TRANSPORTATION TECHNIQUES FOR GEOMETRIC DATA PROCESSING

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Justin Solomon

June 2015

Abstract

Modeling and understanding low- and high-dimensional data is a recurring theme in graphics, optimization, learning, and vision. Abstracting away application domains reveals common threads using geometric constructs like distances, similarities, and curvatures. This shared structure suggests the possibility of developing geometric data processing as a discipline in itself.

To this end, this thesis introduces optimal transportation (OT) as a versatile component of the geometric data processing toolkit. Originally proposed for minimizing the cost of shipping products from producers to consumers, OT links probability and geometry using distributions to encode geometric features and developing metric machinery to quantify their relationships.

To transition OT from theory to practice, we show how to solve previously intractable OT problems efficiently on discretized domains and demonstrate a wide range of applications enabled by this new machinery. We illustrate the advantages and challenges of OT for geometric data processing by outlining my recent work in geometry processing, computer graphics, and machine learning. In each case, we consider optimization aspects of the OT problem for relevant geometric domains—including triangulated surfaces, graphs, and subsets of Euclidean space—and then show how the resulting machinery can be used to approach outstanding problems in surface correspondence, modeling, and semi-supervised learning.

“I will approach this question as one approaches a hippopotamus: stealthily and from the side.”

- R. Mahony

Acknowledgments

The preparation of this dissertation has been a community effort, and I am eternally grateful for the support of countless individuals and organizations throughout my near-decade at Stanford University. I attempt in [Sol15] to enumerate the many contributors to my development as a computer scientist, mathematician, and instructor. To supplement that comprehensive list, here I highlight some individuals whose guidance has strongly shaped my career thus far.

First and foremost, my family patiently has dealt with travel between two coasts and a three-hour time difference during my studies. Upon completion of this degree, I am excited to spend in-person time with my parents Rod and Nancy, sister Julia, brother-in-law Jeff, niece Caroline, and grandmother Juddy; I will always treasure my visits with my grandmother Dolores and wish I could share this milestone with her. My uncle Peter and aunt Dena continue to be enthusiastic supporters of my academic career, and I am excited to celebrate graduation with them.

At Stanford, Leonidas Guibas has been an encouraging academic advisor for my undergraduate, MS, and PhD degrees. His interdisciplinary insight into the world of geometric problem-solving led to the problems and techniques considered in this thesis. Additionally, starting from his time at Stanford and now from afar, Adrian Butscher continues to serve as a close academic mentor and has spent a heroic amount of time helping me understand the intricacies of differential geometry, PDE, and the academic job search.

My PhD research has been carried out with the support of the Hertz Foundation, the NDSEG fellowship, and the NSF Graduate Research Fellowship.

Relevant Publications

This dissertation incorporates material from several papers published over the last several years, in particular the following: [SNB$^+$12, SGB13, SRLB14, SRGB14, SdGP$^+$15].

These papers were developed and written jointly with many coauthors, who no doubt contributed text appearing in this dissertation. This work would not be possible without their patient collaboration.

Contents

Chapter 1

Introduction and Preliminaries

Algorithms for processing modeled shapes and algorithms for processing clouds of data points usually are treated as unrelated solutions to unrelated problems. Superficially, we can apply tactile and spatial intuition to processing shapes as surfaces, while noisy and abstract data may be more understandable through mathematical and statistical formality. Hence, three-dimensional shapes are treated in the geometry processing and vision literatures, while data analysis and pattern extraction are categorized as topics for machine learning and related disciplines.

Shape processing and high-dimensional data processing, however, use remarkably similar geometric language to state objectives and procedures. Three-dimensional objects and datasets both contain distinctive feature points connected by regions of varying curvature. Both encode *intrinsic* notions of proximity and distance along a domain rather than through the surrounding volume. And, both can be edited plausibly by stretching and bending motions.

Merging our understanding of these and other forms of input suggests the necessity and broad application of *geometric data analysis*, considered in the following two senses:

$$\underbrace{\textit{Geometric data}}_{\text{Modifier}}\ \underbrace{\textit{analysis:}}_{\text{Noun}}\quad \text{The analysis of geometric data}$$

$$\underbrace{\textit{Geometric}}_{\text{Modifier}}\ \underbrace{\textit{data analysis:}}_{\text{Noun}}\quad \text{Data analysis using geometric techniques}$$

With these related but distinct applications in mind, a natural research program is to develop machinery widening the scope of “geometric data analysis” from a specialized branch of statistics (see e.g. [Kir00, RR06]) to a broad field encapsulating the mathematical theory, algorithms, and computational applications of shape processing applied to abstract datasets and scans of physical objects alike.

1.1 Motivation

Geometric commonalities are emerging between many branches of computer science, indicating the timeliness of a unified approach. For example, Laplacians in graph theory [Chu96], semi-supervised learning [ZGL03], and computer graphics [SCV14] all by construction exhibit nearly identical properties inspired by classical geometric PDE. Kernel methods from statistics [HSS08] have analogous structure and properties to kernel embeddings used for matching scans of objects [SOG09]. Hamiltonian Monte Carlo sampling for probabilistic models [Nea11] requires the same geometric integrators as numerical simulations seeking to emulate qualitative properties of physical phenomena [SD06].

These examples and many others embody a fundamental shift in the geometric machinery developed for computer science applications. The earliest geometric challenges in computer science arguably were combinatorial, leading to the development of “computational geometry” as its own discipline. Algorithms for spatial subdivision, triangulation, topological sweep, and other tasks fundamental to computational geometry largely process collections of simple geometric primitives like points, line segments, and triangles. In this context, the challenge is to optimize time, space, and approximation complexity of operations organizing and processing this type of data.

While many challenging and application-critical problems remain in computational geometry, the proliferation of cameras and three-dimensional scanning devices for inputting low-dimensional shapes as well as the development of technology for gathering and exploring high-dimensional data have inspired a new context for geometry and computational shape processing. These and other datasets are fundamentally different from those considered in computational geometry, often times exhibiting redundancy, noise, and sampling issues limiting the applicability of exact computation.

Typical challenges in this context involve modeling rather than algorithm development. In particular, this new species of geometric computation draws inspiration not only from classical geometry but also from modern mathematical tools including differential geometry, geometric partial differential equations (PDE), and variational calculus. These sophisticated tools from continuous mathematics are well-suited to modeling shapes beyond Platonic solids, whose distinguishing features are characterized by nonconvexity, singularities, curvature, diffusion structure, distance metrics, embedding, and so on.

Development of computational tools accompanying these fine-grained continuous models yields a new class of research problems. Key themes in this discipline include:

- **Modeling:** Modern geometry and topology provide expansive languages for problems involving shape analysis in its various incarnations. Selecting among the countless pieces of machinery that may be relevant to a given computational problem is itself a formidable challenge. On the theoretical side, the subtleties of different geometric structures provide trade-offs between simplicity and expressiveness, between generality and applicability to a given problem, and so on. On the practical side, choosing models with an eye for the likelihood of accompanying stable, intuitive, and computationally feasible numerical methods requires intuition that can be at odds with purely theoretical consideration of modeling problems.

- **Discretization:** Deciding upon and analyzing a theoretical model is insufficient in the context of geometric data analysis. To bring the model to practical fruition, it must be discretized in a form amenable to storage and computational manipulation. This discretization is a critical decision fundamentally affecting the behavior of geometric algorithms. For example, in computer graphics, there exist many ways to store a two-dimensional surface embedded in three dimensions, e.g., as a triangle mesh, using a subdivision structure, or as a level set of an implicit function. Contexts in which these different discretizations are valuable contrast considerably; triangle meshes are well-suited to discretization of smoothing geometric flows via finite elements, subdivision leads to smooth surfaces that are attractive for rendering, and level set methods have shown considerable success for physical simulation.

- **Optimization:** Countless geometric problems can be posed using *variational* language, in which desirable features or quantities are derived as critical points of constrained or unconstrained optimization problems. After discretization, such variational problems form a class of difficult optimizations requiring specialized consideration. Natural problems in differential geometry easily can become numerically infeasible after applying what might appear to be reasonable discretizations. After

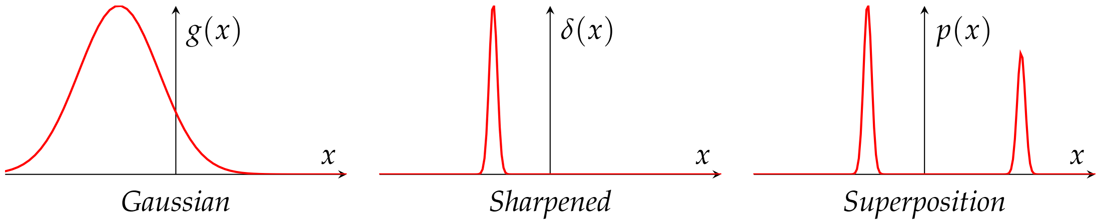


Figure 1.1: (left) A Gaussian distribution expresses the simplest geometric feature—a point—on the one-dimensional x axis; (middle) sharpening the distribution about its mean approaches a δ-function expressing the point with no uncertainty. Probabilistic language is fundamentally broader, however, supporting (right) superposition of multiple features.

all, an optimization constraint requiring a point to be on a fixed geometric domain like a surface likely will be nonconvex, assuming the domain itself is nonconvex. Constructive proofs from differential geometry may or may not suggest stable computational methods for these variational problems. For instance, the typical theoretical construction of harmonic maps between surfaces via heat flow presents fundamental challenges for discretization and convergence via finite elements (FEM), the typical discretization accompanying triangle mesh structures used in computer graphics [Bar05].

This long list of research challenges supports the development of *geometric modeling* as a discipline within computer science that is more clearly orthogonalized from the particular application area or data type. This viewpoint contrasts significantly with the current approach, in which discretizations and models from smooth geometry are adapted within individual research areas, leading to considerable redundancy from one field to the next.

1.2 Probabilistic Geometry

As an initial step toward a general toolbox for geometric data processing, this thesis develops algorithms for manipulating geometry through the use of probability distributions. This probabilistic framework will have many advantages over classical geometric language—which involves individual points or geometric features—by incorporating uncertainty, symmetry, and other phenomena endemic to noisy geometric measurements directly into the computational pipeline.

Figure 1.1 illustrates a simple example of using probabilistic language to express a geometric feature. In this case, we express features on a one-dimensional geometric domain, the real numbers $\mathbb{R}$. One of many ways to think of a Gaussian distribution $g(x)$ is as an expression of a point, the simplest geometric feature, centered at the mean; the standard deviation expresses the level of uncertainty in the location of the feature. Sharpening the Gaussian about its mean approaches a *delta function* $\delta(x)$, which encodes the classical point feature without any uncertainty. It is important to acknowledge, however, that the probabilistic language is fundamentally broader than classical geometric features; for example, two points features can be weighted and superposed into one distribution $p(x)$.

More generally, suppose $\Sigma \subseteq \mathbb{R}^n$ is a geometric domain like a surface or volume. We consider the space of probability measures $\mathrm{Prob}(\Sigma)$, where an element $\mu \in \mathrm{Prob}(\Sigma)$ is a measure taking subsets $U \subseteq \Sigma$ to $\mu(U) \in [0,1]$ with $\mu(\Sigma) = 1$. Example probabilistic expressions of geometric features on Σ include the following:

- A point $p \in \Sigma$ can be encoded as a δ-distribution satisfying

$$\delta_p(U) \overset{\text{def.}}{=} \begin{cases} 1 & \text{when } p \in U \\ 0 & \text{otherwise.} \end{cases}$$

- If Γ is a surface, curve, or other geometric feature embedded in a higher-dimensional space Σ, a distribution encoding Γ can be its intrinsic area measure, normalized so $\mu(\Gamma) = 1$.

- A set of k points $p_1, \ldots, p_k \in \Sigma$ can be encoded as a linear combination $\mu \overset{\text{def.}}{=} \sum_i \delta_{p_i}$.

Some, but not all, probability measures admit *density functions*. If $\mu \in \mathrm{Prob}(\Sigma)$ admits a density function $\rho : \Sigma \to \mathbb{R}^+$, then for all measurable $U \subseteq \Sigma$, we can write

$$\mu(U) = \int_U \rho(x)\, dx.$$

It can be valuable to only consider the subset of measures in $\mathrm{Prob}(\Sigma)$ admitting density functions. Some of our derivations will rely on a restriction to this space, although largely it is for convenience rather than necessity from a mathematical perspective. That is, some derivations will be written in terms of integrals to encourage intuition, although the corresponding statement may hold from a more general measure-theoretic perspective.

A source of challenges in geometric data processing stems from potential nonconvexity of the domain Σ. For example, any optimization with variables taking value on the unit sphere $\Sigma = S^2$ automatically has nonconvex constraints, as the exterior of a sphere is not

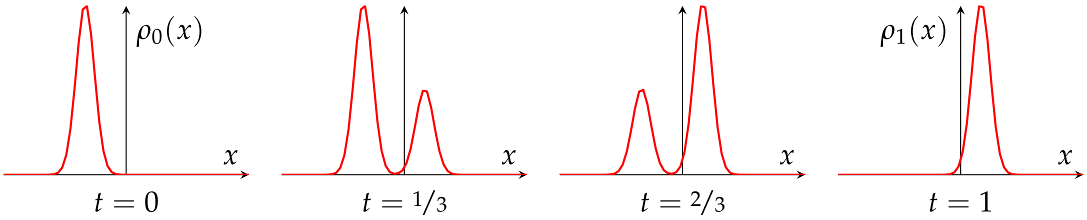


Figure 1.2: Trivial interpolation between two probability distributions $\rho_0(x)$ and $\rho_1(x)$ as $(1-t)\rho_0(x) + t\rho_1(x)$.

convex in $\mathbb{R}^3$. This geometric nonconvexity is a natural source of nonconvex optimization problems; if we are studying the geometry of an arbitrarily bent surface or region in $\mathbb{R}^n$, should expect this structure to manifest itself in any processing operation.

Dealing with distributions in $\text{Prob}(\Sigma)$ is one way around this issue. While Σ may not be a convex shape, $\text{Prob}(\Sigma)$ is a convex space, in the sense that $(1-t)\mu_0 + t\mu_1 \in \text{Prob}(\Sigma)$ for any $t \in [0,1]$ and $\mu_0, \mu_1 \in \text{Prob}(\Sigma)$. This difference brings about totally different discretizations of optimization problems that are more amenable to standard optimization tools. Whereas constraining a point p to be on Σ might be a specialized nonlinear constraint, if Σ is discretized into k pieces then typically elements $\mu \in \text{Prob}(\Sigma)$ roughly can be discretized as vectors $v \in [0,1]^k$ with $\mathbb{1}^\top v = 1$. The set of such vectors is the *probability simplex* and is defined using linear inequality constraints on vectors of probability values.

1.3 Geometric Properties of Distributions

We have argued that probability distributions provide an effective means for encoding geometric features. If we wish to pose geometric problems in terms of these distributions, however, we must be able to analyze and manipulate these features using tools tuned for the probabilistic context.

The algebra of probability distributions superficially resembles that of vectors in $\mathbb{R}^n$ but in reality does not provide a sufficient means for manipulating geometric features. For instance, suppose we are given two probability distributions $\rho_0(x)$ and $\rho_1(x)$ centered at two points $p_0, p_1 \in \mathbb{R}$. As a sample geometric operation, we might wish to recover points $(1-t)p_0 + tp_1$ along the segment from p_0 to p_1 using purely probabilistic operations. As illustrated in Figure 1.2, however, if we use the naïve approach of interpolating distributions algebraically as $(1-t)\rho_0(x) + t\rho_1(x)$, the result “teleports” mass from one peak to the

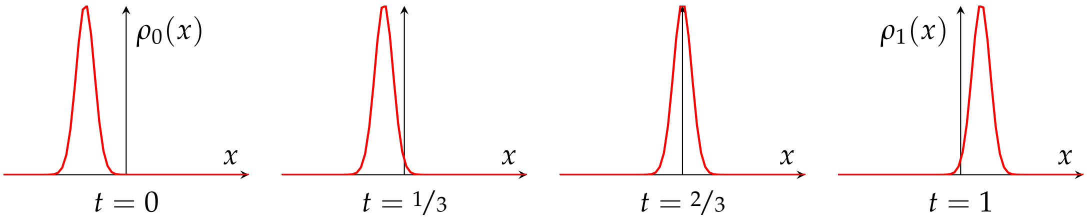


Figure 1.3: Geometric interpolation between two probability distributions $\rho_0(x)$ and $\rho_1(x)$.

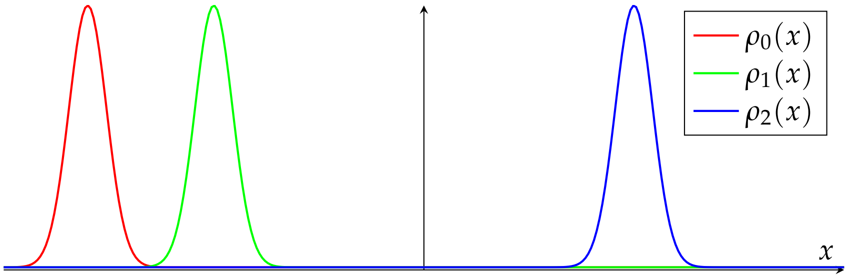


Figure 1.4: Three nearly equidistant distributions $\rho_0(x)$, $\rho_1(x)$, and $\rho_2(x)$ with respect to non-geometric measurements of probabilistic divergence.

other rather than shifting it horizontally. Instead, from a geometric standpoint, we might wish for interpolatory behavior more similar to that illustrated in Figure 1.3; although this can be accomplished by interpolating means of Gaussian distributions, when $\rho(x)$ is not described by a parametric model it is less clear how to recover such behavior.

This lack of geometric structure manifests itself in many measurements within the probabilistic pipeline. As the most fundamental example of a geometric measurement, consider the task of measuring *distance* between probability distributions in a way that respects the geometry of the underlying domain, e.g. the three distributions over $\mathbb{R}$ in Figure 1.4. The hope is to construct a measure of distance between distributions that roughly recovers distances between the peaks, at least in the case that the distributions are centered about individual points.

Two well-known candidate measures of probabilistic divergence are the L_1 distance and Kullback-Leibler (KL) divergence, written in one dimension as

$$\|\rho_0 - \rho_1\|_1 = \int_{-\infty}^{\infty} |\rho_0(x) - \rho_1(x)|\, dx$$

$$\text{and } \mathrm{KL}(\rho_0|\rho_1) = \int_{-\infty}^{\infty} \rho_0(x) \ln \frac{\rho_0(x)}{\rho_1(x)}\, dx.$$

These divergences essentially take a pointwise measure of divergence at each $x \in \mathbb{R}$, $|\rho_0(x) - \rho_1(x)|$ for the L_1 distance and $\rho_0(x) \ln \frac{\rho_0(x)}{\rho_1(x)}$ for KL, and integrate it over the domain. They have the advantage of being computable in closed-form and are accompanied with strong theoretical understanding within the contexts of Lebesgue theory and information theory, respectively.

These pointwise divergences, however, are ill-suited for geometric tasks. In particular, all three distributions ρ_0, ρ_1, and ρ_2 in Figure 1.4 are *nearly equidistant* with respect to these and related measures. Methods like kernelization [HSS08] can incorporate some local geometric information, but do so in a weak way that does not scale with straight-line distance as the means of these distributions are moved farther and farther apart.

1.4 Optimal Transportation

The non-geometric drawbacks of distributional divergences highlighted in §1.3 are well-known, at least in the theoretical community, and a few potential resolutions exist for this problem. Of primary importance is the theory of *optimal transportation* between probability distributions, which builds a theory of probability from a distance measure between distributions incorporating distances along a geometric domain. Here, we summarize a few key points of this theory relevant to our computational discussion in future chapters; we refer the reader to [Vil03] for more detailed treatment.

Suppose M is a manifold with or without boundary; our goal is to compute a distance between any two probability distributions over M. That is, we seek a positive definite, real-valued function $\mathrm{Prob}(M) \times \mathrm{Prob}(M) \to \mathbb{R}^+$ satisfying the triangle inequality. The primary distance function of interest in optimal transportation is the p-Wasserstein distance, computed as follows.

Given two probability distributions $\mu_0, \mu_1 \in \mathrm{Prob}(M)$, a "transportation plan" for transporting the mass distribution described by μ_0 to that described by μ_1 is a probability distribution π on the product space $M \times M$, where we interpret $\pi(U \times V)$ as the amount of mass to be displaced from U to V. To ensure that all the mass in μ_0 is transported to μ_1, we impose the constraints

$$\begin{aligned} \pi(U \times M) &= \mu_0(U) \quad \forall U \subseteq M \\ \pi(M \times V) &= \mu_1(V) \quad \forall V \subseteq M. \end{aligned} \tag{1.1}$$

Now, let $\Pi(\mu_0, \mu_1)$ denote the set of transportation plans satisfying these constraints, and let $d(\cdot, \cdot)$ be the geodesic distance function on M. Then, for $p \geq 1$, we define the p-Wasserstein distance as the optimal value

$$\mathcal{W}_p(\mu_0, \mu_1) \overset{\text{def.}}{=} \left[\inf_{\pi \in \Pi(\mu_0, \mu_1)} \iint_{M \times M} d(x,y)^p \, d\pi(x,y) \right]^{1/p}. \tag{1.2}$$

$\mathcal{W}_p$ can be interpreted as the *cost* of the optimal plan transporting the mass of μ_0 to that of μ_1, when moving mass from x to y costs $d(x,y)^p$. If we use $\mathbb{E}_\pi[\cdot]$ to denote expectations with respect to π, the p-th power of $\mathcal{W}_p(\mu_0, \mu_1)$ can be understood as minimizing an expectation with respect to π:

$$\mathcal{W}_p^p(\mu_0, \mu_1) \overset{\text{def.}}{=} \inf_{\pi \in \Pi(\mu_0, \mu_1)} \mathbb{E}_\pi[d(\cdot, \cdot)^p]. \tag{1.3}$$

When $d(\cdot, \cdot)$ is a distance, the p-Wasserstein distance satisfies all metric axioms and has several attractive theoretical properties—see [Vil03, §7] for details.

