pi), \\ > 0, & \text{if } z \neq 0 \text{ and } \theta \in (-\pi, 0). \end{cases}\end{aligned}$$

Since $\theta(\psi)$ and $\psi(\theta)$ are antipodes of each other, they have the same behavior, and therefore

$$\theta''(\psi) \begin{cases} = 0, & \text{if } z = 0, \\ < 0, & \text{if } z \neq 0 \text{ and } \psi \in (0, \pi), \\ > 0, & \text{if } z \neq 0 \text{ and } \psi \in (-\pi, 0). \end{cases} \quad (11)$$

Applying the chain rule to $g(\psi)$, we find that

$$g'(\psi) = -f''\left(\frac{1}{\theta'(\psi)}\right) \frac{\theta''(\psi)}{\theta'(\psi)^2},$$

and as $f'' > 0$, since f is strictly convex, we conclude that the sign of g' is opposite to the sign of $\theta''(\psi)$. The statement then follows from (11). \square

Note that both $g(\psi)$ and $\psi'(\theta)$ are even functions.

Theorem A.1 (Divergence Gradient Theorem for the unit disk)

For any partition $-\pi < \theta_1 < \dots < \theta_n \leq \pi$ of the unit circle with $n \geq 3$, the reduced f -divergence distance to the origin in (10) satisfies

$$\nabla d_f(z) = 0 \iff z = 0.$$

Proof For each arc (θ_j, θ_{j+1}) , the mean-value theorem states that there exists some $\tau_j \in (\theta_j, \theta_{j+1})$, such that $\psi'(\tau_j) = \frac{\psi_j - \psi_{j+1}}{\theta_j - \theta_{j+1}}$. Now let $\rho_j = \psi(\tau_j) \in (\psi_j, \psi_{j+1}) \subset (-\pi, \pi]$ and define the piecewise constant periodic function $w(\psi) = g(\rho_j)$, $\psi \in (\psi_j, \psi_{j+1}]$. Obviously there exist k and l such that $|\rho_k| \leq |\rho_j| \leq |\rho_l|$ for all j , that is, the indices k and l correspond to the leftmost and rightmost ρ 's, respectively. In particular, as shown in Figure 13, l is either 1, $n-1$, or n . Since g is strictly increasing in $(0, \pi)$ and strictly decreasing

in $(-\pi, 0)$ by Lemma A.5, $w(\psi)$ is monotonically (but not strictly) increasing in $(\psi_l, \psi_k]$ (if $\psi_k > 0$, otherwise we use $(\psi_l, \psi_k + 2\pi]$ and the same principle applies below), and monotonically decreasing in $(\psi_k, \psi_l]$ by construction. Now let $[e^{i\psi_a}, e^{i\psi_b}]$ be the diameter orthogonal to $[e^{i\psi_k}, e^{i\psi_l}]$. The gradient of d_f can then be written as

$$\begin{aligned}\nabla d_f(z) &= \frac{i}{\pi(1-|z|^2)} \oint_C w(\psi) de^{i\psi} \\ &= \frac{i}{\pi(1-|z|^2)} \left(\int_{\psi_l}^{\psi_k} w(\psi) de^{i\psi} + \int_{\psi_k}^{\psi_l} w(\psi) de^{i\psi} \right) \\ &= \nabla d_f^1 + \nabla d_f^2\end{aligned}$$

and the projection of ∇d_f^1 onto $[e^{i\psi_l}, e^{i\psi_k}]$ be expressed as

$$\begin{aligned}\langle ie^{i\psi_a}, \nabla d_f^1 \rangle &= \left\langle ie^{i\psi_a}, \frac{i}{\pi(1-|z|^2)} \int_{\psi_l}^{\psi_k} w(\psi) de^{i\psi} \right\rangle \\ &= \frac{-1}{\pi(1-|z|^2)} \left\langle e^{i\psi_a}, \int_{\psi_l}^{\psi_k} iw(\psi) e^{i\psi} d\psi \right\rangle \\ &= \frac{-1}{\pi(1-|z|^2)} \int_{\psi_l}^{\psi_k} \operatorname{Re}(-iw(\psi) e^{i(\psi_a - \psi)}) d\psi \\ &= \frac{-1}{\pi(1-|z|^2)} \int_{\psi_l}^{\psi_a} w(\psi) \sin(\psi_a - \psi) d\psi \\ &\quad + \frac{-1}{\pi(1-|z|^2)} \int_{\psi_a}^{\psi_k} w(\psi) \sin(\psi_a - \psi) d\psi \\ &= \frac{-1}{\pi(1-|z|^2)} \int_{\psi_l}^{\psi_a} (w(\psi) - w(2\psi_a - \psi)) \sin(\psi_a - \psi) d\psi.\end{aligned}$$

Now observe that for $\psi \in (\psi_l, \psi_a)$ we have

$$2\psi_a - \psi \in (\psi_a, 2\psi_a - \psi_l) = (\psi_a, \psi_k),$$

and therefore, since w is monotonically increasing in $(\psi_l, \psi_k]$,

$$w(\psi) \leq w(2\psi_a - \psi).$$

Moreover, $\sin(\psi_a - \psi) > 0$, because $\psi_a - \psi \in (0, \psi_a - \psi_l) \subset (0, \pi)$. Overall, we conclude that

$$\int_{\psi_l}^{\psi_a} (w(\psi) - w(2\psi_a - \psi)) \sin(\psi_a - \psi) d\psi \leq 0,$$

hence $\langle ie^{i\psi_a}, \nabla d_f^1 \rangle \geq 0$, with equality if and only if $w(\psi)$ is constant on (ψ_l, ψ_k) , which happens only in the case $k = l + 1$, when the harmonic measure of the arc (θ_l, θ_k) reduces to a single coordinate. Similarly, the projection of ∇d_f^2 onto $[e^{i\psi_l}, e^{i\psi_k}]$ satisfies

$$\langle ie^{i\psi_a}, \nabla d_f^2 \rangle = \left\langle ie^{i\psi_a}, \frac{i}{\pi(1-|z|^2)} \int_{\psi_k}^{\psi_l} w(\psi) de^{i\psi} \right\rangle \geq 0,$$

again with equality if and only if $w(\psi)$ is constant on (ψ_k, ψ_l) , or equivalently $l = k + 1$. Overall, we conclude $\langle ie^{i\psi_a}, \nabla d_f(z) \rangle > 0$, hence $\nabla d_f(z) \neq 0$, as long as $n > 2$. If $n = 2$, then w is constant over the two integral intervals, hence $\langle ie^{i\psi_a}, \nabla d_f(z) \rangle = 0$. \square

Remark A.1 For $n = 2$, the proof of Theorem A.1 shows that $\nabla d_f(z)$ is orthogonal to $[e^{i\psi_l}, e^{i\psi_k}]$ and may vanish for certain z . In fact, Eq. (8) implies that the harmonic measure of the arc (θ_1, θ_2) at any z on the circle through $e^{i\theta_1}$, $e^{i\theta_2}$, and the origin is equal to the harmonic measure at the origin itself, so that $d_f(z) = 0$ and $\nabla d_f(z) = 0$.