Discrete approximations of this distance that are suited for computational applications are difficult to obtain due to computational complexity. For instance, a discretization of $\pi \in \Pi(\mu_0, \mu_1)$ in (1.2) requires a quadratic number of optimization variables, since $\pi(x,y)$ is a function of two positions $x, y \in M$. Additionally, if we are to use general optimization machinery to compute $\mathcal{W}_p(\mu_0, \mu_1)$, we must be able to precompute or approximate $d(x,y)$ for any pair $x, y \in M$ before even *posing* the linear program. This scaling, which appears in many known discretizations of $\mathcal{W}_p$, is prohibitively expensive for large meshes.

Computational challenges aside, there are a few special cases of $\mathcal{W}_p$ of interest in the context of geometric data processing, highlighted below. These special cases will form the foundation for algorithms and applications we consider in future chapters.

1.4.1 Earth Mover's Distance

One special case of the p-Wasserstein distance occurs in the discrete case. Suppose $p, q \in [0,1]^n$ are histograms over n bins, satisfying $\mathbb{1}^\top p = \mathbb{1}^\top q = 1$. Furthermore, suppose for $i, j \in \{1, \ldots, n\}$, we are given a value $D_{ij} \in \mathbb{R}^+$ measuring the distance between bin i and bin j; this matrix will serve the role of the geodesic distance function $d(\cdot, \cdot)$ in (1.2). Then, the *earth mover's distance* (EMD) between p and q is defined as follows:

$$\mathrm{EMD}(p,q) \overset{\text{def.}}{=} \left\{ \begin{array}{rl} \min_T & \sum_{ij} D_{ij} T_{ij} \\ \text{s.t.} & T \geq 0 \\ & T\mathbb{1} = p \\ & T^\top \mathbb{1} = q. \end{array} \right. \tag{1.4}$$

The optimization variable T represents a discrete transportation plan, and the constraints are discrete analogs of the marginalization constraints in (1.1).

This distance, coined and introduced to the vision community in [RTG00], is commonly applied as a metric for comparing histogram-based image and shape descriptors, most popularly the SIFT descriptor of image keypoints [Low99]. The metric structure of p-Wasserstein distances in (1.2) depends critically on the distance structure of the underlying geodesic distance function $d(\cdot,\cdot)$, but the discrete formulation in (1.4) makes no assumption on the matrix D. A proof of the triangle inequality for $\mathrm{EMD}(\cdot,\cdot)$ when D is a metric matrix is given in [CA14] by adapting a proof from the continuous theory of optimal transportation.

1.4.2 One-Wasserstein Distance

The one-Wasserstein distance might be thought of as the most direct analog of the earth mover's distance from §1.4.1 to distributions over surfaces. Now, the distance matrix D is replaced a a pairwise geodesic distance function $d(\cdot,\cdot)$. The resulting problem (1.2) with $p = 1$ is measure-theoretical in nature, but is stated essentially using a continuous linear program and benefits from similar analysis.

Both discrete and continuous versions of the one-Wasserstein distance benefit from structure in the matrix of transportation costs D or $d(\cdot,\cdot)$. In particular, suppose $y \in M$ is on the shortest path from $x \in M$ to $z \in M$. Then, the cost of moving mass from x to z is identical to the cost of moving mass first from x to y and subsequently from y to z; this is a reflection of the fact that $d(x,z) = d(x,y) + d(y,z)$. We will leverage this property to formulate efficient methods for evaluating the one-Wasserstein distance in future chapters. A similar property does not hold for the two-Wasserstein distance, since

$$d(x,z)^2 = [d(x,y) + d(y,z)]^2 \geq d(x,y)^2 + d(y,z)^2.$$

We will use "one-Wasserstein distance" and "earth mover's distance" interchangeably in considering the case of distributions over continuous domains.

The one-Wasserstein distance can be computed in closed-form when $M = \mathbb{R}$, the real line. Suppose we are given a distribution $\rho(x)$. We define its *cumulative distribution function* $\mathrm{CDF}_\rho(x)$ as the integral

$$\mathrm{CDF}_\rho(x) \overset{\text{def.}}{=} \int_{-\infty}^{x} \rho(\bar{x})\, d\bar{x}.$$

Then,

$$\mathcal{W}_1(\rho_0, \rho_1) = \int_{-\infty}^{\infty} |\mathrm{CDF}_{\rho_0}(x) - \mathrm{CDF}_{\rho_1}(x)|\, dx. \tag{1.5}$$

Sadly, it is hard to define cumulative distance functions in greater than one dimension, and this formula is not easily extended to other domains.

1.4.3 Two-Wasserstein Distance

Many computational applications of optimal transportation focus on the one-Wasserstein distance, due to its introduction to the vision community in [RTG00] and intuitive link to discrete minimum-cost flow problems. This situation contrasts with the *theory* of optimal transportation, which focuses on the $p > 1$ and in particular $p = 2$ cases.

The two-Wasserstein distance arguably is the best understood transportation distance from a theoretical perspective:

$$\mathcal{W}_2^2(\mu_0, \mu_1) \overset{\text{def.}}{=} \inf_{\pi \in \Pi(\mu_0, \mu_1)} \iint_{M \times M} d(x,y)^2\, d\pi(x,y).$$

This distance inherits favorable properties from the Hilbert space structure of the L_2 inner product. One way to see a potential connection is to define a π-weighted inner product over $M \times M$ as

$$\langle f, g \rangle_\pi \overset{\text{def.}}{=} \iint_{M \times M} f(x,y) g(x,y)\, d\pi(x,y).$$

A similar product can be defined when f, g are replaced by vector fields when M is a surface. Then, the objective for computing $\mathcal{W}_2$ can be written $\langle d, d \rangle_\pi$. As an example mathematical application of this added structure, [JKO98] and subsequent papers use it to pose PDEs like the Fokker-Planck equation as gradient flows in the Wasserstein metric, leading to stable numerical integrators for this challenging problem.

Similarly to (1.5), the two-Wasserstein distance between one-dimensional distributions

$\rho_0(x)$ and $\rho_1(x)$ can be calculated in closed form:

$$\mathcal{W}_2^2(\rho_0, \rho_1) = \int_0^1 (\mathrm{CDF}_{\rho_0}^{-1}(x) - \mathrm{CDF}_{\rho_1}^{-1}(x))^2 \, dx.$$

These formulas make the connection to functional analysis more concrete: The two-Wasserstein distance corresponds to an L_2 norm on inverse CDF space, while the one-Wasserstein distance corresponds to an L_1 norm on non-inverted CDFs. Regretfully, this connection is obscured in higher dimensions and on non-Euclidean domains.

1.5 Optimization over Wasserstein Distance

Regardless of the shape of M and choice of $p \in \mathbb{R}$, the optimization in (1.2) for p-Wasserstein distances is essentially a linear program that can be minimized using convex machinery. This observation provides potential for relaxation of otherwise non-convex and difficult-to-discretize problems in classical geometry.

As a simple example, suppose we are given m points $x_1, \ldots, x_m \in M$ and wish to find their *barycenter* with respect to the shortest-path distance $d(\cdot, \cdot)$ on M to the p-th power. A variational approach to finding such a point might be to solve the following minimization:

$$\min_{x \in M} \sum_{k=1}^{m} d(x, x_k)^p.$$

This problem is non-convex whenever M is non-convex. Instead, however, we could solve the following *convex* problem in Wasserstein distances:

$$\min_{\rho(x) \in \mathrm{Prob}(M)} \sum_{k=1}^{m} \mathcal{W}_p^p(\rho, \delta_{x_k}).$$

This minimization finds the barycenter of a set of δ-functions centered at the x_k's, treated as distributions rather than points.

The Wasserstein barycenter with δ-function constraints can be considered a relaxation of the original barycenter problem. In particular, if $\rho(x)$ is constrained to be a δ-function, then the point it encodes necessarily will be the desired barycenter. When this δ constraint is relaxed, however, the optimal $\rho(x)$ may not be a δ-function; in this case, it is less clear that we can read off the optimal solution to the original problem. Unlike analogous problems constructed using information-theoretic notions of probabilistic divergence, e.g. minimizing a sum of KL divergences rather than transportation distances, we prove in Chapter 5 that there exist reasonable conditions under which the relaxation is tight, that is, under which the optimal $\rho(x)$ is indeed a δ-function. Furthermore, empirical evidence indicates that extracting peaks of ρ when it is not a δ-function is a reasonable heuristic for the general barycenter problem.

This example illustrates a larger theme in the application of optimal transportation to geometric data processing. The original geometric barycenter problem has relatively few variables but is non-convex by construction. Contrastingly, the problem can be relaxed to a larger-scale optimization, in this case over distributions rather than points, with the potential of optimality in restricted cases.

There are many ways to derive this relaxation from geometric features to distributions. For instance, in the context of graphical models, a similar optimization appears as the relaxed version of an *a posteriori* problem with pairwise potentials proportional to $d(x,y)^p$. Precise characterization of when these relaxations are expected to succeed largely remains an open problem, with increased relevance given recent developments in numerical optimization over Wasserstein space.

1.6 Related Work

Having established the basic language and challenges of optimal transportation, we now briefly highlight existing computational techniques for transportation drawn from different research fields. In future chapters, we will introduce additional background material pertaining to the particular discussion at hand.

The original formulation of optimal transportation, introduced in [Kan42], involves a linear program connecting a pair of distributions. The cost of moving density from one point to another is specified using a fixed dense matrix of pairwise costs. As outlined in [BDM09], a variety of linear program solvers and dedicated combinatorial schemes have been devised for this problem. These methods scale up to a few thousand variables and were applied to graphics applications in [BvdPPH11] and in [LD11]. They do not scale to large domains such as images with millions of pixels, however, and are not tailored for advanced problems like barycenter computation. Assorted approximations of EMD truncate large distances [PW09] or are specialized to discrete domains like graphs [Tak10].

Specific instances of optimal transportation can be efficiently solved by leveraging tools from computational geometry. The transportation cost from continuous to pointwise measures, for instance, can be computed either via multiscale algorithms [Mér11, STTP14] or through Newton iterations on Euclidean spaces [dGBOD12, ZSG$^+$13]. More recently, this Newton-based approach for optimal transportation was extended to discrete surfaces [dGMMD14]. Transportation distances between point clouds and line segments also were approximated in 2D based on a triangulation tiling of the plane and greedy point-to-segment clustering [dGCSAD11]. Another line of work proposes a dynamical formulation for optimal transportation with an additional time variable. Most prominently, for squared distance costs, Benamou and Brenier [BB00a] compute transportation distances by minimizing the cost of advecting one distribution to another in time; Chapter (2) presents a related technique for one-Wasserstein distances.

In graphics and vision, EMD and its optimal transportation counterparts have been applied to a variety of problems. EMD was first introduced to the vision community in [RTG00] and since has been used to compare histograms and other descriptors. More recently, [BvdPPH11] applies approximations of optimal transportation to interpolate between BRDFs, intensity histograms, and other simple distributions; similar problems are considered in [BRPP14] after defining the barycenter of a set of distributions with respect to approximated transportation distances. [dGCSAD11, dGBOD12] compute transportation distances from two-dimensional point sets for application in shape processing and blue noise generation, while [MMdGD11] employs a similar formulation to triangulation problems. These distances also have been applied to geometry analysis [LD11, LPD11], spherical parameterization [DT10], and matching [Mém11, SNB$^+$12]. None of these approaches, however, is able to compute EMDs or related distributional distances intrinsic to meshed geometry without aggressive approximation or restriction to a simpler domain.

1.7 Overview

This dissertation explores the application of optimal transportation, most prominently evaluation of and optimization over Wasserstein distances, to solve practical problems in geometric data processing. We will focus on a representative applications drawn from relevant research areas:

- *Computer graphics:* Evaluation of distances between points and features on geometric domains comprises a basic operation in computer graphics systems for modeling shapes and scenes as well as navigating collections of modeled data. Furthermore, interpolation between signals over images and shapes is a key problem in computer graphics sharing structure with the "displacement interpolation" structure of Wasserstein space.

- *Surface correspondence:* Optimizing for a map from one surface into another is a key step in pipelines for geometry processing and medical imaging. Relaxing point-to-point maps to a probability-valued space reveals tractable, convex problems for mapping that incorporate uncertainty and symmetry into the fundamental representation of correspondences.

- *Semi-supervised learning:* Many problems in semi-supervised learning can be posed as propagation of distribution-valued data over a graph domain; modeling in this space naturally leads to optimization problems involving Wasserstein distances. Furthermore, through constructions similar to the basic example in §1.5, the same machinery suggests a general pipeline for manifold-valued learning.

In each case, we will consider the fundamental challenges of evaluating and manipulating geometric features as probability distributions over the relevant domain. Once we establish stable computational tools, we will sample the applications of this machinery as part of larger systems for understanding signals on geometric domains.

We will begin by considering algorithms for evaluating $\mathcal{W}_1$ (Chapter 2) and $\mathcal{W}_2$ (Chapter 3) as well as derived quantities. These chapters mainly focus on the computational challenges of transportation problems. Applications discussed in these chapters are relatively direct extensions of the algorithms under consideration. In the remaining chapters, we show how to apply Wasserstein distances to models for more complex computational problems. Chapter 4 uses these distances to develop theoretical models for mapping between surfaces, yielding a flexible probabilistic framework that incorporates uncertainty and symmetry into the registration pipeline. Chapter 5 uses optimal transportation to approach a problem in semi-supervised machine learning for predicting histogram-valued labels on graphs. Finally, Chapter 6 concludes with a broader discussion of future challenges in applied optimal transportation and processing of geometric data.

Chapter 2

Earth Mover's Distance on Triangulated Surfaces

A common task in geometry processing is the computation of various classes of distances between points on or inside a discrete surface. For example, many shape matching algorithms need clues about the relative positioning and orientations of features to be matched, which can be obtained from pairwise feature distances. It is desirable for such distances to be true metrics, intrinsic, globally shape-aware, smooth, and insensitive to noise and topology changes, without inducing considerable distortion on the underlying metric. In particular, the level sets of the distance function should be evenly spaced, in a visual sense, along the surface.

Early approaches to defining and computing intrinsic distances do not satisfy all of these requirements. Despite their central place in classical differential geometry, geodesic distances have many shortcomings for computational applications, such as being sensitive to noise and topology and not being globally shape-aware, that is, not conforming to geometric features of the surface as distances increase [LRF10]. Spectral distances [CLL$^+$05, FPRS07, LRF10] overcome these shortcomings but can be unintuitive with unevenly-spaced isocontours. The drawbacks of geodesic and spectral distances indicate that a hybrid approach is needed. The approximation of geodesic distance formulated in [CWW13] is an important step in this direction. While these approximations are smoothed versions of the geodesic distance that are robust and globally shape-aware, they may not be symmetric or satisfy the triangle inequality.

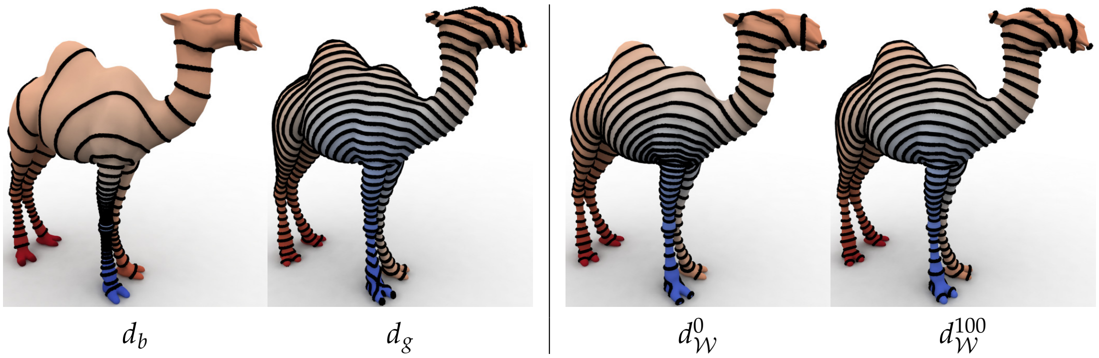


Figure 2.1: (left) Biharmonic (d_b) and geodesic distances (d_g) from the foot of a camel model; (right) distances computed using our approach with 0 ($d^0_{\mathcal{W}}$) and 100 ($d^{100}_{\mathcal{W}}$) spectral terms. Unlike d_b, even our most aggressive spectral approximation $d^0_{\mathcal{W}}$ has smooth, isotropic, and evenly-spaced level sets, while adding spectral terms makes our distance converge to d_g. We visualize distances from a single source vertex to all others by color, with blue indicating small distance and red indicating large distance. We also include isocontours at equally-spaced intervals, shown in black.

In this chapter, we introduce a novel hybrid approach for computing a variety of surface distances that have all of the desired properties. The key idea is to consider the more general problem of computing distances between probability distributions supported on the mesh. Hence, our goal is to compute the earth mover's distance, or one-Wasserstein distance, between distributions represented as functions on mesh vertices. Once we have a means to compute the EMD between general distributions, we then consider various specializations including computation of a class of surface distances generalizing the geodesic distance.

The computation of the EMD along a surface could be performed using a brute force linear programming approach. This approach, however, not only is computationally infeasible on reasonably-sized meshes but also leads to a "chicken-and-egg problem," since such a formulation requires precomputing all pairwise geodesic distances between mesh vertices. Therefore, one of our contributions is to make use of an alternative *differential* formulation of EMD that can be discretized using finite elements (FEM). Furthermore, a spectral expansion of the optimization variables reveals successive approximations to EMD that are fast to compute and are themselves distance metrics.

Our approach has several benefits. First, our family of distances is general and provides a principled and efficient way to compute distances between probability distributions supported on various types of features. Second, if we consider δ-distributions centered on surface points, we obtain a family of distances that ranges from the geodesic distance (since the EMD between two δ-distributions reduces to geodesic distance) to a novel spectral distance. The latter is perceptual convincing (Figure 2.1) despite having a simple formulation in terms of the Green's function of the Laplace-Beltrami operator. Third, inspired by [RLF09], we develop a means of extending our distances from the surfaces to the surrounding volume, obtaining a pointwise distance metric on $\mathbb{R}^3$ that reduces to geodesic distance when the selected points are on the surface. Finally, our machinery enables a number of applications to problems in path planning, surface analysis, and other fields.

2.1 Related Work

Existing approaches to defining and computing intrinsic distances broadly can be categorized as "primal" or "dual." A *primal* approach operates on the surface mesh directly to obtain exact or approximate distances. For example, [MMP87, SSK$^+$05] apply such an approach for finding exact geodesic distances (i.e. lengths of shortest paths constrained to lie on the mesh), while [KS98, CHK13] approximate geodesic distances to achieve faster run times. As discussed in [LRF10], however, despite their connection to classical differential geometry, geodesics have a number of shortcomings for computational applications, such as being sensitive to noise and topology and not being globally shape-aware.

These shortcomings inspired the development of *dual* methods for distance computation. Dual distances lift the problem to an alternative space, such as the set of real-valued functions on the mesh, where relationships between function values are used as proxies for inferring distances on the underlying domain. The most popular dual distances are spectral distances, such as the diffusion [CLL$^+$05], commute time [FPRS07], and biharmonic [LRF10] distances. These distances can be unintuitive, however, with isocontours that are unevenly-spaced along the surface. This artifact is a fundamental problem, because dual approaches achieve global shape-awareness and robustness by averaging over many paths whose structure can depend on the curvature and local diameter of the surface.

The drawbacks of completely primal or dual methods indicate that a hybrid approach integrating properties of both approaches may be called for. In [CWW13] an approximation of geodesic distance was formulated by integrating the normalized gradient field of the heat kernel. This approximation gives smoothed versions of the geodesic distance parameterized by the time parameter of the heat kernel. While robust and globally shape-aware, these are not guaranteed to be true distance metrics. Another hybrid distance was proposed in [PBDSH13], where geodesics between sampled vertices are embedded in Euclidean space using multi-dimensional scaling (MDS). The embedding is interpolated to the entire mesh by solving a biharmonic equation, and Euclidean distances in the embedding space provide a distance measure on the entire mesh. Since it is generally impossible to embed geodesic distances exactly into Euclidean space, this approach is likely to give inconsistent results when run repeatedly.

2.2 Distance Computation

2.2.1 One-Wasserstein Distances from Flows

A remarkable observation from the theory of optimal transportation provides a differential strategy for evaluating $\mathcal{W}_1$ under suitable regularity. Suppose M is a compact surface and $\mu_0, \mu_1 \in \operatorname{Prob}(M)$ admit densities $\rho_0, \rho_1 : M \to \mathbb{R}^+$. Then, the optimal value of the following convex optimization yields the EMD between μ_0 and μ_1:

$$\mathcal{W}_1(\mu_0, \mu_1) = \begin{cases} \inf_J \int_M \|J(x)\| \, dx \\ \text{s.t. } \nabla \cdot J(x) = \rho_1(x) - \rho_0(x) \\ \quad J(x) \cdot n(x) = 0 \ \forall x \in \partial M. \end{cases} \tag{2.1}$$

This optimization computes a vector field J on M whose boundary ∂M has normal $n(x)$ using a convex energy with linear constraints. In the language of fluid dynamics, it can be thought of as an Eulerian alternative to the Lagrangian formulation (1.2) when $p = 1$, i.e. points $x \in M$ watch probabilistic mass move past along flow lines of $J(x)$. Indeed, (2.1) first arose as the "Beckmann problem" in network flow [Bec52]. See [San13] for further analysis of this problem and its connection to optimal transportation, and see [Vil03, §1.2.3], [FM02], and [San09] for a broader discussion.

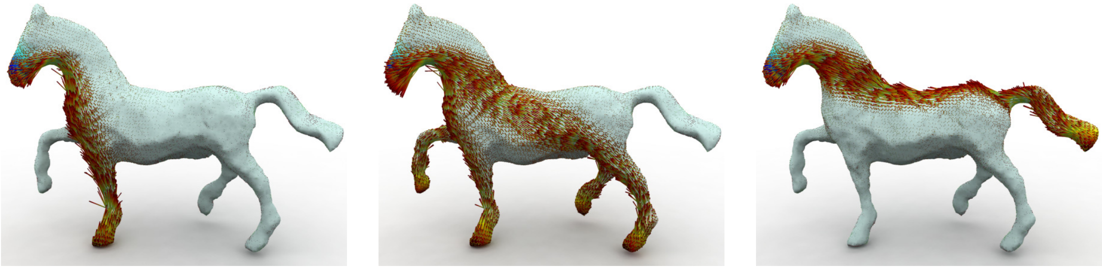


Figure 2.2: Vector fields J transporting mass to a distribution concentrated on the nose of a horse from a distribution on (left) one of its hooves, (middle) all four of its hooves, and (right) its tail.

In this chapter, we use (2.1) as a starting point for the computation of a discrete approximation of the 1-Wasserstein distance between two probability distributions on M. With only one unknown $J(x)$ per point $x \in M$, this optimization scales linearly with the size of the mesh rather than the quadratic scaling of (1.2).

2.2.2 Properties of the 1-Wasserstein Distance

Given its definition via geodesic distances in (1.2), it comes as no surprise that the 1-Wasserstein distance $\mathcal{W}_1$ is intricately linked with the metric structure of M even if it is computed using differential techniques. Here we state some properties relevant to our target applications in geometry processing and graphics.

The Beckmann problem recasts the transportation problem (1.2) in Eulerian language as finding the direction of *steady-state flow* of mass from ρ_0 to ρ_1. The vector field J can be thought of the velocity of this flow; examples of J computed using the discrete method in §2.3 are shown in Figure 2.2. Since an optimal flow moves mass as efficiently as possible, the following proposition is intuitively clear and follows from the first-order optimality conditions for (2.1):

Proposition 1. *Let J solve the optimization* (2.1) *with given densities ρ_0, ρ_1. Then, flow lines of J are geodesics on M.*

Proof. If we let $\lambda : M \to \mathbb{R}$ be a Lagrange multiplier function for the divergence constraint, then the Lagrangian of (2.1) is given by

$$\mathcal{L}(J,\lambda) = \int_M \left[\|J(x)\| + \lambda(x)(\nabla \cdot J(x) + \rho_0(x) - \rho_1(x))\right] dx.$$

Let J be a critical point of the variational problem (2.1) and introduce a variation vector field δJ. Assume $\delta J\big|_{\partial M} \equiv 0$ so that the Neumann boundary condition is maintained. Also assume $\delta J(x) = 0$ whenever $J(x) = 0$. Taking the variation of the Lagrangian in the δJ direction yields

$$0 = \frac{d}{d\varepsilon}\mathcal{L}(J + \varepsilon\,\delta J, \lambda)\bigg|_{\varepsilon=0} = \int_M \delta J(x)\left[\frac{J(x)}{\|J(x)\|} - \nabla\lambda(x)\right] dx.$$

Since this holds for all δJ, we have shown $\nabla\lambda(x) = {}^{J(x)}/_{\|J(x)\|}$ whenever $J(x) \neq 0$. This shows that $\|\nabla\lambda(x)\| = 1$, and thus by the eikonal equation λ is a geodesic distance function [Arn03]; hence flows of $J(x)$ are either constant or geodesics. □

A feature of the 1-Wasserstein distance distinguishing it from p-Wasserstein distances with $p > 1$ is that its optimal transportation plans are not unique [Vil03, §2.4.6]. That is, it is known that mass is transported along geodesics, but not how far a particle of mass travels along any given geodesic. One optimal plan obtained from $J(x)$ is the Dacorogna-Moser construction [Vil03, Chapter 5]. In this construction, we let J solve (2.1) and define $\rho_t \stackrel{\text{def.}}{=} (1-t)\rho_0 + t\rho_1$. Now we consider the flow $z : [0,1] \times M \to M$ of the ordinary differential equation (ODE):

$$\dot{z} = -\frac{J(z)}{\rho_t(z)}\,. \tag{2.2}$$

So, z produces a curve $t \mapsto z(t, x_0) \in M$ satisfying the ODE with initial conditions $z(0, x_0) = x_0$. Moreover, since the velocity of this curve is proportional to J, it is also a geodesic. The Dacorogna-Moser plan takes the mass at each $x_0 \in M$ and moves it to $z(1, x_0)$.

Next, by definition the Eulerian velocity of the Dacorogna-Moser flow is $V_t(x) = -J(x)/\rho_t(x)$. Consequently, the transport equation $\frac{\partial\rho_t}{\partial t} + \nabla\cdot(\rho_t V_t) = 0$ satisfied by (ρ_t, V_t) shows $\nabla\cdot J = \rho_1 - \rho_0$. This tells us that if ρ_0 is advected under the flow z, the resulting time-dependent family of densities is ρ_t.

Finally, the following result shows that one can recover pointwise geodesic distances from the 1-Wasserstein distance in the special case of infinitely sharply peaked δ-distributions.

Proposition 2. *Let δ_p be a delta distribution centered at $p \in M$, and let χ_Q be the uniform distribution supported on a subset $Q \subseteq M$. Let J be the solution of the optimization problem* (2.1). *Then,*

1. *The flow lines of J are geodesics from p to all points of $\mathcal{Q}$.*
2. *As $\mathcal{Q} \to \{q\}$ and $\chi_{\mathcal{Q}} \to \delta_q$ appropriately, then $\mathcal{W}(\delta_p, \chi_{\mathcal{Q}})$ converges to the geodesic distance between p and q.*

Proof. The first part is a consequence of Proposition 1 while the second part follows from the weak convergence properties of Wasserstein distances and the fact that Wasserstein distances between delta-distributions always reduce to geodesic distances since there is only one transport plan in $\Pi(\delta_p, \delta_q)$, namely the plan that assigns all the mass at p to q. □

2.3 Simplification and Discretization

The formulation of EMD in (2.1) is amenable to discretization on triangle meshes, commonly encountered in graphics and geometry processing. In this section we propose a discretization using finite elements (FEM) that admits straightforward optimization.

2.3.1 Vector Field Decomposition

The *Helmholtz-Hodge* decomposition shows that any vector field J on M can be written as [Sch95, PP03]

$$J(x) = \nabla f(x) + \mathcal{R} \cdot \nabla g(x) + h(x),$$

where $\mathcal{R}$ is the linear operator that rotates a vector 90° clockwise in the tangent plane. The "gradient part" of J is the vector field ∇f, the "curl part" of J is the vector field $\mathcal{R} \cdot \nabla g$, and the "harmonic part" of J is the vector field h satisfying $\nabla \cdot h(x) = 0$ and $\nabla \times h(x) = 0$. In case $\partial M \neq \varnothing$, we must additionally impose Neumann boundary conditions on J. This boundary condition reflects the fact that shortest-path curves cannot leave the surface, so when they reach they boundary they must become tangential.

Substituting this decomposition into (2.1) and using the fact that $\nabla \cdot (\mathcal{R} \cdot \nabla g) = \nabla \cdot h =$

0 yields the equivalent optimization

$$\mathcal{W}_1(\mu_0,\mu_1) = \begin{cases} \inf\limits_{f,g,h} \displaystyle\int_M \|\nabla f(x) + \mathcal{R}\cdot\nabla g(x) + h(x)\|\, dx \\ \\ \text{s.t. } \Delta f(x) = \rho_1(x) - \rho_0(x) \\ \qquad g(x) = 0 \text{ and } {}^{\partial f(x)}/_{\partial n} = 0\ \forall x \in \partial M \\ \qquad \nabla\cdot h(x) = 0 \text{ and } \nabla\times h(x) = 0\,. \end{cases} \tag{2.3}$$

This form shows that f is determined independently of the optimization by solving $\Delta f = \rho_1 - \rho_0$ with Neumann boundary conditions; this equation has a solution since ρ_0 and ρ_1 integrate to 1. We therefore compute $\mathcal{W}_1$ in two steps, one for finding f and one for finding g and h; moreover, eliminating f from the optimization leaves an essentially unconstrained optimization for g and h.

2.3.2 Spectral Reduction

We further simplify (2.3) using the spectral decomposition of the Laplacian Δ. First, we obtain a basis for functions on M by solving the eigenvalue problem $\Delta\phi_i = \lambda_i\phi_i$ with Dirichlet boundary conditions $\phi_i|_{\partial M} = 0$. The gradients and rotated gradients of these functions comprise a basis for gradient and curl fields of M. Additionally, the set of harmonic vector fields on M admits a basis with dimension equal to two times the genus of M [TACSD06].

Denoting a combined basis for curl fields and for harmonic fields as $\psi_1, \psi_2, \psi_3, \ldots$, we therefore can write the unknown vector field as $\mathcal{R}\cdot\nabla g + h = \sum_i c_i\psi_i$ where c_i are unknown coefficients. After solving $\Delta f = \rho_1 - \rho_0$ to precompute the vector field $v = \nabla f$, (2.3) can be recast as the unconstrained optimization:

$$\mathcal{W}_1(\mu_0,\mu_1) = \inf_{\{c_i\}} \int_M \left\| v(x) + \sum_i c_i\psi_i(x) \right\| dx\,. \tag{2.4}$$

This objective is convex in the unknowns c_i. Boundary conditions are not needed because they have been incorporated into the ψ_i's.

2.3.3 Discretization via Finite Elements (FEM)

Using triangle mesh geometry to discretize M, we express scalar functions $f : M \to \mathbb{R}$ with one value per vertex interpolated to faces using piecewise linear "hat" functions. Vector fields are piecewise constant per face, allowing for a gradient operator ∇ taking functions on the vertices to vector fields on the faces.

We solve the Poisson equation $\Delta f = \rho_1 - \rho_0$ with Neumann boundary conditions for f using a first-order finite elements approach as in [Say08]. This sparse linear solve can be carried out at interactive rates without the need for spectral approximation. Then, we compute the curl and harmonic components of J. For the curl vector fields, we examine two options for choosing a basis as above, trading off between speed and quality. The most accurate solutions are obtained simply by writing g with one value per vertex. Alternatively, we can improve timings with some cost in accuracy by writing g in a truncated basis of low-frequency eigenvectors of the Laplacian matrix Δ. We use a method like [TACSD06] to compute a basis for harmonic vector fields.

In our discretization, (2.4) becomes the following optimization:

$$\inf_{\{c_i\}} \sum_{t \in T} a_t \left\| v_t + \sum_i c_i \psi_{it} \right\|, \tag{2.5}$$

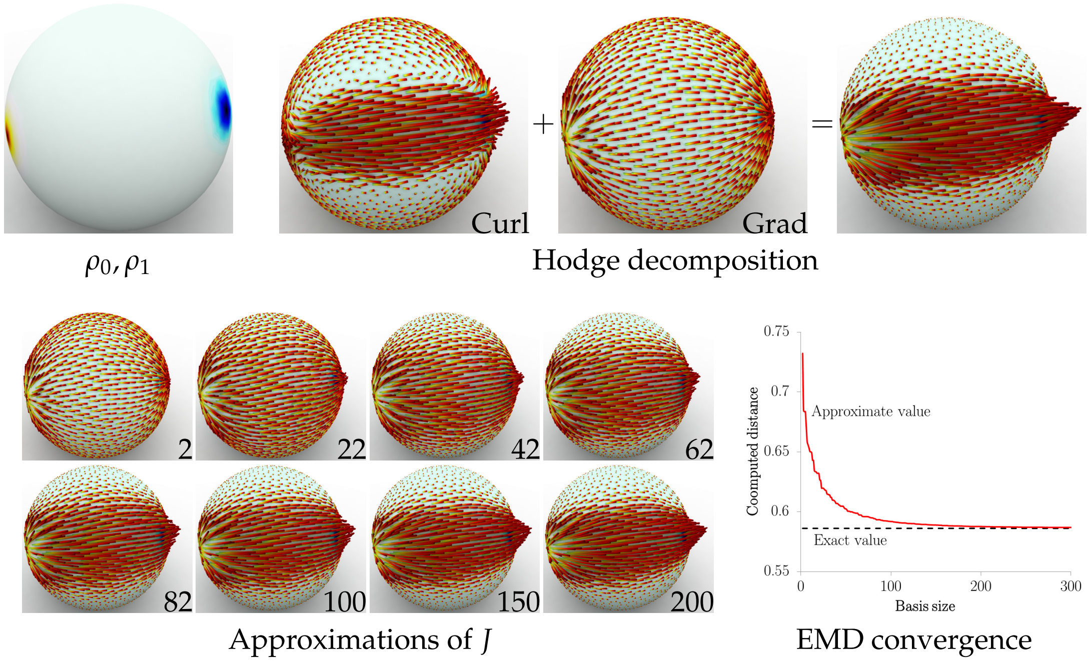
where T is the set of triangles in M, each triangle $t \in T$ has area a_t, ψ_{it} is the value of the basis element ψ_i on triangle t, and v_t is the gradient of the piecewise-linear f defined above.

If we use a truncated eigenbasis for g, we are only approximating the distance $\mathcal{W}_1$. Effectively this constrains certain coefficients c_i of the expansion in (2.4) to zero, and thus these approximations *overestimate* $\mathcal{W}_1$. Figure 2.3 illustrates convergence as the number of eigenfunctions is increased; even a small spectral basis provides a strong approximation of $\mathcal{W}_1$.

2.3.4 Properties of the Discretization

Since we discretized (2.4) we can expect its properties to hold approximately for the discretization. We can, however, prove that one important property of the discretization holds exactly, even with spectral approximation:

Proposition 3. *Minima of* (2.5) *satisfy the triangle inequality for discrete probability distributions*



ρ_0, ρ_1

Hodge decomposition

Approximations of J

EMD convergence

Figure 2.3: (top left) Two distributions ρ_0, ρ_1 on a sphere colored yellow and blue; (top right) the Hodge decomposition of the vector field J taking ρ_0 to ρ_1; (lower left) approximations of J with more and more curl basis functions (basis size on lower right of each sphere); (lower right) EMD between ρ_0 and ρ_1 as a function of basis size.

represented using one value per vertex on M, even if the bases for curl and harmonic vector fields are truncated.

Proof. One can show directly that the quantity in the right hand side of (2.1) satisfies the properties of a distance without appealing to the equivalence of (2.1) with the 1-Wasserstein distance. That is, the right hand side of (2.1) is symmetric and non-negative, vanishing only if $J = 0$ or $\rho_0 = \rho_1$. Moreover, the triangle inequality holds by the linearity of the divergence operator (so if J_{ij} satisfies $\nabla \cdot J_{ij} = \rho_i - \rho_j$ then $\nabla \cdot (J_{12} + J_{23}) = \rho_1 - \rho_2 + \rho_2 - \rho_3 = \rho_1 - \rho_3$) and the fact that $\|\cdot\|$ itself satisfies the triangle inequality. The same considerations now guarantee that the discrete approximations of (2.1) are also distance functions. □

Thus, our approximations of $\mathcal{W}_1$ are in fact distances in themselves.

```
**function** WEISZFELD-WASSERSTEIN($\rho_0, \rho_1$)
    ▷ $\rho_0, \rho_1$ have one value per vertex
    ▷ Concatenate $B_t$'s vertically to obtain $B$

    $f \leftarrow \Delta^+(\rho_1 - \rho_0)$                                   ▷ Solve for gradient part
    $v \leftarrow \nabla f$                                                      ▷ Compute gradient vector field

    $c_t \leftarrow 0\ \forall t \in T$                                          ▷ Initialize vector field to zero
    **for** $i \leftarrow 1, 2, 3, \ldots$                                       ▷ Iterate until convergence
        $r_t \leftarrow B_t c + w_t\ \forall t \in T$                            ▷ Compute residuals
        $R \leftarrow [r_1; r_2; \cdots; r_{|T|}]$                               ▷ Concatenate residuals

        $D \leftarrow \mathrm{diag3}(\|r_1\|^{-\frac{1}{2}} \ldots \|r_{|T|}\|^{-\frac{1}{2}})$   ▷ Diagonal of inverse residual roots repeated 3×

        $c \leftarrow c - (DB)^+ DR$                                             ▷ Least-squares for next iterate

    **return** $J_t = v_t + B_t c\ \ \forall t \in T$
```

Figure 2.4: Weiszfeld algorithm for optimizing $\mathcal{W}_1$, using the steps outlined in [Li98]; in challenging test cases the least-squares solve can be regularized slightly for numerical stability. A^+R denotes the least-squares solution X to the system $AX \approx R$.

2.4 Optimization

We derive an algebraic form for (2.5) by assembling the coefficients c_i into a vector c and the vectors ψ_{it} for a given triangle t into the columns of a matrix $A_t \in \mathbb{R}^{3\times k}$. After defining $w_t \overset{\text{def.}}{=} a_t v_t$ and $B_t \overset{\text{def.}}{=} a_t A_t$, (2.5) becomes the minimization problem

$$\inf_{\{c\}} \sum_t \|B_t c + w_t\| .$$

In this form, our optimization problem attempts to minimize a sum of Euclidean norms. This classical problem, known as the "geometric median" or "continuous location" problem, appears in the optimization literature and can be solved using a variety of techniques. The most well-known classical approach is Weiszfeld's algorithm, originally proposed in [Wei37], an iteratively-reweighted least-squares technique with convergence guarantees [Pla11]. Figure 2.4 states the algorithm adapted to our problem.

In Figure 2.5, we also provide a lightweight optimization method based on the alternating direction method of multipliers (ADMM) [BPC+11]; this new approach derived below suffers from fewer conditioning problems and solves an identical linear system in each iteration, allowing it to be pre-factored for *all* EMD computations on a surface.

```
function ADMM-WASSERSTEIN($\rho_0, \rho_1$)
    ▷ $\rho_0, \rho_1$ have one value per vertex
    ▷ Concatenate $B_t$'s vertically to obtain $B$

    $f \leftarrow \Delta^+(\rho_1 - \rho_0)$                                   ▷ Solve for gradient part
    $v \leftarrow \nabla f$                                                     ▷ Compute gradient vector field

    for $i \leftarrow 1, 2, 3, \ldots$                                          ▷ Iterate until convergence
        $z_t \leftarrow B_t c + w_t - \frac{y_t}{\beta}$                        ▷ Update vector field $J$
        $\alpha_t \leftarrow \begin{cases} 1 - \frac{1}{\beta\|z_t\|} & \beta\|z_t\| > 1 \\ 0 & \text{otherwise} \end{cases}$
        $J_t \leftarrow a_t z_t$
        $c \leftarrow \left(\sum_t B_t^\top B_t\right)^{-1} \left[\sum_t B_t^\top \left(\frac{y_t}{\beta} + J_t - w_t\right)\right]$   ▷ Update coefficients; can pre-factor
        $y_t \leftarrow y_t + \beta(J_t - B_t c - w_t)$                         ▷ Update dual

    return $J_t \ \forall t \in T$
```

Figure 2.5: ADMM algorithm for optimizing $\mathcal{W}_1$, derived in the supplemental document, with parameter $\beta > 0$.

To derive the ADMM approach, we define per-triangle vectors J_t and solve the following equivalent problem:

$$\begin{array}{rl} \inf_{c,J} & \sum_t \|J_t\| \\ \text{s.t.} & J_t = B_t c + w_t. \end{array}$$

This constrained optimization has the augmented Lagrangian:

$$L_\beta \stackrel{\text{def.}}{=} \sum_t \left[\|J_t\| + y_t^\top (J_t - B_t c - w_t) + \frac{\beta}{2} \|J_t - B_t c - w_t\|^2 \right].$$

ADMM alternates between three steps detailed below:

$$\begin{aligned} J &\leftarrow \arg\min_J L_\beta(J, c, y) \\ c &\leftarrow \arg\min_c L_\beta(J, c, y) \\ y_t &\leftarrow y_t + \beta(J_t - B_t c - w_t). \end{aligned}$$

J **update.** We can optimize L_β over J independently for each face since the sum over t decouples in this step. Defining $J_t^0 = B_t c + w_t$ and henceforth in this section dropping the t subscript, we wish to solve

$$\min_J \left[\|J\| + y^\top J + \frac{\beta}{2} \|J - J^0\|^2 \right].$$

This objective is convex, and we could run generic machinery. But in fact we can solve this problem in closed form via the derivation below.

We can simplify the optimization objective by "completing the square:"

$$\begin{aligned}\|J\| + y^\top J + \frac{\beta}{2}\|J - J^0\|^2 &= \|J\| + y^\top J + \frac{\beta}{2}(\|J\|^2 - 2(J^0)^\top J) + \text{const.}\\ &= \|J\| + \frac{\beta}{2}\|J\|^2 + (y - \beta J^0)^\top J + \text{const.}\\ &= \|J\| + \frac{\beta}{2}\left[\|J\|^2 + \frac{2}{\beta}(y - \beta J^0)^\top J\right] + \text{const.}\\ &= \|J\| + \frac{\beta}{2}\left[\|J\|^2 - 2z^\top J\right] + \text{const.}\\ &= \|J\| + \frac{\beta}{2}\|J - z\|^2 + \text{const.}\end{aligned}$$

Here, we defined $z \overset{\text{def.}}{=} -\frac{1}{\beta}(y - \beta J^0)$. So, we equivalently can solve the following optimization:

$$\min_J \left[\|J\| + \frac{\beta}{2}\|J - z\|^2\right],$$

In this form, it is clear we can write $J = az$ for some $a \in \mathbb{R}$ (to prove this separate J into components orthogonal and parallel to z; the former must be zero). Then, we can write

$$\min_a \left[|a|\|z\| + \frac{\beta}{2}(a-1)^2\|z\|^2\right],$$

or equivalently

$$\min_a \left[|a| + d(a-1)^2\right],$$

where $d = \frac{\beta}{2}\|z\|$. This final simplification is solvable using elementary techniques. Clearly $a \in [0, 1]$, so $|a| = a$. If $f(a) = a + d(a-1)^2$, then $f'(a) = 1 + 2d(a-1) = 0 \implies a = 1 - \frac{1}{2d}$. We have $a > 0 \iff 1 - \frac{1}{2d} > 0 \iff d > 1/2$. Hence, in the end we must have

$$a = \begin{cases} 1 - \frac{1}{2d} & d > \frac{1}{2}\\ 0 & \text{otherwise.}\end{cases}$$

c **update.** For this update step, we can write:

$$0 = \nabla_c L_\beta = \sum_t \left[-B_t^\top y_t - \beta B_t^\top (J_t - w_t) + \beta B_t^\top B_t c\right].$$

Dividing by β and moving terms shows:

$$\left(\sum_t B_t^\top B_t\right) c = \sum_t B_t^\top \left(\frac{y_t}{\beta} + J_t - w_t\right).$$

This is a *small* matrix solve if we use the Laplace-Beltrami basis, and it can be prefactored.

2.5 Pointwise Distances

In this section we consider the problem of computing intrinsic distances between points on surface meshes. Using our machinery, we first introduce a family of pointwise distance metrics $d_{\mathcal{W}}^k(\cdot,\cdot)$, with $k = 0,\ldots,n_{\text{vert}}$, and state their theoretical properties. Next, we investigate the practical properties of these distances, including their behavior on realistic meshes, qualitative comparison to commonly used distances, and empirical sensitivity to mesh perturbations.

To define the distance between two mesh points $p, q \in M$, we consider two distributions $\delta_p, \delta_q \in \text{Prob}(M)$ that have mass only at p and q, resp.; discretely, these distributions are nonzero only at individual vertices. Then, we define a distance metric on M as $d_{\mathcal{W}}(p,q) \overset{\text{def.}}{=} \mathcal{W}_1(\delta_p, \delta_q)$. We can compute this distance using spectral approximation as in Section 2.3 with basis size k and denote this approximate distance by $d_{\mathcal{W}}^k(\cdot,\cdot)$. Note that $k = 0$ corresponds to the case when the curl and harmonic terms are removed altogether.

Properties of our family of pointwise distances follow directly from our discussion in previous sections. Both in the discrete and the continuous cases, $d_{\mathcal{W}}^k(\cdot,\cdot)$ is a true distance metric for all values of k. Also, given any two points $p, q \in M$, we have

$$d_{\mathcal{W}}^0(p,q) \geq d_{\mathcal{W}}^1(p,q) \geq \cdots \geq d_{\mathcal{W}}^{n_{\text{vert}}}(p,q) = d_g(p,q),$$

where $d_g(\cdot,\cdot)$ is a discretization of geodesic distance. This final distance may not coincide exactly with the discrete geodesic distance along triangle faces but still satisfies symmetry, the triangle equality, and a discretization of the eikonal equation simultaneously.

The initial member of our family, $d_{\mathcal{W}}^0(\cdot,\cdot)$, has a particularly simple form suitable for efficient implementation. This distance can be computed as $\int_M \|\nabla f(x)\|\, dx$, where f satisfies $\Delta f = \delta_p - \delta_q$. It follows that $f(x) = G(x,p) - G(x,q)$, where $G(\cdot,\cdot)$ is the Green's function of Laplace-Beltrami operator, establishing that:

$$d^0_{\mathcal{W}}(p,q) = \int_M \|\nabla_x G(x,p) - \nabla_x G(x,q))\| \, dx.$$

We can compare $d^0_{\mathcal{W}}$ to a state-of-art spectral distance, the biharmonic distance of [LRF10], which can be written

$$d_b(p,q) = \left(\int_M (G(x,p) - G(x,q))^2 \, dx \right)^{\frac{1}{2}}.$$

Despite the resemblance, there are fundamental differences: In addition to taking gradients, our distance is based on an ℓ_1 rather than ℓ_2 norm. It is a classical result that ℓ_2-norms have smaller embedding capacity than ℓ_1 in that a point set that can be embedded isometrically into ℓ_2 can also be embedded into ℓ_1, but not vice versa [DL09]. In the context of geodesic distances on manifolds, the main obstruction is that non-unique geodesics cannot be supported by Euclidean distance. Indeed, every point x on a shortest geodesic connecting points p and q satisfies $d_g(p,x) + d_g(x,q) = d_g(p,q)$, and under isometric embedding into Euclidean space this means that the image of x lies on the segment connecting the images of p and q under the embedding. For example, geodesic distances on the sphere can be embedded isometrically in ℓ_1 (c.f. [DL09]) but not ℓ_2, since considering all the geodesics connecting the two poles it follows that the entire sphere must be mapped to a straight segment. This helps explain why existing spectral distances, largely based on ℓ_2 norms, have nonintuitive disparately spaced isocontours. Contrastingly, we will see that $d^0_{\mathcal{W}}$ has isocontours that are relatively evenly spaced; we attribute this property to the larger embedding capacity of ℓ_1-norm and to the fact that it has much in common with the larger optimization for $d_{\mathcal{W}}$.

Experiments. Figure 2.6 shows examples of $d^k_{\mathcal{W}}$ for increasing k on a square, a sphere, a half sphere, and a torus. Even the purely spectral distance $d^0_{\mathcal{W}}$ has isotropic and evenly-distributed isolines, especially close to the center point. For example, our distances on the square have convex level sets, unlike the biharmonic distance d_b, which is biased toward the boundary. Similarly, our distances do not stretch at the top of the torus like d_b. As k increases, $d^k_{\mathcal{W}}$ converges to geodesic distance even in the presence of holes and a nonempty

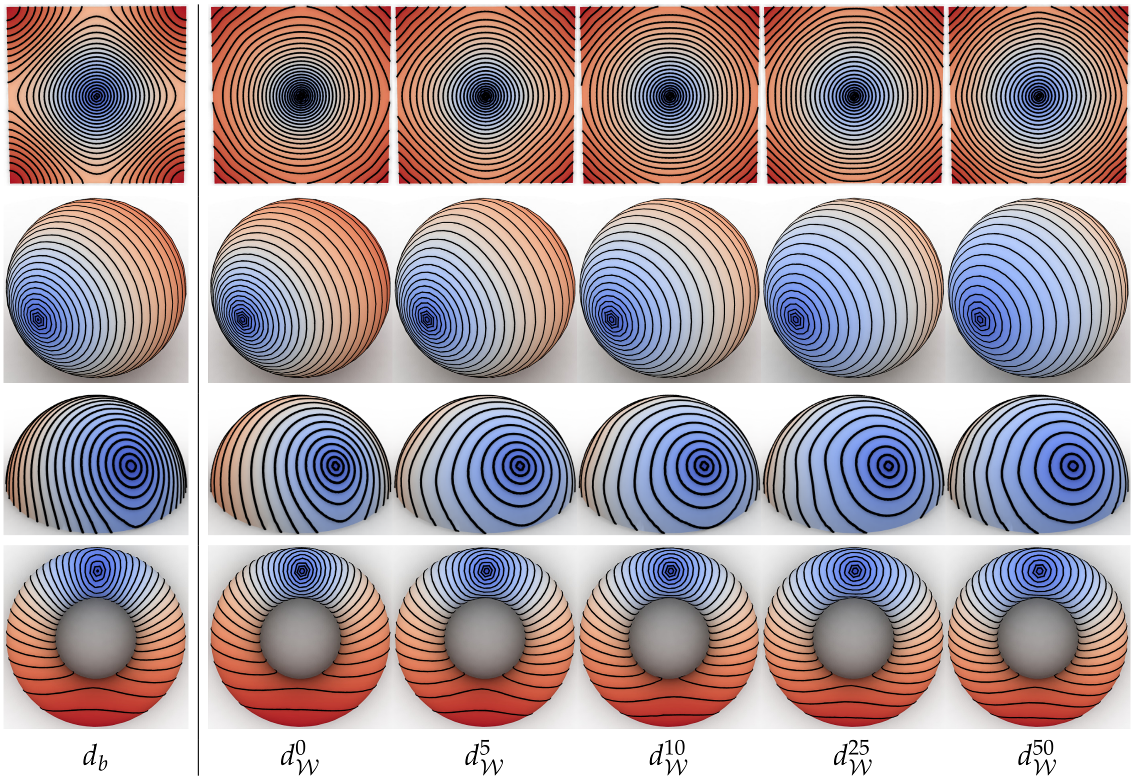


Figure 2.6: Convergence of $d^k_{\mathcal{W}}$ for increasing k and comparison with biharmonic distances d_b [LRF10]. Some anisotropic behavior is specific to these "primitive" shapes, which have spectra with repeated eigenvalues that cannot be grouped in multiples of five; this phenomenon is unlikely on general shapes.

boundary; [CWW13, §3.4] determines boundary conditions experimentally since boundary conditions for the Varadhan theorem only hold when $t \to 0$.

Figure 2.7 shows examples of $d^0_{\mathcal{W}}$, $d^{100}_{\mathcal{W}}$, geodesic distances, and biharmonic distances [LRF10] on a variety of meshes. As can be seen from these images, $d^0_{\mathcal{W}}$ and $d^{100}_{\mathcal{W}}$ both enjoy the best of both the "primal" and "dual" worlds. Like the biharmonic distance, our distances are smooth and follow the natural cross-sections of the shape even in more distant areas. Similarly to geodesic distance, we find that even $d^0_{\mathcal{W}}$ has isotropic and evenly-spaced level sets even though it is the lowest-order approximation of d_g; this is in contrast to biharmonic distance that may have unevenly-spaced isocontours at different parts of a mesh.

A few examples in Figure 2.7 typify the advantages of our new distances. On the boy model, we can see that biharmonic distances are strongly anisotropic in the horizontal

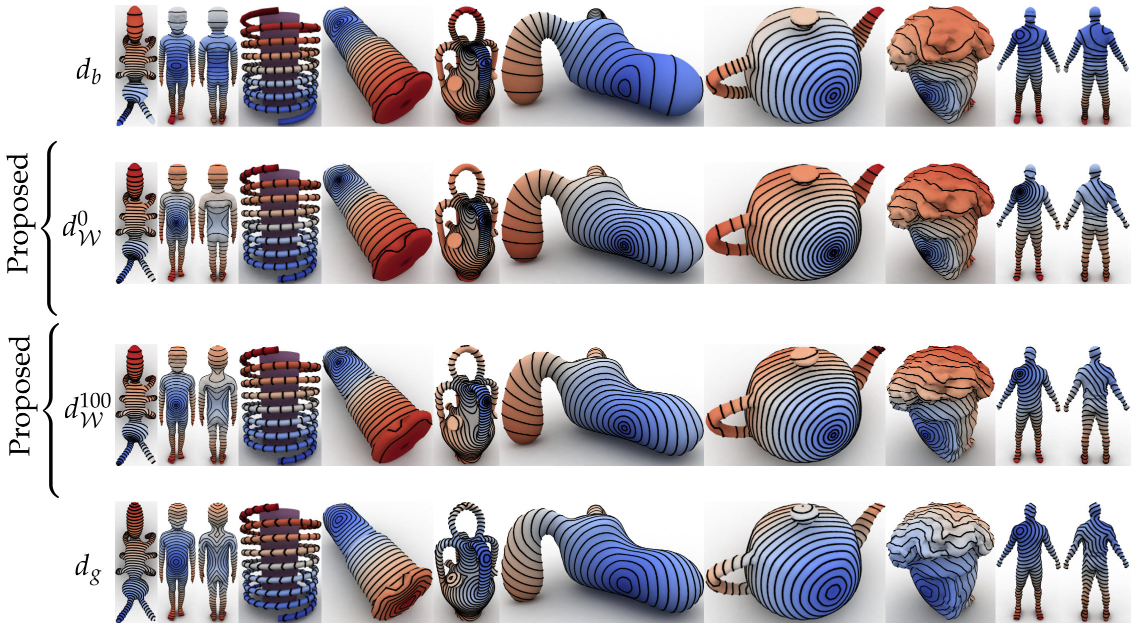


Figure 2.7: Level sets of distance functions on a variety of shapes.

direction and unevenly spaced away from the center point; on the other hand, geodesic distances on the back of the same model have numerous artifacts due to a lack of differentiability. The bearing model also shows similar anisotropy for d_b, while as k increases our distances are able to capture level sets on the cap of the mesh. Our distances also are stable in the hair of the bust model, maintaining a reasonable distribution and smoothness despite high-frequency changes in geometry.

Figure 2.8 demonstrates the stability of $d_{\mathcal{W}}^0$ and $d_{\mathcal{W}}^{100}$ to common geometric perturbations. Figure 2.8(a) shows the insensitivity of these distances to per-vertex noise. Here the addition of Gaussian noise to the mesh leads to little change in the distance as evidenced by the coloring and the isolines. Figure 2.8(b) confirms the theoretical isometry invariance property of our distance — the isolines and coloring are in near correspondence between the armadillo model and its nearly isometric deformation. Finally, Figure 2.8(c) shows insensitivity to tessellation; the distance remains almost unchanged as the mesh is refined considerably.

Figure 2.9 compares our technique to [CWW13]. Metric properties hold for our distances at all levels of spectral truncation even after discretization, while their smoothed

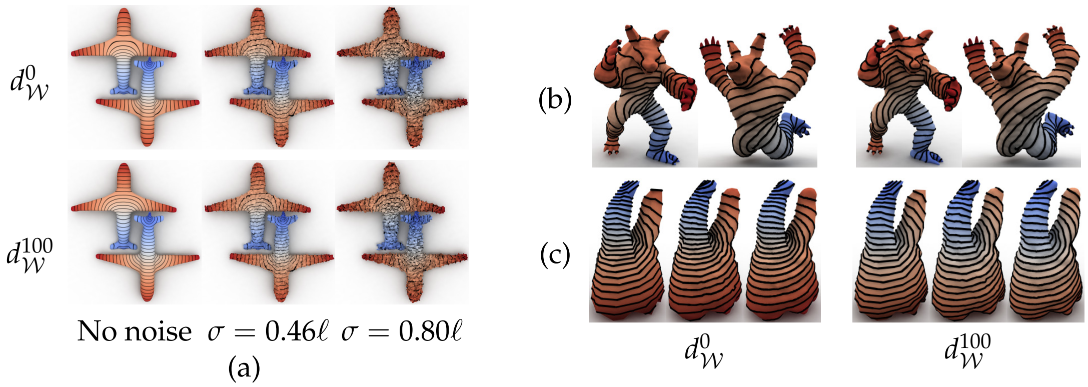


Figure 2.8: Sensitivity of $d^0_{\mathcal{W}}$ and $d^{100}_{\mathcal{W}}$ to geometric noise; vertices are perturbed using a Gaussian distribution (standard deviation written in terms of the average edge length ℓ of the original mesh). (b) Stability to isometric deformation; (c) stability to remeshing (examples have 146, 598, and 9337 vertices, resp.).

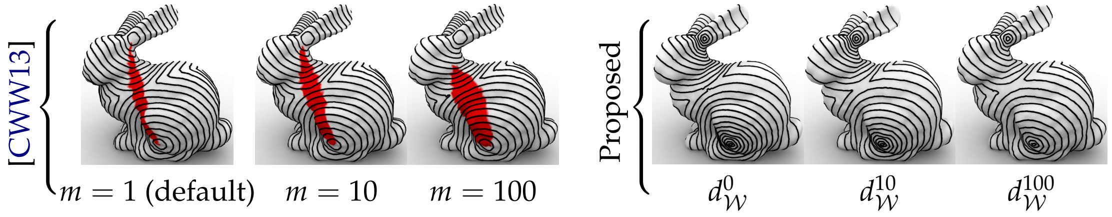


Figure 2.9: For fixed $p, q \in M$, distances using [CWW13] fail to satisfy the triangle inequality $d(p,x) + d(x,q) \geq d(p,q)$ at the red points $x \in M$, shown for various smoothing parameters m; level sets of $f(x) \overset{\text{def.}}{=} \min(d(p,x), d(q,x))$ are shown in black. Contrastingly, $d^k_{\mathcal{W}}$ is guaranteed to satisfy the triangle inequality even after discretization; numerical experiments in the bottom row confirm this relationship.

Mesh	n_{vert}	d_g	d_h	d_b	$d_{\mathcal{W}}^0$	$d_{\mathcal{W}}^{20}$	$d_{\mathcal{W}}^{100}$
Bearing	3182	0.050	0.002	3.52	3.86	30.8	41.4
David	5197	0.096	0.003	10.09	6.18	86.5	121.2
Dog	3716	0.056	0.002	4.66	3.27	38.7	59.8
Teapot	3900	0.063	0.002	6.25	3.87	45.2	57.9
Man	10050	0.18	0.006	42.2	23.2	312.0	511.9

Table 2.1: Timing in seconds for selected experiments in Figure 2.7; the time represents time taken to compute distances from a single source to all targets. In addition to geodesic distances d_g from fast marching, we include timings reported by the optimized C++ implementation of [CWW13] as d_h.

Mesh size		M **for** d_g		M **for** d_h		M **for** d_b		M **for** $d_{\mathcal{W}}^0$	
n_{vert}	n_{tri}	**2**	**100**	**2**	**100**	**2**	**100**	**2**	**100**
2k	4k	0.06	2.60	0.03	0.23	0.03	0.58	0.03	1.22
4k	9k	0.13	6.25	0.05	0.45	0.06	1.42	0.06	2.84
8k	16k	0.24	11.76	0.10	0.97	0.14	4.97	0.14	7.33
16k	32k	0.70	34.93	0.20	1.97	0.33	13.07	0.34	18.45
53k	105k	2.74	121.94	0.71	10.36	1.03	51.99	0.97	68.53
111k	222k	8.06	432.28	2.04	15.14	10.91	289.02	11.00	322.11

Table 2.2: Timing in seconds for all-pairs shortest paths between a sampling of M points.

geodesics at larger and larger diffusion times no longer benefit from an infinitesimal relationship with geodesic distances. This deviation can cause the triangle inequality to fail, as shown in red. Their smoothed distances also are not symmetric, that is, they may not satisfy $d(a,b) = d(b,a)$. While averaging forward and backward distances repairs this issue, it comes at the cost of considerably slower computation for tasks like finding the distance from a single source to all targets, replacing a single linear solve with one for each target.

Computation time. Table 3.1 reports time to compute single-source distances, including geodesic and biharmonic, for a variety of meshes on a 2.40GHz Intel Xeon processor with 23.5GB RAM. The implementation is done in MATLAB, using the ADMM optimization in Figure 2.5. Our new spectral distance $d_{\mathcal{W}}^0$ is efficient to compute by factorizing the Laplacian and performs similarly to the biharmonic distance; in fact, in this test $d_{\mathcal{W}}^0$ outperforms d_b considerably on larger meshes because it requires Green's functions of the Laplace-Beltrami operator rather than the denser bilaplacian operator. Computing smoothed geodesic functions $d_{\mathcal{W}}^k$ introduces computational cost scaling with the number of eigenfunctions.

Table 2.2 compares timing of computing all-pairs distances between a subsample of points on assorted meshes using d_g, d_b, and $d^0_{\mathcal{W}}$. As in Table 3.1, the linear solve step for computing $d^0_{\mathcal{W}}$ takes less time than that for d_b; integrating the derivative of the Laplace-Beltrami Green's function, however, requires an additional iteration over the faces of the mesh, adding computation time for $d^0_{\mathcal{W}}$ in this test. Here, d_b and $d^0_{\mathcal{W}}$ compute one Green's function per source point and then only find pairwise distances between the prescribed points; no such optimization is available for d_g, which must compute distances to all vertices from each source point.

2.6 Volumetric Distances

The problem of computing volumetric distances respecting a given boundary mesh arises in a number of applications, e.g. path planning. Since the straightforward approach of computing shortest paths within a 3D polyhedron is NP-hard [CR87], previous work introduced an approach based on interpolating a prescribed distance on the boundary mesh to the surrounding space inside and outside the shape [RLF09]. While this approach is efficient, it requires an MDS-like embedding of prescribed pairwise distances and hence cannot exactly interpolate the geodesic distance.

Here, we show how our machinery can be used to obtain a volumetric distance reproducing geodesic distance when restricted to the boundary mesh. Like [RLF09], we use barycentric coordinates but in a fundamentally different way—by considering them as distributions and computing EMDs between them.

For a given point x in the interior of M, its barycentric coordinates with respect to a mesh M with vertices $v_i, i = 1, \ldots, n_{\text{vert}}$, are weights $w_i(x), i = 1, \ldots, n_{\text{vert}}$. We recall three properties of these weights: the *Lagrange property* $w_i(v_j) = \delta_{ij}$ (the Kronecker delta); the *partition of unity* property $\sum_i w_i(x) = 1$ with $w_i(x) \geq 0$; and the *linear precision* property $\sum_i w_i(x) v_i = x$.

To compute our volumetric distance between p and q, we consider their barycentric coordinates $\{w_i(p)\}_{i=1}^{n_{\text{vert}}}$ and $\{w_i(q)\}_{i=1}^{n_{\text{vert}}}$ as distributions $\mu_p, \mu_q \in \text{Prob}(M)$; this is possible due to the partition of unity property. Then, our distance is defined as $d_{\mathcal{W}}(p,q) \overset{\text{def.}}{=} \mathcal{W}_1(\mu_p, \mu_q)$; as before, this is a true distance metric. If $d_{\mathcal{W}}(p,q) = 0$, then $\mu_p = \mu_q$, and so $p = q$, because by linear precision property, barycentric coordinates determine the point uniquely.

Our volumetric distance satisfies all of the relevant properties listed in [RLF09]. If $\mathcal{W}_1$ is

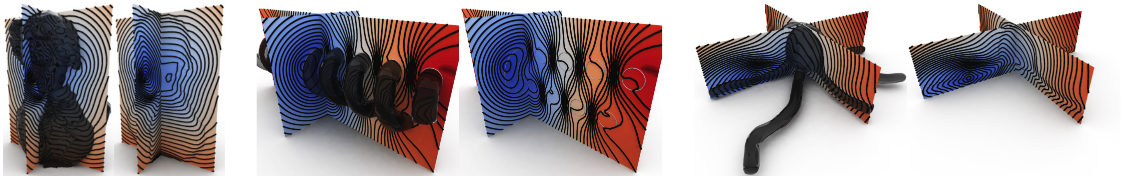


Figure 2.10: Examples of volumetric distances. The left image in each pair shows a surface (Beethoven bust, spiral, octopus resp.) cut by two planes; the right image shows the volumetric distance function on the two planes.

computed without approximation, $d_{\mathcal{W}}(p,q)$ reduces to geodesic distance when $p,q \in M$; indeed, due to the Lagrange property, in this case $\mu_p = \delta_p, \mu_q = \delta_q$, and we are back in the setting of previous section. Furthermore, the following maximum principle holds: if p and q have non-negative barycentric coordinates then the volumetric distance between p and q is no more than the geodesic diameter of the boundary mesh, i.e. $d_{\mathcal{W}}(p,q) \leq \max_{x,y\in M} d_g(x,y)$; this bound follows directly from (1.2) by upper-bounding $d(x,y)$. Also, $d_{\mathcal{W}}$ is bounded below by Euclidean distance, that is $d_{\mathcal{W}}(p,q) \geq \|p-q\|$.

Our differential definition (2.1) of $d_{\mathcal{W}}$ continues to be a distance metric when we allow μ_p and/or μ_q to have negative values in its density function, despite the weaker connection to the theory of optimal transportation. In fact, even our proof of the upper bound $d_{\mathcal{W}}(p,q) \geq \|p-q\|$ remains valid. This relaxation allows us to consider *any* choice of $p,q \in \mathbb{R}^3$ rather than restricting to the interior or convex hull of M, even if coordinates become negative.

Experiments. In our experiments, we use mean value coordinates [JSW05]; these coordinates can become negative both in the interior and exterior of M, but as noted above this departure from $\text{Prob}(M)$ does not raise any issues.

Figure 2.10 shows examples of this distance function in the space around a surface mesh. As before, we select a single source point and then compute the distance to other points in the volume near the object. We visualize these distances on two orthogonal planes using the same color coding as in the previous section.

Given $p,q \in \mathbb{R}^3$, we generate a path from p to q via gradient descent on $d_{\mathcal{W}}(\cdot,q)$ starting at p. Since our distance is fast to compute, its gradient can be computed numerically in relatively little time. Figure 2.11 shows examples of paths from such a process, stepped using simple forward Euler integration. Remarkably, the paths are tuned to the geometry

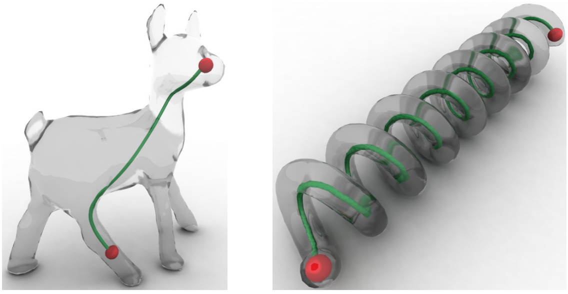


Figure 2.11: Paths constructed by gradient descent on the volumetric distance in the interiors of surfaces.

of the boundary mesh and connect points in the interior without crossing M.

2.7 Additional Applications

Here we suggest some less obvious applications of $\mathcal{W}_1$ to assorted geometry problems. These applications demonstrate the stability of our approach and suggest additional classes of problems for which it can be a valuable tool.

Path planning. We can incorporate the convex energy for $\mathcal{W}_1$ into larger optimizations to formulate applications of our distances to more complex problems. Although the methods in Figures 2.4 and 2.5 no longer apply directly to the problems below, they are all convex and can be optimized using interior point methods.

As an initial example, the distributions ρ_0 or ρ_1 can be promoted to optimization variables to solve path planning problems. For example, suppose M is a mesh of a floor plan and that ρ_0 approximates a distribution of occupants in different parts of M. To find the most efficient way to move all the occupants to a restricted subset of points $S \subseteq M$, we can solve the optimization

$$\begin{aligned} &\inf_{\rho_1} \mathcal{W}_1(\rho_0, \rho_1) \\ &\text{s.t. } \rho_1(x) = 0 \ \forall x \notin S \\ &\qquad \rho_1(x) \geq 0 \ \forall x \in M \\ &\qquad \int_M \rho_1(x)\, dx = 1. \end{aligned} \tag{2.6}$$

Adding a small multiple of $\int_M \rho_1(x)^2\, dx$ has a regularizing effect on ρ_1 when smoothness

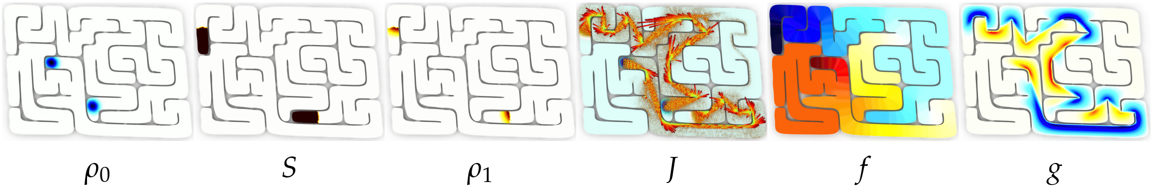


Figure 2.12: A distribution ρ_0 on a maze; a set $S \subseteq M$ of points marked in black; ρ_1 from optimizing (2.6); the corresponding field J; and the functions $f, g : M \to \mathbb{R}$ such that $J = \nabla f + \mathcal{R} \cdot \nabla g$.

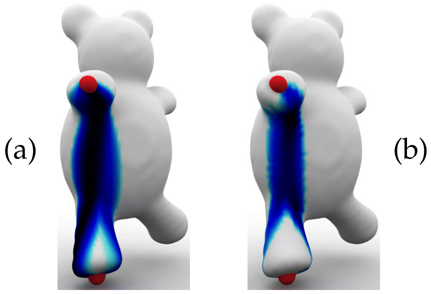


Figure 2.13: (a) Fuzzy geodesic function between the two red points computed using [SCF10]; (b) vector field norms $\|J\|$.

is desired.

Figure 2.12 shows this optimization applied to solving a maze. In this example, ρ_0 is concentrated at two different points on the maze M, which has two possible exits. The optimization (2.6) matches the two points to their closest exits; this can be seen in the vector field J, which traces a path from the starting points to their targets. We also show the decomposition $J = \nabla f + \mathcal{R} \cdot \nabla g$. Here, the gradient part ∇f encodes large-scale motions in the maze while the rotational part $\mathcal{R} \cdot \nabla g$ helps mass round corners in the maze efficiently; the quality of f alone reflects a connection to path planning algorithms using harmonic functions, e.g. [CBW90].

Fuzzy geodesics. Recall our intuition that the vector field $J(x)$ is large at points x that see mass move past as ρ_0 advects toward ρ_1 according to the optimal matching. Suppose ρ_0 and ρ_1 are concentrated near two points $p_0, p_1 \in M$, resp. Then, $J(x)$ is large near geodesic curves between p_0 and p_1.

Inspired by [SCF10], the norm $\|J(\cdot)\| : M \to \mathbb{R}$ provides a "fuzzy geodesic" function related to the likelihood that a geodesic connecting points in the support of ρ_0 to points in

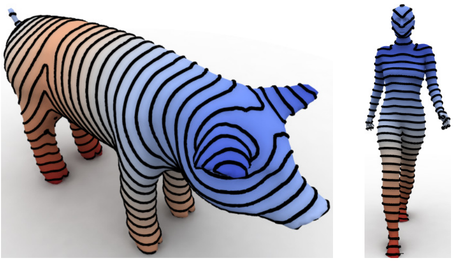


Figure 2.14: Distances computed from distributions (left) on the ears and tail of a pig model and (right) from the collar of a human model.

the support of ρ_1 should pass through $x \in M$. In particular, we can put small Gaussians around p_0 and p_1 and record $\|J\|$ resulting from optimizing (2.1); an example is shown in Figure 2.13.

Distance to features. There are many ways to use our framework to formulate distances that are aware of features rather than points. Most directly, to compute the distance from $p \in M$ to $S \subseteq M$, we can solve (2.6) with ρ_0 concentrated at p and ρ_1 restricted to have support on S. A more efficient alternative, however, is suggested in the proof of Proposition 1. When computing $\mathcal{W}_1$, the Lagrange multiplier $\lambda : M \to \mathbb{R}$ for the $\nabla \cdot J = \rho_1 - \rho_0$ constraint satisfies the eikonal equation and hence is a geodesic distance. Thus, we can compute a geodesic function λ that is aware of S by computing $\mathcal{W}_1$ between the uniform distribution on M and a distribution concentrated on the feature of interest and using the dual multipliers. Figure 2.14 shows this dual variable distance for computing distances to multiple points on a surface and distances to a curve.

Anisotropic distances. As suggested in [San13], the integral $\int_M \|J(x)\|\, dx$ from (2.1) can be replaced with a more general integral $\int_M \|A(x) \cdot J(x)\|\, dx$ to yield anisotropic transportation distances by modifying the metric of M. Figure 2.15 shows two examples in which the function A is a nonnegative scalar guiding shortest paths to favorable areas or avoiding obstacles. In particular, inspired by [LGNL10] we are able to compute distances along a brain model that favor motion along ridges by weighting J using mean curvature. More generally, matrix-valued A can be used favor diffusion in a single direction; we leave consideration of the design of A to future work.

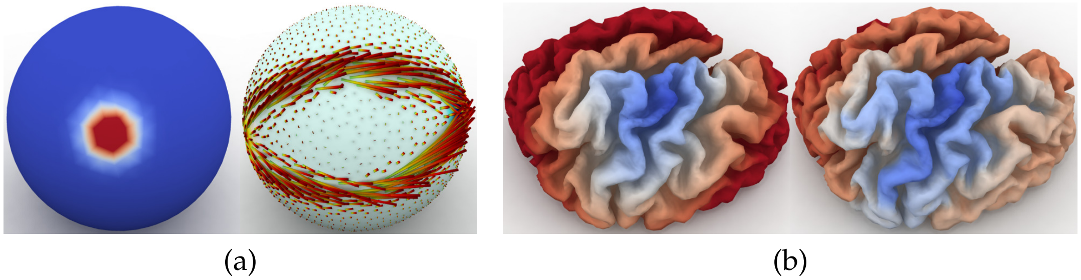


Figure 2.15: (a) The example from Figure 2.3 recomputed using weights on J shown on the left; the resulting J avoids the high-weight area. (b; left) $d_{\mathcal{W}}$ on a brain model; (b; right) anisotropic distances weighted to favor mean curvatures similar to that of the source point; the anisotropic distances on the brain model favor the gyri because the source point is on top of a ridge.

Barycenters. [AC11] suggests minimizing a sum of transportation distances to find the barycenter of a set of distributions on a surface. Rather than resorting to approximations, e.g. in [BvdPPH11, BRPP14], our formulation of $\mathcal{W}_1$ allows us to solve this problem directly on a surface mesh.

Suppose we are given a set $\{\rho_1, \rho_2, \ldots, \rho_k\} \subset \text{Prob}(M)$. We can find a barycenter of these distributions by solving

$$\begin{aligned} &\inf_{\rho} \sum_{i=1}^{k} [\mathcal{W}_1(\rho_i, \rho)]^2 \\ &\text{s.t. } \rho(x) \geq 0 \; \forall x \in M \\ &\qquad \int_M \rho(x)\, dx = 1. \end{aligned} \tag{2.7}$$

The distances $\mathcal{W}_1(\rho_i, \rho)$ are squared to imitate the units of the classical barycenter problem between points on M.

Figure 2.16 shows two applications of this optimization. In the first example, the barycenter of six distributions concentrated on the fingers and side of a hand model is centered at the upper palm; this output accurately represents points equally close to those favored by the six distributions. In the second example, a pointwise barycenter problem is encoded probabilistically using delta distributions at four points on the surface. The optimized ρ has sharp support, making it possible to isolate a single point as the barycenter.

We find empirically that the barycenter of a set of delta distributions is strongly peaked about a single point but defer a discussion of theoretical sharpness properties future work. In this case, however, the optimal optimization objective is exactly the sum of squared

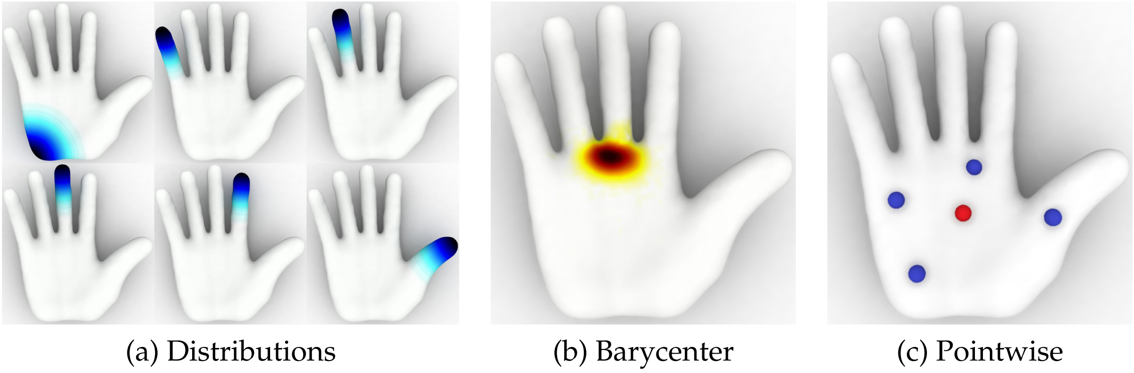


Figure 2.16: (a) Six probability distributions on a surface; (b) the barycenter of these distributions from (2.7); (c) the barycenter of the four blue points in red, computed using the same technique.

geodesic distances from the barycenter to each of the input points. In this way, this strategy reveals an alternative to [PBDSH13] for averaging sets of points on a surface.

2.8 Discussion and Conclusion

It is possible to envision many applications and extensions of the approaches to distance computation presented in this chapter. As we have seen in many instances above, incorporating EMD and its spectral approximations into various geometric optimizations yields meaningful intrinsic information about the surface that easily and efficiently can be incorporated into machinery for larger problems.

While the experiments in Figures 2.7, 2.9, and others show that our technique has desirable properties compared to fast approximations, its runtime is limited thanks to the iterative optimization. Adjustments to methods for computing $d^k_{\mathcal{W}}$ could yield gains in efficiency. Approximating the least-squares solves for the Weiszfeld algorithm in Figure 2.4 may improve timing but more care is needed to guarantee convergence. The ADMM parameter β in Figure 2.5 could be adjusted automatically, e.g. via [HYW00], and alternative methods for geometric median problems may require fewer iterations [Li98, QSZ02, ZTS03, PJ08]. More generally, the optimization (2.1) is a second-order cone program, which can be minimized using commercial solvers.

Our distances also could be extended in various ways. The EMD is the $p = 1$ member of the class of p-Wasserstein distances between distributions. While we take advantage of the simple structure of the $p = 1$ case, Wasserstein distances with $p > 1$ have stronger regularity properties and can be evaluated using somewhat more involved flow techniques [Vil03]. While some specialized numerical methods exist for this case, our FEM discretization and spectral approximations may provide some insight into these problems and their applications in graphics.

We have focused on solving the differential EMD optimization problem (2.1) on triangle meshes, since they are the most common structures approached in geometry processing, but our formulation largely is general and could be applied to other structures admitting differential operators. For example, [LLZ13, LZ13] and others present divergence, gradient, and Laplacian operators acting directly on point clouds and hence easily can be substituted into our formulation. More abstractly, [JLYY11] and others adapt Helmholtz-Hodge structure to functions on graphs, revealing potential machine learning applications of our work.

Even without these extensions, our distances stand alone as practical tools for geometry processing. Even the lowest-order approximations of our distances are stable, smooth, and geometrically meaningful, and more accurate versions are easily evaluated using the proposed iterative method. Given the innumerable existing uses of EMD, spectral and geodesic distances, and volumetric distance, we are confident that differential earth mover's distances will be a useful and straightforward alternative for characterizing the intrinsic geometry and relationships between features of surfaces.

Chapter 3

Convolutional Wasserstein Distances

The techniques in the previous chapter were specific to computing $\mathcal{W}_1$. This distance between distributions is geometrically meaningful, but optimizations like barycenter computation from §2.7 required manipulation of somewhat unnatural quantities like $\mathcal{W}_1^2$ to avoid a lack of strict convexity. In this chapter, we instead develop a numerical technique for approximating the *two*-Wasserstein distance $\mathcal{W}_2$, which is better suited for optimization problems in which the variable is a distribution minimizing a sum of transportation distances. Rather than discretizing these distances directly, we will satisfy ourselves with a regularized approximation.

3.1 Introduction

Recent developments show that incorporating two-Wasserstein distances into optimization objectives yields powerful tools for manipulating distributions for tasks like density interpolation, barycenter computation, and correspondence estimation. As a simple example, suppose we are given two delta functions δ_x, δ_y centered at $x, y \in \mathbb{R}^2$. While the Euclidean average $(\delta_x+\delta_y)/2$ is bimodal at x and y, solving for the distribution that minimizes the sum of squared two-Wasserstein distances to δ_x and δ_y yields a Dirac at the midpoint $(x+y)/2$, thus offering a geometric notion of the midpoint of two distributions. This behavior is specific to optimizations involving the p-Wasserstein distance with $p > 1$ rather than $p \geq 1$; when $p = 1$, an analogous optimization admits multiple optima including the undesirable Euclidean average.

A limiting factor in two-Wasserstein distance evaluation is the complexity of the underlying minimization problem. As with $\mathcal{W}_1$, the linear program describing $\mathcal{W}_2$ has a quadratic number of variables and time complexity scaling at least cubically in the size of the domain [Bc99]. A flow-based formulation for $\mathcal{W}_2$ similar to (2.1) exists, but it exhibits quadratic scaling identical to the original problem due to the introduction of an additional independent variable $t \in [0,1]$ [BB00b]:

$$\mathcal{W}_2(\mu_0,\mu_1)^2 = \begin{cases} \inf_{J,\rho} \int_0^1 \int_M \frac{\|J(x,t)\|^2}{\rho(x,t)}\,dx\,dt \\ \text{s.t. } \nabla \cdot J(x,t) = \frac{\partial \rho(x,t)}{\partial t} \\ \quad J(x,t)\cdot n(x) = 0\ \forall x \in \partial M, t \in [0,1] \\ \quad \rho(x,0) = \rho_0(x)\ \forall x \in M \\ \quad \rho(x,1) = \rho_1(x)\ \forall x \in M \end{cases} \tag{3.1}$$

Furthermore, this optimization problem involves a convex but nonlinear objective, requiring e.g. semidefinite or cone constraints to solve it using standard optimization machinery.

This chapter introduces a fast, scalable numerical framework for problems involving two-Wasserstein distances over geometric domains. This work draws insight from recent advances in machine learning approximating optimal transportation distances using entropic regularization [Cut13]. We adapt this approach to continuous domains using faithful finite elements discretizations of the corresponding optimization problems. This yields a novel approach to optimal transportation without computing or storing pairwise distances on arbitrary shapes.

After discretization, our algorithm for approximating Wasserstein distances becomes a simple iterative scheme with linear convergence, whose iterations require convolution of vectors against discrete diffusion kernels—hence the name *convolutional Wasserstein distance*. We also leverage our framework to design methods for interpolation between distributions, computation of weighted barycenters of sets of distributions, and more complex distribution-valued correspondence problems. Each of these problems is solved with straightforward iterative methods scaling linearly in the size of the data and domain. We demonstrate the versatility of our methods with examples in image processing, shape analysis, and BRDF interpolation.

3.2 Related Work

The key contribution of this chapter is a framework for aggregating and averaging information from multiple densities using $\mathcal{W}_2$. Many of the tasks we consider have been proposed in previous work, although the accompanying computational tools do not scale to the applications we consider. Examples include barycenter computation [AC11], density propagation over graphs [SRLB14], and computation of "soft" correspondence maps [SNB$^+$12]. These problems are typically solved via a multi-marginal linear program [AC11, KP13], which is infeasible for large-scale domains. One work-around approaches the dual of the linear program using L-BFGS with subgradient directions [COO14], but this strategy suffers from poor conditioning and noisy results.

Regularization provides a promising way to approximate solutions of transportation problems and derived models. While interior point methods long have used barrier functions to transform linear programs into strictly convex problems, entropic regularizers in the particular case of optimal transportation provide several key advantages outlined in [Cut13]. With entropic regularization, optimal transportation is solved using an iterative scaling method known as the iterative proportional fitting procedure (IPFP) or Sinkhorn-Knopp algorithm [DS40, Sin67], which can be implemented in parallel GPGPU architectures and used to compute e.g. the barycenter of thousands of distributions [CD14].

Here, we leverage the efficiency of iterative scaling methods for entropy-regularized transport and related problems, principally [Cut13, BCC$^+$15]. By posing regularized transport in continuous language, we couple the efficiency of these algorithms with discretization on domains like surfaces and images. This change is not simply notational but rather leads to much faster iteration through connection to Gaussian kernels on images and the heat kernel of a surface; these kernels can be evaluated without precomputing a matrix of pairwise distances. We demonstrate applications of the resulting methods for large-scale transport on tasks relevant to computer graphics applications.

3.3 Preliminaries

We consider a compact, connected Riemannian manifold M rescaled to have unit volume and possibly with boundary, representing a domain like a surface or image plane. We use

$d : M \times M \to \mathbb{R}_+$ to denote the geodesic distance function, so $d(x, y)$ is the shortest distance from x to y along M. We use $\text{Prob}(M)$ to indicate the space of probability measures on M and $\text{Prob}(M \times M)$ to refer to probability measures on the *product space* of M with itself. To avoid confusion, we will refer to elements $\mu_0, \mu_1, ... \in \text{Prob}(M)$ as *marginals* and to joint probabilities $\pi_0, \pi_1, ... \in \text{Prob}(M \times M)$ as *couplings*.

We refer the reader to §1.4 for basics of optimal transportation on M. More specific to our current discussion, the modified transportation problems we consider involve quantities from information theory, whose definitions we recall in the remainder of this section. We additionally refer the reader to [CT06] for detailed discussion.

A coupling π is *absolutely continuous* with respect to the volume measure when it admits a density function p, so that $\pi(U) = \int_U p(x, y)\, dx\, dy\,, \forall U \subseteq M \times M$. To simplify notation, we will use π to indicate both the measure and its density.

The (differential) entropy of a coupling π on $M \times M$ is defined as the concave energy

$$H(\pi) \overset{\text{def.}}{=} -\iint_{M \times M} \pi(x, y) \ln \pi(x, y)\, dx\, dy. \tag{3.2}$$

By definition, $H(\pi) = -\infty$ when π is not absolutely continuous, and $H(\pi) = 0$ when π is a measure of uniform density $\pi(x, y) \equiv 1$.

Given an absolutely continuous measure $\pi \in \text{Prob}(M \times M)$ and a positive function $\mathcal{K}$ on $M \times M$, we define the *Kullback-Leibler* (KL) divergence between π and $\mathcal{K}$ as

$$\text{KL}(\pi|\mathcal{K}) \overset{\text{def.}}{=} \iint_{M \times M} \pi(x, y) \left[\ln \frac{\pi(x, y)}{\mathcal{K}(x, y)} - 1 \right] dx\, dy. \tag{3.3}$$

3.4 Regularized Optimal Transportation

In this section, we present a modification of two-Wasserstein distances suitable for computation on geometric domains. In our exposition, we first assume that the pairwise distance function $d(\cdot, \cdot)$ is known and then leverage heat kernels to alleviate this requirement.

3.4.1 Entropy-Regularized Wasserstein Distance

Following e.g. [Cut13, BCC+15], we modify the objective of the optimal transportation problem by adding an entropy term $H(\pi)$ promoting spread-out transportation plans π.

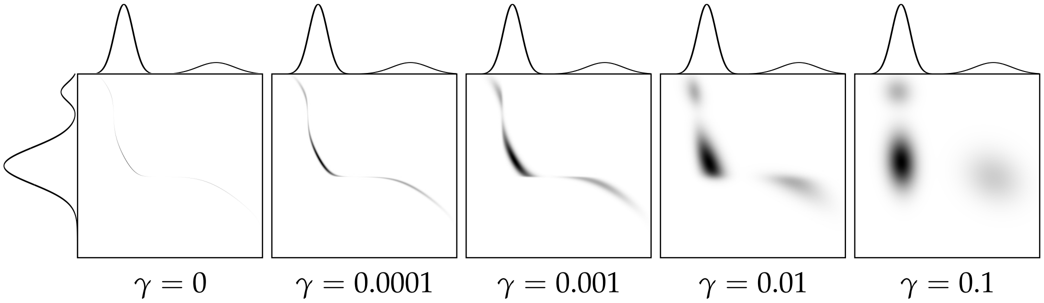


Figure 3.1: Transportation plans with different values of γ, with 1D quadratic costs; $\mu_0, \mu_1 \in \text{Prob}([0,1])$ are shown on the axes.

The *entropy-regularized* 2-*Wasserstein distance* is then defined as:

$$\mathcal{W}^2_{2,\gamma}(\mu_0, \mu_1) \overset{\text{def.}}{=} \inf_{\pi\in\Pi} \left[\iint_{M\times M} d(x,y)^2\, \pi(x,y) dx dy - \gamma H(\pi) \right], \tag{3.4}$$

where we have used the shorter notation Π for $\Pi(\mu_0, \mu_1)$. This regularized version of optimal transport is often called the "Schrödinger problem," and we refer to [Léo12] for discussion of its connection to non-regularized transport, recovered as $\gamma \to 0$.

When $\gamma > 0$, the solution π to (3.4) is an absolutely continuous measure, since otherwise the entropy term is indefinite. The term $-H(\pi)$ also makes the objective strictly convex, and therefore a unique minimizer exists. Figure 3.1 illustrates couplings π obtained using increasing values of γ, resulting in increasingly smooth solutions.

We can associate the distance $d(\cdot,\cdot)$ to a kernel $\mathcal{K}_\gamma$ of the form:

$$\mathcal{K}_\gamma(x,y) = e^{-d(x,y)^2/\gamma}, \quad d(x,y)^2 = -\gamma \ln \mathcal{K}_\gamma(x,y). \tag{3.5}$$

By combining (3.3), (3.4) and (3.5) algebraically, the entropy-regularized Wasserstein distance can be computed from the smallest KL divergence from a coupling $\pi \in \Pi$ to the kernel $\mathcal{K}_\gamma$:

$$\mathcal{W}^2_{2,\gamma}(\mu_0, \mu_1) = \gamma \left[1 + \min_{\pi\in\Pi} \text{KL}(\pi|\mathcal{K}_\gamma) \right]. \tag{3.6}$$

This minimization is convex, due to the convexity of KL on the first argument π, with linear equality constraints induced by the marginals μ_0 and μ_1. As observed in the discrete case [Cut13, BCC$^+$15], it provides a new interpretation for the regularized transportation problem: the optimal plan π is the projection of the distance-based kernel $\mathcal{K}_\gamma$ onto Π, enforcing marginals while minimizing the loss of information quantified by KL divergence.

3.4.2 Wasserstein Distance via Heat Kernel

So far, our method requires a distance function $d(\cdot,\cdot)$ to construct $\mathcal{K}_\gamma$. This assumption is adequate for domains with analytical and fast algorithms for convolution against $\mathcal{K}_\gamma$, like the image plane. It becomes cumbersome, however, for arbitrary manifolds, since precomputing pairwise distances requires quadratic space and considerable computation time. Instead, we propose an alternative to the distance-based kernel $\mathcal{K}_\gamma$ making our method suitable for arbitrary domains.

Define $\mathcal{H}_t(x,y)$ to be the heat kernel determining diffusion between $x, y \in M$ after time t; in particular, $\mathcal{H}_t$ solves the heat equation $\partial_t f_t = \Delta f_t$ with initial condition f_0 through the map

$$f_t(x) = \int_M f_0(y)\mathcal{H}_t(x,y)\,dy.$$

Similar to [CWW13], we associate the heat kernel $\mathcal{H}_t$ to the geodesic distance function $d(\cdot,\cdot)$ based on the Varadhan's formula [Var67], which states that the distance $d(x,y)$ can be recovered by transferring heat from x to y over a short time interval:

$$d(x,y)^2 = \lim_{t\to 0}\left[-2t\ln\mathcal{H}_t(x,y)\right]. \tag{3.7}$$

Setting $t \overset{\text{def.}}{=} \gamma/2$ in (3.7), we approximate the kernel $\mathcal{K}_\gamma$ as:

$$\mathcal{K}_\gamma(x,y) \approx \mathcal{H}_{\gamma/2}(x,y),$$

and, as an implication, we can replace the convolution of an arbitrary function f against $\mathcal{K}_\gamma$ by the solution of the diffusion equation for a time step $t = \gamma/2$ and with f as the initial condition. We thus denote $\mathcal{W}_{2,\mathcal{H}_t}$ as the diffusion-based approximation of $\mathcal{W}^2_{2,\gamma}$, i.e.:

$$\mathcal{W}^2_{2,\mathcal{H}_{\gamma/2}}(\mu_0,\mu_1) \overset{\text{def.}}{=} \gamma\left[1 + \min_{\pi\in\Pi}\mathrm{KL}(\pi|\mathcal{H}_{\gamma/2})\right]. \tag{3.8}$$

Developing conditions for convergence of $\mathcal{W}^2_{2,\mathcal{H}_{\gamma/2}}$ as $\gamma \to 0$ is a challenging topic for future research.

Although $\mathcal{W}_{2,\mathcal{H}}$ and $\mathcal{W}_{2,\gamma}$ are symmetric in μ_0 and μ_1, the self-distances $\mathcal{W}_{2,\mathcal{H}}(\mu,\mu)$ and

$\mathcal{W}^2_{2,\gamma}(\mu,\mu)$ are never exactly zero for a given μ. We also observe that these values only satisfy the triangle inequalities approximately, notably for small γ (see [Cut13, Theorem 1]). Hence, as in [CWW13], the regularized quantities we manipulate are not distances, strictly speaking. These approximations are, however, a very small price to pay to obtain algorithms scaling near-linearly with the size of the mesh.

3.5 Convolutional Wasserstein Distance

We now detail our numerical framework to carry out regularized optimal transportation on discretized domains. Our method computes regularized Wasserstein distances by constructing optimal transportation plans through iterative kernel convolutions—we thus name the results *convolutional Wasserstein distances*. In what follows, we use $\oslash$ and $\otimes$ to indicate elementwise division and multiplication.

Requirements for computing convolutional distances are minimal:

- The domain M, discretized into n elements, with functions and densities represented as vectors $f \in \mathbb{R}^n$.
- A vector $a \in \mathbb{R}^n_+$ of "area weights," with $a^\top \mathbb{1} = 1$, defined so that
$$\int_M f(x)\,dx \approx a^\top f.$$
- A symmetric matrix H_t discretizing the kernel $\mathcal{H}_t$ such that
$$\int_M f(y)\mathcal{H}_t(\cdot,y)dy \;\approx\; H_t(a \otimes f).$$
It is sufficient to know how to *apply* H_t to vectors, rather than storing it explicitly as a matrix in $\mathbb{R}^{n\times n}_{+,*}$.

For images, the natural discretization is an $n_1 \times n_2$ grid of pixels (so $n = n_1 n_2$). In this case, $a \overset{\text{def.}}{=} \mathbb{1}/n_1 n_2$ and H_t is the operator convolving images with a Gaussian of standard deviation $\sigma^2 = \gamma$. Notice that Varadhan's theorem is not needed in this domain, since the heat kernel of the plane is *exactly* a Gaussian in distance.

For triangle meshes, we take n to be the number of vertices and the area vector a as lumped areas proportional to the sum of triangle areas adjacent to a given vertex. Given the cotangent Laplacian $L \in \mathbb{R}^{n\times n}$ [Mac49] and a diagonal area matrix D_a (D_v denotes the diagonal matrix with elements in vector v), we discretize the heat kernel by solving the diffusion equation via an implicit Euler integration [DMSB99] with time step $t = \gamma/2$, i.e.,

$$w = H_t(a \otimes v) \iff (D_a + \gamma/2 L)\, w = a \otimes v.$$

$D_a + \gamma/2L$ can be pre-factored before distance computation, rendering heat kernel convolution equivalent to a near-linear time back-substitution. This feature is particularly valuable since we apply the heat kernel repeatedly. Our implementation uses a sparse Cholesky factorization [Dav06] with γ proportional to the maximum edge length [CWW13]; higher accuracy can be obtained via substeps. Our discretization generalizes to geometric domains like point clouds, tetrahedral meshes, graphs, and polygonal surfaces with well-established discrete Laplacians (and therefore heat kernels).

We encode a distribution $\mu \in \operatorname{Prob}(M)$ as a vector $p \in \mathbb{R}^n_+$ with $a^\top p = 1$ and a distribution $\pi \in \operatorname{Prob}(M \times M)$ as $\pi \in \mathbb{R}^{n\times n}_+$ with $a^\top \pi a = 1$. The discrete KL divergence between a discrete distribution π and an arbitrary $H \in \mathbb{R}^{n\times n}_{+,*}$ is then defined as

$$\mathrm{KL}(\pi|H) \overset{\text{def.}}{=} \sum_{ij} \pi_{ij} a_i a_j \left[\ln \frac{\pi_{ij}}{H_{ij}} - 1\right]. \tag{3.9}$$

Given discrete distributions p_0 and p_1, we model plans $\pi \in \Pi(p_0, p_1)$ as matrices $\pi \in \mathbb{R}^{n\times n}_+$ with $\pi a = p_0$ and $\pi^\top a = p_1$. Finally, the convolutional Wasserstein distance is computed via

$$\boxed{\mathcal{W}^2_{2,H_t}(p_0, p_1) \overset{\text{def.}}{=} \gamma \left[1 + \min_{\pi \in \Pi} \mathrm{KL}(\pi|H_t)\right].} \tag{3.10}$$

Similarly to the continuous case, the minimization in (3.10) is convex with linear constraints on π. Its complexity is tied to the variable π, which scales quadratically in n. We overcome this issue using the following result:

Proposition 4. *The transportation plan $\pi \in \Pi(p_0, p_1)$ minimizing* (3.10) *is of the form $\pi = D_v H_t D_w$, with unique vectors $v, w \in \mathbb{R}^n$ satisfying*

$$\begin{cases} D_v H_t D_w a = p_0, \\ D_w H_t D_v a = p_1. \end{cases} \tag{3.11}$$

Proof. Decompressing notation, the optimization can be written as

$$\begin{array}{rl}\min_{\pi\in\mathbb{R}^{n\times n}} & \sum_{ij}\pi_{ij}\ln\left[\frac{\pi_{ij}}{eH_{ij}}\right]a_ia_j\\ \text{s.t.} & \pi a = p_0\\ & \pi^\top a = p_1.\end{array}$$

After introducing Lagrange multipliers $\lambda_0, \lambda_1 \in \mathbb{R}^n$, the first-order optimality conditions for this system take the form

$$-a_ia_j\ln\frac{\pi_{ij}}{H_{ij}} = a_j\lambda_{0i} + a_i\lambda_{1j}\ \forall i,j\in\{1,\ldots,n\}.$$

Equivalently, we can write

$$\pi_{ij} = H_{ij}\exp\left(-\frac{\lambda_{0i}}{a_i}\right)\exp\left(-\frac{\lambda_{1j}}{a_j}\right).$$

Take $v \overset{\text{def.}}{=} \exp(-\lambda_0 \oslash a)$ and $w \overset{\text{def.}}{=} \exp(-\lambda_1 \oslash a)$, where $\oslash$ denotes elementwise division. Then, this last expression shows $\pi = D_vH_tD_w$. Applying symmetry of H_t and substituting into the two constraints shows (3.11). □

Therefore, rather than computing a matrix π, we can instead compute a pair of vectors (v, w), reducing the number of unknowns to $2n$. This proposition generalizes a result in [Cut13] with the introduction of area weights a. We can find (v, w) by alternating projections onto the linear marginal constraints via an area-weighted version of *Sinkhorn's algorithm* [Sin64], detailed in Algorithm 1.

We simplify the convolutional distance evaluated at the end of the Sinkhorn algorithm as follows:

$$\begin{aligned}\gamma\left[1+\mathrm{KL}(\pi|H_t)\right] &= \gamma\sum_{ij}\pi_{ij}\ln\frac{\pi_{ij}}{(H_t)_{ij}}a_ia_j\\ &= \gamma\sum_{ij}\pi_{ij}\ln(v_iw_j)a_ia_j \text{ since } H_t = D_vH_tD_w\\ &= \gamma\left[\sum_i a_i(\ln v_i)\sum_j\pi_{ij}a_j + \sum_j a_j(\ln w_j)\sum_i\pi_{ij}a_i\right]\\ &= \gamma\left[\sum_i a_i(\ln v_i)p_{0i} + \sum_j a_j(\ln w_j)p_{1j}\right]\end{aligned}$$

```
function CONVOLUTIONAL-WASSERSTEIN($p_0, p_1; H_t, a$)
    // Sinkhorn iterations
    $v, w \leftarrow \mathbb{1}$
    for $i = 1, 2, 3, \ldots$
        $v \leftarrow p_0 \oslash H_t(a \otimes w)$
        $w \leftarrow p_1 \oslash H_t(a \otimes v)$

    // KL divergence
    return $\gamma\, a^\top [(p_0 \otimes \ln v) + (p_1 \otimes \ln w)]$
```

Algorithm 1: Sinkhorn iteration for convolutional Wasserstein distances. $\otimes, \oslash$ denote elementwise multiplication and division, resp.

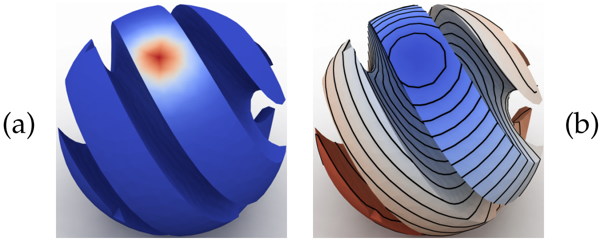


Figure 3.2: $\mathcal{W}^2_{2,H_t}$ between δ distributions (a) as a vertex-to-vertex distance (b; computed with $\gamma = 10^{-5}$ — slight smoothing).

$$\begin{aligned} &\text{since } \pi a = p_0 \text{ and } \pi^\top a = p_1 \\ &= \gamma\, a^\top \left[(p_0 \otimes \ln v) + (p_1 \otimes \ln w) \right]. \end{aligned}$$

As in [SCV14], $\mathcal{W}^2_{2,H_t}$ between distributions centered at individual vertices can be used as point-to-point distances. Figure 3.2 shows one example computed using our algorithm. The resulting pointwise distance squared is exactly the logarithm of H_t. Since Crane et al. [CWW13] previously proposed a specialized algorithm using the heat kernel for pointwise distances via this approximation, we instead will focus on more general problems involving optimal transportation not considered in their work.

Timing & numerics. To evaluate efficiency, we compare three approaches to approximating $\mathcal{W}_2$: a linear program discretizing (1.2) (with $p = 2$), regularized distances with a full distance-based kernel [Cut13], and convolutional Wasserstein distances $\mathcal{W}^2_{2,H_t}$. The linear program is solved using state-of-the-art parallel optimization [MOS14], with all-pairs distances along mesh edges from an $O(n^2 \log n)$ algorithm [Joh77]. [Cut13] and our

$\lvert V\rvert$	$\lvert T\rvert$	PD	LP	[Cut13]	PF	$\mathcal{W}^2_{2,H_t}$
693	1382	0.10	9.703	0.625	0.00	1.564
1150	2296	0.28	36.524	1.284	0.00	0.571
1911	3818	0.79	*	2.725	0.02	1.010
3176	6348	2.15	*	5.435	0.03	1.553
5278	10552	6.47	*	10.490	0.06	2.477
8774	17544	18.55	*	23.326	0.10	4.516
14584	29164	53.41	*	*	0.17	8.152

Table 3.1: Timing (in sec.) for approximating $\mathcal{W}_2$ between random distributions on triangle meshes, averaged over 10 trials. An asterisk * denotes time-out after one minute. Pairwise distance (PD) computation is needed for the linear program (LP) and [Cut13]; timing for this step is written separately. Cholesky pre-factorization (PF) is needed for convolutional distance and is similarly separated.

convolutional distances are implemented in Matlab, the former using the all-pairs distance matrix converted to a kernel and the latter using pre-factored Cholesky decomposition. All tests were run with tolerance 10^{-5} on a 2.40GHz Intel Xeon processor with 23.5GB RAM; for this test, γ is chosen as 1% of the median transport cost.

Table 3.1 shows results of this experiment on meshes of the same shape with varying density. Both regularized approximations of $\mathcal{W}_2$ outperform the linear program by a significant margin that grows with the size of the problem. Our method also outperforms [Cut13] with a dense kernel matrix, both by avoiding explicit pairwise distance computation and via the pre-factored diffusion operator; the difference is particularly notable on large meshes for which the kernel takes a large amount of memory. The one exception is the smallest mesh, for which our method took longer to converge due to numerical issues from the discretized heat equation.

The Sinkhorn algorithm is known to converge at a linear rate [FL89, Kni08], and similar guarantees exist for alternating projection methods [ER11]. These bounds give a rough indicator of the number of iterations needed to compute convolutional distances and derived quantities used in §3.6. In practice, the convergence rate depends on the sharpness of the kernel and of the distributions p_0 and p_1. The experiments reported in Table 3.1 show that the time to convergence is reasonable for challenging cases; most distance computations converge within 10-20 iterations when γ was chosen on the order of the average edge length, with faster convergence as γ is increased. Finally, we point out that numerical issues may appear when γ is smaller than the resolution of the domain, since the kernel operator may become ill-conditioned.

3.6 Optimization Over Distances

An advantage of convolutional Wasserstein distances is the variety of optimizations into which they can be incorporated. Then, the goal is not to evaluate Wasserstein distances but rather to optimize for distributions minimizing an objective constructed out of them.

3.6.1 Wasserstein Barycenters

The *Wasserstein barycenter* problem attempts to summarize a collection $(\mu_i)_{i=1}^k$ of probability distributions by taking their weighted average with respect to the Wasserstein distance. Following [AC11], given a set of weights $\alpha = (\alpha_i)_{i=1}^k \in \mathbb{R}_+^k$, it is defined as the following convex problem over the space of measures

$$\min_\mu \textstyle\sum_{i=1}^k \alpha_i \mathcal{W}_2^2(\mu, \mu_i). \tag{3.12}$$

After discretization, we can pose the barycenter problem as

$$\min_p \textstyle\sum_{i=1}^k \alpha_i \mathcal{W}_{2,H_t}^2(p, p_i). \tag{3.13}$$

Substituting transportation plans yields an equivalent problem:

$$\begin{array}{rl} \min_{\{\pi_i\}} & \sum_{i=1}^k \alpha_i \mathrm{KL}(\pi_i | H_t) \\ \text{s.t.} & \pi_i^\top a = p_i \ \forall i \in \{1, \ldots, k\} \\ & \pi_i a = \pi_1 a \ \forall i \in \{1, \ldots, k\}. \end{array}$$

The first constraint enforces that π_i marginalizes to p_i in one direction, and the second constraint enforces that all the π_i's marginalize to the same p in the opposite direction.

As suggested by Benamou et al. [BCC+15], the expanded problem can be viewed as a *projection* with respect to KL divergence from H_t (repeated k times) onto the constraint set $\mathcal{C}_1 \cap \mathcal{C}_2$, where

$$\begin{aligned} \mathcal{C}_1 &\overset{\text{def.}}{=} \{(\pi_i)_{i=1}^k : \pi_i^\top a = p_i \ \forall i \in \{1, \ldots, k\}\} \\ \mathcal{C}_2 &\overset{\text{def.}}{=} \{(\pi_i)_{i=1}^k : \pi_i a = \pi_j a \ \forall i, j \in \{1, \ldots, k\}\}. \end{aligned}$$

Problems of this form can be minimized using *iterated Bregman projection* [Bre67], which initializes all the π_i's to H_t and then cyclically projects the current iterate onto one $\mathcal{C}_i$ at a time. Unlike the full optimization, projections onto $\mathcal{C}_1$ and $\mathcal{C}_2$ individually can be written in closed form, as explained in the following propositions:

Proposition 5. *The KL projection of $(\pi_i)_{i=1}^k$ onto $\mathcal{C}_1$ satisfies $\operatorname{proj}_{\mathcal{C}_1} \pi_i = \pi_i D_{p_i \oslash \pi_i^\top a}$ for each $i \in \{1, \ldots k\}$.*

Proof. The problem decouples, and hence projection can be carried out one transportation matrix at a time. Expanding the objective for a single transportation matrix yields the following problem:

$$\begin{array}{rl}\min_{\bar\pi \in \mathbb{R}^{n\times n}} & \sum_{ij} \bar\pi_{ij} \ln \left[\frac{\bar\pi_{ij}}{e\pi_{ij}}\right] a_i a_j \\ \text{s.t.} & \bar\pi^\top a = p,\end{array}$$

where $\bar\pi$ is the projection of π onto $\mathcal{C}_1$. For Lagrange multiplier $\lambda \in \mathbb{R}^n$, the first-order optimality condition for element $\bar\pi_{ij}$ is

$$-a_i a_j \ln \frac{\bar\pi_{ij}}{\pi_{ij}} = a_i \lambda_j \implies \bar\pi_{ij} = \pi_{ij} \exp\left(-\frac{\lambda_j}{a_j}\right).$$

After taking $c \overset{\text{def.}}{=} \exp(-\lambda \oslash a)$, this expression shows $\bar\pi = \pi D_c$. Since $\bar\pi^\top a = p$, we now can write $D_c \pi^\top a = p$, showing $c = p \oslash \pi^\top a$, as needed. □

Proposition 6. *The KL projection of $(\pi_i)_{i=1}^k$ onto $\mathcal{C}_2$ satisfies $\operatorname{proj}_{\mathcal{C}_2} \pi_i = D_{p \oslash d_i} \pi_i$ for each $i \in \{1, \ldots k\}$, where $d_i = \pi_i a$ and $p = \prod_i d_i^{\alpha_i / \sum_\ell \alpha_\ell}$.*

Proof. Take $(\bar\pi_i)_{i=1}^k$ to be the projection onto $\mathcal{C}_2$, with unknown common marginal p. As in [BCC+15], expanding the optimization problem provides the form

$$\begin{array}{rl}\min_{\{\bar\pi_\ell\}, p} & \sum_{\ell ij} \alpha_\ell \bar\pi_{\ell ij} \ln \left[\frac{\bar\pi_{\ell ij}}{e\pi_{\ell ij}}\right] a_i a_j \\ \text{s.t.} & \bar\pi_\ell a = p \; \forall \ell \in \{1, \ldots, k\}.\end{array}$$

The Lagrange multiplier expression for this optimization is

$$\Lambda \overset{\text{def.}}{=} \sum_\ell \left(\sum_{ij} \alpha_\ell \bar\pi_{\ell ij} \ln \left[\frac{\bar\pi_{\ell ij}}{e\pi_{\ell ij}}\right] a_i a_j + \lambda_\ell^\top (\bar\pi_\ell a - p) \right).$$

Differentiating with respect to $\bar\pi_{\ell ij}$ shows

$$0 = \frac{\partial \Lambda}{\partial \bar\pi_{\ell ij}} = \alpha_\ell a_i a_j \ln \frac{\bar\pi_{\ell ij}}{\pi_{\ell ij}} + \lambda_{\ell i} a_j,$$

or equivalently,

$$\bar{\pi}_{\ell ij} = \pi_{\ell ij} \exp\left(-\frac{\lambda_{\ell i}}{a_i \alpha_\ell}\right).$$

Taking $c_\ell \overset{\text{def.}}{=} \exp(-\lambda_\ell \oslash a)$, we can write $\bar{\pi}_\ell = D_{c_\ell^{1/\alpha_\ell}} \pi_\ell$.

Differentiating Λ with respect to p shows

$$\begin{aligned} &= \nabla_p \Lambda = -\sum_\ell \lambda_\ell \\ \implies \prod_\ell c_\ell &= \exp\left(-\sum_\ell \lambda_\ell \oslash a\right) = \mathbb{1}. \end{aligned}$$

Define $d_\ell \overset{\text{def.}}{=} \pi_\ell a$. Then, substituting our new variables into the constraint $\bar{\pi}_\ell a = p$ shows

$$\begin{aligned} c_\ell^{1/\alpha_\ell} \otimes d_\ell = p \; \forall \ell \\ \implies c_\ell = (p \oslash d_\ell)^{\alpha_\ell}. \end{aligned}$$

Define $A \overset{\text{def.}}{=} \sum_\ell \alpha_\ell$. By the relationship above,

$$\begin{aligned} \mathbb{1} &= \prod_\ell c_\ell = \prod_\ell (p \oslash d_\ell)^{\alpha_\ell} = p^A \prod_\ell d_\ell^{-\alpha_\ell} \\ \implies p &= \prod_\ell d_\ell^{\alpha_\ell / A}. \end{aligned}$$

Hence, $c_\ell^{1/\alpha_\ell} = p \oslash d_\ell$, showing $\bar{\pi}_\ell = D_{p \oslash d_\ell} \pi_\ell$. □

The propositions, originally presented without area weights in [BCC+15], show that the necessary Bregman projections can be carried out via pre- or post-multiplication by diagonal matrices. Hence, we can store and update vectors $v_i, w_i \in \mathbb{R}^n$ so that $\pi_i = D_{v_i} H_t D_{w_i}$. If M is represented using n samples, this reduces storage and algorithmic runtime by a factor of n.

Algorithm 2 documents the barycenter method. It initializes all the π_i's to H_t by taking $v_i = w_i = \mathbb{1}$ for all i and then alternatingly projects using the formulas above. The only operations needed are applications of H_t and elementwise arithmetic. We never need to store the matrix of H_t explicitly and instead *apply* it iteratively; this structure is key when H_t represents a heat kernel obtained by solving a linear system or convolution over an

```
function WASSERSTEIN-BARYCENTER($\{p_i\}, \{\alpha_i\}; H_t, a$)
    // Initialization
    $v_1, \ldots, v_k \leftarrow \mathbb{1}$
    $w_1, \ldots, w_k \leftarrow \mathbb{1}$

    // Iterate over $\mathcal{C}_i$'s
    for $j = 1, 2, 3, \ldots$
        $p \leftarrow \mathbb{1}$
        for $i = 1, \ldots, k$
            // Project onto $\mathcal{C}_1$
            $w_i \leftarrow p_i \oslash H_t(a \otimes v_i)$
            $d_i \leftarrow v_i \otimes H_t(a \otimes w_i)$
            $p \leftarrow p \otimes d_i^{\alpha_i}$

        // Optional
        $p \leftarrow$ ENTROPIC-SHARPENING($p, H_0; a$)

        // Project onto $\mathcal{C}_2$
        for $i = 1, \ldots, k$
            $v_i \leftarrow v_i \otimes p \oslash d_i$
    return $p$
```

Algorithm 2: Wasserstein barycenter using iterated Bregman projection. Both of the inner **for** loops can be parallelized over i.

image.

Entropic Sharpening. Barycenters computed using Algorithm 2 have similar qualitative structure to barycenters with respect to the true Wasserstein distance $\mathcal{W}_2$ but may be smoothed thanks to entropic regularization. This can create approximations of the barycenter that qualitatively appear too diffuse.

We introduce a simple modification of the projection method counteracting this phenomenon. Define the entropy of p to be

$$H(p) \overset{\text{def.}}{=} -\sum_i a_i p_i \ln p_i.$$

We expect the non-regularized Wasserstein barycenter of a set of distributions to have entropy bounded by that of the input distributions $(p_i)_{i=1}^k$. Hence, take $H_0 \overset{\text{def.}}{=} \max_i H(p_i)$

(or a user-specified bound). Then, we can modify the barycenter problem slightly:

$$\begin{array}{rl} \min_p & \sum_{i=1}^k \alpha_i \mathcal{W}_{2,H_t}^2(p,p_i) \\ \text{s.t.} & H(p) \leq H_0. \end{array} \tag{3.14}$$

That is, we wish to find a distribution with bounded entropy that minimizes the sum of transportation distances.

The problem in (3.14) is not convex, but we apply Bregman projections nonetheless. We augment $\mathcal{C}_2$ with an entropy constraint:

$$\overline{\mathcal{C}}_2 \overset{\text{def.}}{=} \mathcal{C}_2 \cap \{(\pi_i)_{i=1}^k : H(\pi_i a) + a^\top \pi_i a \leq H_0 + 1 \; \forall i \in \{1,\ldots,k\}\}.$$

The $a^\top \pi_i a$ term is for algebraic convenience in proving the proposition below; at convergence, $a^\top \pi_i a = 1$ and this term cancels with the 1 on the right-hand side of the inequality. Remarkably, despite the nonconvexity, KL projection onto $\overline{\mathcal{C}}_2$ can be carried out efficiently:

Proposition 7. *There exists $\beta \in \mathbb{R}$ such that the KL projection of $(\pi_i)_{i=1}^k$ onto $\overline{\mathcal{C}}_2$ satisfies $\operatorname{proj}_{\overline{\mathcal{C}}_2} \pi_i = D_{p \oslash d_i} \pi_i$ for all $i \in \{1,\ldots,k\}$, where $d_i = \pi_i a$ and $p = \left(\prod_i d_i^{\alpha_i}\right)^\beta$.*

Proof. Similarly to the previous proposition, we write the optimization problem as follows:

$$\begin{array}{rl} \min_{\{\bar\pi_\ell\},p} & \sum_{\ell ij} \alpha_\ell \bar\pi_{\ell ij} \ln\left[\frac{\bar\pi_{\ell ij}}{e\pi_{\ell ij}}\right] a_i a_j \\ \text{s.t.} & \bar\pi_\ell a = p \; \forall \ell \in \{1,\ldots,k\} \\ & \sum_i a_i p_i (\ln p_i - 1) \geq -H_0 - 1. \end{array}$$

When the constraint is inactive, the optimization is solved by the previous proposition. Hence, we will focus on the active case, that is, when $\sum_i a_i p_i(\ln p_i - 1) = -H_0 - 1$.

The Lagrange multiplier expression for this optimization is

$$\begin{aligned} \Lambda \overset{\text{def.}}{=} & \sum_\ell \left(\sum_{ij} \alpha_\ell \bar\pi_{\ell ij} \ln\left[\frac{\bar\pi_{\ell ij}}{e\pi_{\ell ij}}\right] a_i a_j + \lambda_\ell^\top (\bar\pi_\ell a - p) \right) \\ & + \gamma \left(\sum_i a_i p_i (\ln p_i - 1) + H_0 + 1 \right). \end{aligned}$$

Differentiating with respect to λ_ℓ, γ, π, and p yields the following optimality criteria:

$$\begin{aligned}
p &= \pi^\ell a \;\forall \ell \in \{1, \ldots, k\} \\
-H_0 - 1 &= \sum_i a_i p_i (\ln p_i - 1) \\
0 &= \alpha_\ell a_i a_j \ln \frac{\bar{\pi}_{\ell ij}}{\pi_{\ell ij}} + \lambda_{\ell i} a_j \;\forall i, j, \ell \\
0 &= \gamma a_i \ln p_i - \sum_\ell \lambda_{\ell i} \;\forall i.
\end{aligned}$$

As before, the third condition shows

$$\bar{\pi}_{\ell ij} = \pi_{\ell ij} \exp\left(-\frac{\lambda_{\ell i}}{a_i \alpha_\ell}\right).$$

The fourth condition shows

$$p^\gamma = \exp\left(\sum_\ell \lambda_\ell \oslash a\right).$$

Take $c_\ell \overset{\text{def.}}{=} \exp(-\lambda_\ell \oslash a)$. Then, the conditions above become

$$\begin{aligned}
\bar{\pi}_{\ell ij} &= \pi_{\ell ij} c_{\ell i}^{1/\alpha_\ell} \\
p_i^\gamma &= \prod_\ell c_{\ell i}.
\end{aligned}$$

Define $d_\ell \overset{\text{def.}}{=} \pi_\ell a$. Since $p = \bar{\pi}_\ell a$, for all ℓ we can write

$$p_i = \sum_j \bar{\pi}_{\ell ij} a_j = \sum_j \pi_{\ell ij} c_{\ell i}^{1/\alpha_\ell} a_j = c_{\ell i}^{1/\alpha_\ell} d_{\ell i}.$$

Taking the log of both sides of this expression and the relationship $p_i^\gamma = \prod_\ell c_{\ell i}$ shows

$$\begin{aligned}
\alpha_\ell \ln p_i &= \ln c_{\ell i} + \alpha_\ell \ln d_{\ell i} \;\forall \ell \\
\gamma \ln p_i &= \sum_\ell \ln c_{\ell i}.
\end{aligned}$$

Summing the first equation over ℓ and removing the $c_{\ell i}$ term by the second equation shows

$$\left(-\gamma + \sum_\ell \alpha_\ell\right) \ln p_i = \sum_\ell \alpha_\ell \ln d_{\ell i}$$

```
function ENTROPIC-SHARPENING($p, H_0; a$)
    if $H(p) + a^\top p > H_0 + 1$ then
        $\beta \leftarrow$ FIND-ROOT($a^\top p^\beta + H(p^\beta) - (1 + H_0); \beta \geq 0$)
    else $\beta \leftarrow 1$
    return $p^\beta$
```

Algorithm 3: Entropic sharpening method; we default to $\beta = 1$ when no root exists but rarely encounter this problem in practice.

$$\implies p_i = \prod_\ell d_{\ell i}^{\alpha_\ell / (-\gamma + \sum_{\ell'} \alpha_{\ell'})}.$$

Identically to the previous proposition, $\bar{\pi}_\ell = D_{p \oslash d_\ell} \pi_\ell$, with this new choice of p; taking $\gamma = 0$ recovers the inactive constraint case. Defining

$$\beta \overset{\text{def.}}{=} \frac{1}{-\gamma + \sum_\ell \alpha_\ell}$$

provides the desired formula. □

That is, the entropy-constrained projection step takes the result of the unconstrained projection to the β power to achieve the entropy bound. The exponent β can be computed using single-variable root-finding (e.g. bisection or Newton's method), as shown in Algorithm 3. Empirically, the Bregman algorithm converges to a near-barycenter with limited entropy when using this new projection step as long as H_0 is on the order of the entropy of the p_i's. For difficult test cases, higher-quality barycenters can be recovered by first solving the problem without an entropy constraint and then iteratively introducing entropic sharpening with tightening bounds.

Figure 3.3 illustrates the effect of the bound H_0 on the barycenter of two distributions. As H_0 decreases, the barycenter becomes sharply peaked about its modes, counteracting the aggressive regularization.

3.6.2 Displacement Interpolation

The 2-Wasserstein distance $\mathcal{W}_2$ has a distinguishing *displacement interpolation* property [McC97]. $\mathcal{W}_2(\mu_0, \mu_1)$ is the length of a geodesic $\mu_t : [0, 1] \to \text{Prob}(M)$ in $\text{Prob}(M)$ with respect to a metric induced by squared geodesic distances on M. The time-varying sequence of distributions μ_t transitions from μ_0 to μ_1, moving mass continuously along geodesic

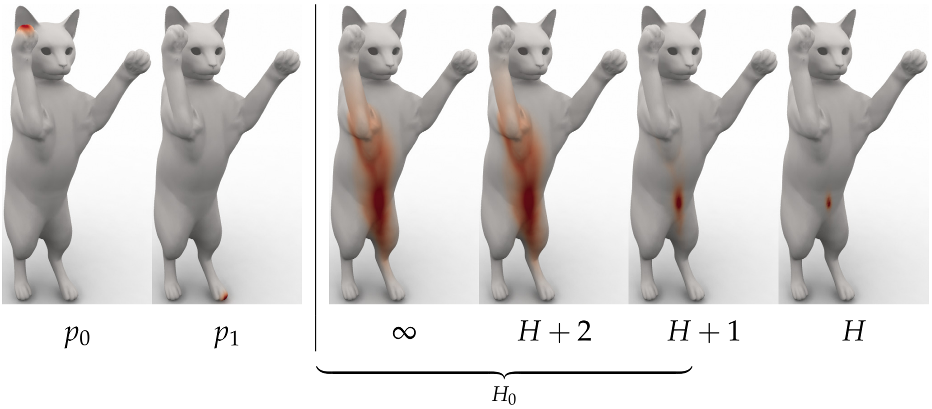


Figure 3.3: Barycenters with different levels of entropic sharpening, controlled by H_0. Here, $H \overset{\text{def.}}{=} \max\{H(p_1), H(p_2)\} \approx -2.569$.

paths on M. As a point of comparison, Chapter 2 uses flows along M to evaluate the 1-Wasserstein distance $\mathcal{W}_1$; the resulting interpolation, however, is given by the trivial formula $\mu_t = (1-t)\mu_0 + t\mu_1$.

Agueh and Carlier [AC11] prove under suitable regularity that the interpolating path μ_t from μ_0 to μ_1 satisfies

$$\mu_t = \inf_{\mu \in \operatorname{Prob}(M)} \left[(1-t)\mathcal{W}_2^2(\mu_0, \mu) + t\mathcal{W}_2^2(\mu, \mu_1)\right], \tag{3.15}$$

for all $t \in [0,1]$. This formula provides a means to compute μ_t directly rather than optimizing an entire path in probability space.

In the discrete case, given $p_0, p_1 \in \operatorname{Prob}(M)$ we wish to find a time-varying p_t interpolating between the two. To do so, we define

$$p_t \overset{\text{def.}}{=} \min_{p \in \operatorname{Prob}(M)} \left[(1-t)\mathcal{W}_{2,H_t}^2(p_0, p) + t\mathcal{W}_{2,H_t}^2(p, p_1)\right]. \tag{3.16}$$

This can be minimized using Algorithm 2 with $\alpha = (1-t, t)$.

Figure 3.4 shows displacement interpolation between two multi-peaked distributions on a triangle mesh, with and without entropic sharpening. Again, sharpening avoids entropy introduced by the regularizer.

```
function WASSERSTEIN-PROPAGATION($V, E, V_0, p(V_0); H_t, a$)
    // Initialization
    $v_1, \ldots, v_{|E|} \leftarrow \mathbb{1}$
    $w_1, \ldots, w_{|E|} \leftarrow \mathbb{1}$
    // Iterate over $\mathcal{C}_i$'s
    for $j = 1, 2, 3, \ldots$
        for $v \in V$
            if $v \in V_0$ then
                $p \leftarrow p_0(v)$
                // Project adjacent $\pi_e$'s
                for $e \in N(v)$
                    if $e = (w, v)$ then $w_e \leftarrow p \oslash H_t(a \otimes v_e)$
                    if $e = (v, w)$ then $v_e \leftarrow p \oslash H_t(a \otimes w_e)$
            else if $v \notin V_0$ then
                // Estimate distribution
                $\omega \leftarrow \sum_{v \in e} \alpha_e$
                $p_v \leftarrow \mathbb{1}$
                for $e \in N(v)$
                    if $e = (w, v)$ then $d_e \leftarrow w_e \otimes H_t(a \otimes v_e)$
                    if $e = (v, w)$ then $d_e \leftarrow v_e \otimes H_t(a \otimes w_e)$
                    $p_v \leftarrow p_v \otimes d_e^{\alpha_e / \omega}$
                for $e \in N(v)$
                    if $e = (w, v)$ then $w_e \leftarrow w_e \otimes p_v \oslash d_e$
                    if $e = (v, w)$ then $v_e \leftarrow v_e \otimes p_v \oslash d_e$
    return $p_1, \ldots, p_{|V|}$
```

Algorithm 4: Wasserstein propagation via Bregman projection.

3.6.3 Wasserstein Propagation

Generalizing the barycenter problem, we consider the "Wasserstein propagation" problem posed in more detail in Chapter 5. Suppose $G = (V, E)$ is a graph with edge weights $\alpha : E \to \mathbb{R}_+$; take $|V| = m$. With each vertex $v \in V$, we associate a label $\mu_v \in \mathrm{Prob}(M)$, whose value is a distribution over another domain M. Given fixed labels μ_v on a subset of vertices $V_0 \subseteq V$, we interpolate to the remaining vertices in $V \backslash V_0$ by solving

$$\min_{(\mu_i)_{i=1}^m} \sum_{(v,w) \in E} \alpha_{(v,w)} \mathcal{W}_2^2(\mu_v, \mu_w),$$

subject to the constraint that μ_v is fixed for all $v \in V_0$.

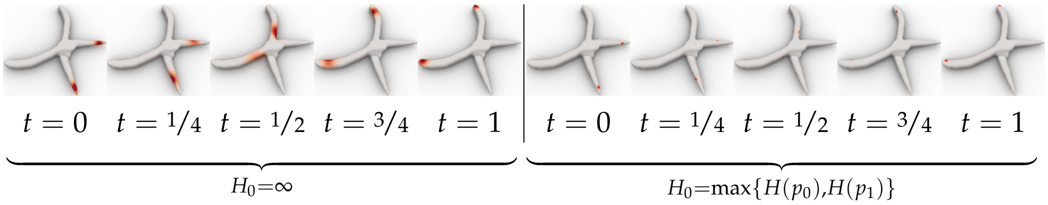


Figure 3.4: Displacement interpolation without (left) and with (right) entropy limits. The optimization implicitly matches the two peaks at $t = 0$ and $t = 1$ and moves mass smoothly from one distribution to the other.

As an example, suppose we are given two meshes and wish to find a map from vertices of one to vertices of the other. We can relax this problem by instead constructing maps to probability distributions *over* vertices of the second mesh [SNB+12]. Given ground-truth correspondences for a few vertices, the optimization above fills in missing data.

Propagation can be modeled using convolutional distances as

$$\begin{aligned} \min_{p_v} \quad & \textstyle\sum_{(v,w)\in E} \alpha_{(v,w)} \mathcal{W}^2_{2,H_t}(p_v, p_w) \\ \text{s.t.} \quad & p_v \text{ fixed } \forall v \in V_0. \end{aligned} \tag{3.17}$$

Following the optimizations in previous sections, we instead optimize over transportation matrices π_e for each $e \in E$:

$$\begin{aligned} \min_{\pi_e} \quad & \textstyle\sum_{e\in E} \alpha_{(v,w)} \mathrm{KL}(\pi_e | H_t) \\ \text{s.t.} \quad & \pi_e a = p_v \ \forall e = (v, w) \\ & \pi_e^\top a = p_w \ \forall e = (v, w) \\ & p_v \text{ fixed } \forall v \in V_0. \end{aligned}$$

The interpolated p's will be distributions because they must have the same integrals as the p's in V_0. Algorithm 4 uses iterated Bregman projection to solve this problem by iterating over one vertex in V at a time, projecting onto constraints fixing all marginals for that vertex. Applying Propositions 5 and 6, we can write $\pi_e = D_{v_e} H_t D_{w_e}$ and update the v_e's and w_e's using simple rules.

Propagation encapsulates many other optimizations in Wasserstein space. Figure 3.5 illustrates two examples. The convolutional barycenter problem (§3.6.1) is exactly propagation where G is a star graph, with vertices in V_0 on the spokes and the unknown distribution p associated with the center. An alternative model for displacement interpolation

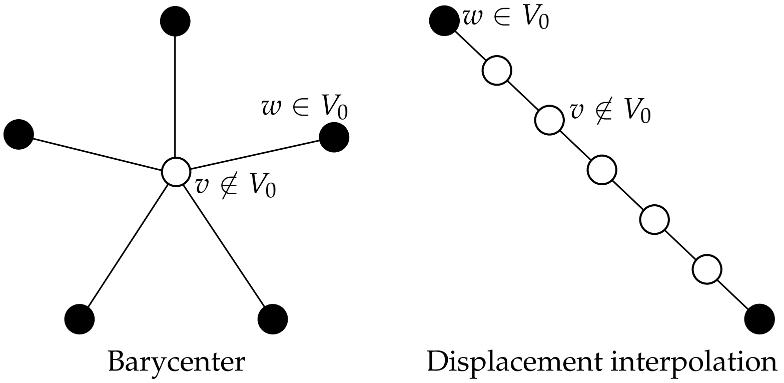


Figure 3.5: Wasserstein propagation can be used to model barycenter problems and displacement interpolation. Here, we show the corresponding graph $G = (V, E)$; vertices in V_0 have solid shading.

(§3.6.2) discretizes $t \in [0, 1]$ as a line graph, with two vertices in V_0 at the ends of the interval. This model is different from (3.15), which *directly* predicts the interpolation result at time t rather than computing the entire interpolation simultaneously.

3.7 Applications

Equipped with the machinery of convolutional transportation, we now describe several graphics applications directly benefiting from these distances and related optimization problems.

Shape interpolation. A straightforward application of Wasserstein barycenters is shape interpolation. We represent k shapes $(S_i)_{i=1}^k \subset [-1, 1]^2$ using normalized indicator functions $(\chi(S_i)/\mathrm{vol}(S_i))_{i=1}^k \in \mathrm{Prob}([-1, 1]^2)$. Given weights $(\alpha_i)_{i=1}^k$, we compute the approximate indicator function of an averaged shape as the minimizer $\mu \in \mathrm{Prob}([-1, 1]^2)$ of $\sum_i \alpha_i \mathcal{W}_{2,H_t}^2(\mu, \chi_i)$; this indicator easily can be sharpened if a true binary function is desired.

Figure 3.6 shows barycenters between four shapes with bilinear weights. Unlike the mean $\sum_i \alpha_i \chi_i(S_i)/\mathrm{vol}(S_i)$, shapes obtained using Wasserstein machinery smoothly transition between the inputs, creating plausible intermediate shapes. Figure 3.7 provides a 1D interpolation example, with simple post-processing (thresholding and coloring) to recover boundaries. Figures 3.8, 3.9, and 3.10 show analogous examples in three dimensions. We represent a surface volumetrically using the normalized indicator function of its interior.

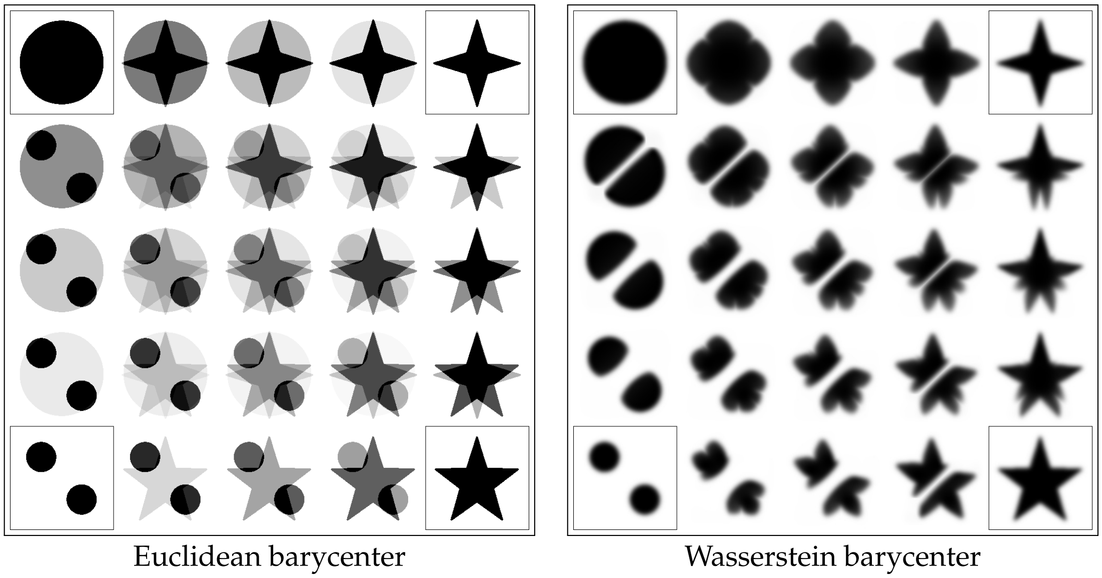


Figure 3.6: Interpolating indicators using linear combinations (left) is ineffective for shape interpolation, but convolutional Wasserstein barycenters (right) move features by matching mass of the underlying distributions.

We interpolate the resulting distributions using convolutional barycenters and extract the level set corresponding to the half the maximum probability value. This volumetric approach can handle topological changes, e.g. interpolating between a shape with two components (lower left) and three singly-connected shapes (remainder).

BRDF design. The BRDF $f(\omega_i, \omega_o)$ of a material defines how much light it reflects from each incoming direction ω_i to each outgoing direction ω_o. If ω_i is fixed, all the outgoing directions fall on a hemisphere defined by the surface normal. After scaling, the BRDF values for ω_o form a probability distribution over the hemisphere. Hence, displacement interpolation can be applied to interpolate between materials, as in [BvdPPH11].

We use convolutional barycenters to combine more than two distributions at a time. For each incoming direction in the sampled BRDF, the values associated to the outgoing directions are organized in a 2D grid by spherical angle. We use weighted Wasserstein barycenters to interpolate this data. The spherical heat kernel H_t is approximated by the fast approximate Gaussian convolution from [Der93]. Spherical geometry is accounted for by modulating the width of this separable filter. We render images using the interpolated BRDFs using PBRT [PH10].

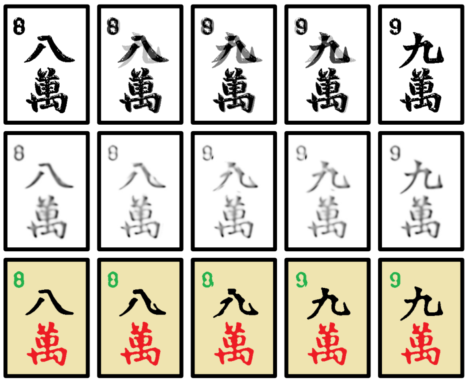


Figure 3.7: “Generalized Mahjong:” Linear (top) and displacement (middle) interpolation between two images; while it is less sharp, the displacement interpolation result can be post-processed using simple image filters to generate a nontrivial interpolation (bottom; see e.g. the tip of the “9” character rotating outward).

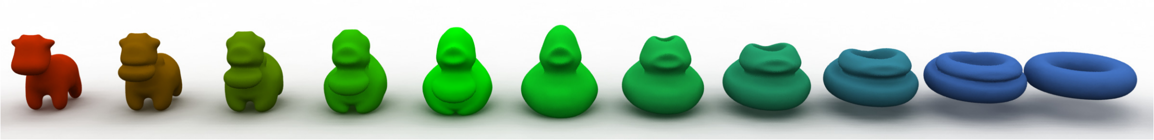


Figure 3.8: Shape interpolation from a cow to a duck to a torus via convolutional Wasserstein barycenters on a $100 \times 100 \times 100$ grid, using the method at the beginning of §3.7.

Figure 3.11 shows interpolation between four BRDFs using our technique, yielding continuously-moving highlights. The corner BRDFs are sampled from closed-form materials [Bli77, AS00]; the remaining BRDFs are interpolated.

Color histogram manipulation. In image processing, optimal transportation has proven useful for color palette manipulations like contrast adjustment [Del06] and color transfer [PKD07] via 1D transportation. Previous methods for this task avoid carrying out multi-dimensional transport, e.g. using 1D sliced approximations or cumulative axis-aligned transport [PKD07, BRPP14, PPC11] or can support only coarse histograms [FPPA14]. Convolutional transport, however, handles large-scale 2D chrominance histograms directly.

We transfer color over the CIE-Lab domain by modifying the one-dimensional L (luminance) and two-dimensional ab (chrominance) channels independently, where luminance takes values in $[1, 100]$ and chrominance takes values in $M = [-128, 128]^2$. Remapping L

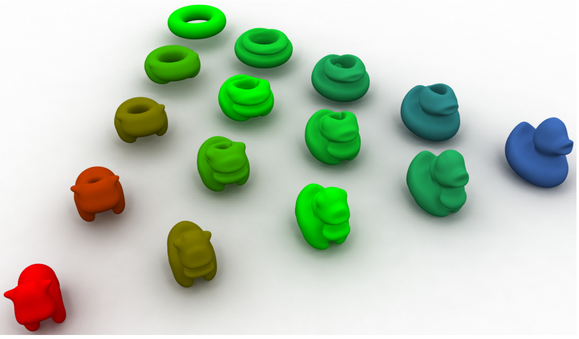


Figure 3.9: Shape interpolation in 3D, expanded from Figure 3.8.

requires 1D transport, which is computable in closed form [Vil03]; we describe the processing of the ab channel below.

Suppose we express the ab components of k images as a set of functions $(f_i)_{i=1}^k$, where $f_i : [0,1]^2 \to M$ takes a point on the image plane and returns an ab chrominance value. The chrominance histogram μ_i associated to f_i is the push-forward of the uniform measure $\mathcal{U}$ on $[0,1]^2$ by the map f_i, satisfying $\mu_i(A) = \mathcal{U}(f_i^{-1}(A))$ for $A \subset M$. It is approximated numerically by a discrete histogram p_i on an uniform rectangular grid over M.

For a given set of weights $\alpha \in \mathbb{R}_+^k$, we solve the barycenter problem (3.12) using Algorithm 2. This provides the weighted barycenter $\mu \in \mathrm{Prob}(M)$, discretized as a vector p. The algorithm furthermore provides the scaling factors (v_i, w_i) for each $i = 1, \ldots, k$, which define the transport maps $\pi_i = D_{v_i} K D_{w_i}$ between each input histogram p_i and the barycenter p. This discrete coupling π_i should be understood as a discretization of a continuous coupling $\pi_i(x, y)$ between each μ_i and μ.

For each i, we introduce a map $T_i : M \to M$, defined on the support of μ_i (i.e. the set of $x \in M$ such that $\mu_i(x) > 0$), by

$$\forall x \in M, \quad T_i(x) = \tfrac{1}{\mu_i(x)} \int_M \pi_i(x,y) y \, dy.$$

This integral is computed numerically as a sum over the grid, where π_i is used in place of π_i.

The rationale behind this definition is that as $\gamma \to 0$, the regularized coupling π_i converges to a measure supported on the graph of the optimal matching between μ_i and the barycenter; this phenomenon is highlighted in Figure 3.1. Thus, as $\gamma \to 0$, T_i converges to the optimal transport map. It can thus be used to define a corrected image $f_i^\alpha \overset{\text{def.}}{=} T_i \circ f_i$

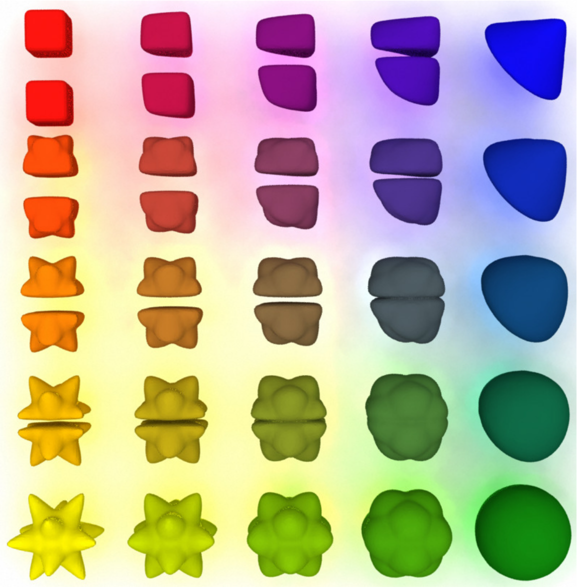
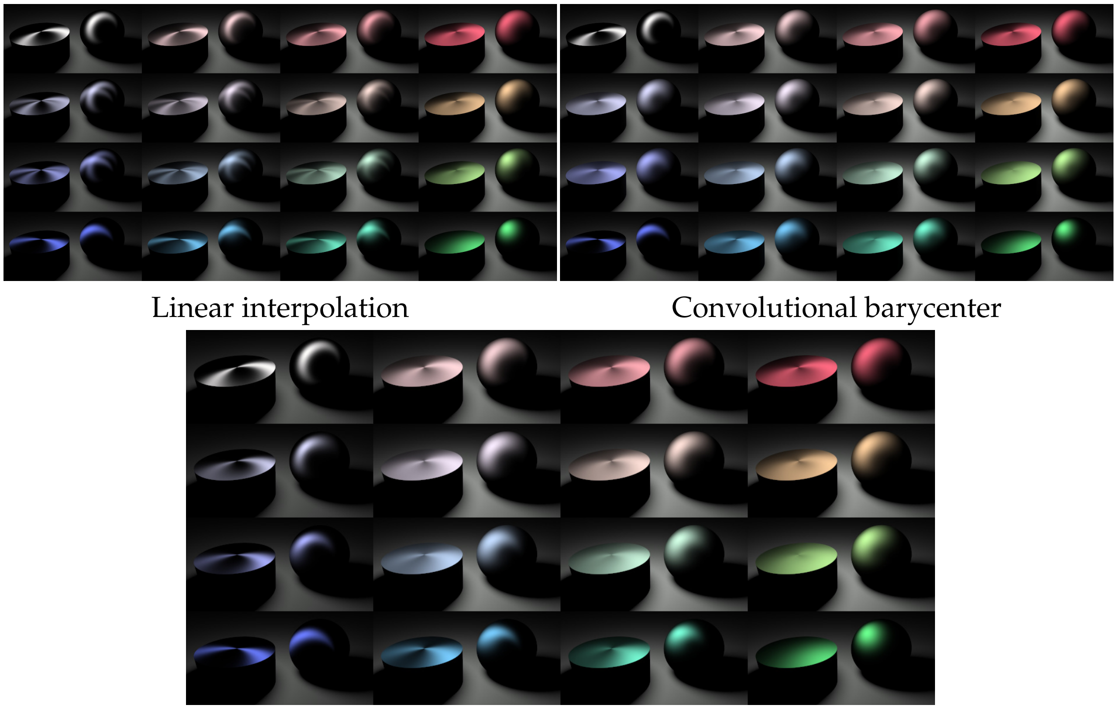


Figure 3.10: Three-dimensional shape interpolation. The four corner shapes are represented using normalized indicator functions on a 60×60×60 volumetric grid; barycenters of the distributions are computed using bilinear weights.

whose chrominance histogram matches μ. Figure 3.12 shows an application of the method to $k = 2$ input images.

Skeleton layout. Suppose we are given a triangle mesh $M \subset \mathbb{R}^3$ and a skeleton graph $G = (V, E)$ representing the topology of its interior. For instance, if M is a human body shape, then G might have "stick figure" topology. To relate G directly to the geometry of M, we might wish to find a map $V \mapsto \mathbb{R}^3$ embedding the vertices of the graph into the interior of the surface.

We can approach this problem using Wasserstein propagation (§5.2.2). We take as input the positions of vertices in a small subset $V_0 \subseteq V$. As suggested in §2.6, we express the position of each $v \in V_0$ as a distribution $p_v \in \text{Prob}(M)$ using barycentric coordinates computed using the algorithm by Ju et al. [JSW05]. Distributions $p_v \in \text{Prob}(M)$ can be interpolated along G to the remaining $v \notin V_0$ via Wasserstein propagation with uniform edge weights. The computed p_v's serve as barycentric coordinates to embed the unlabeled vertices. Thanks to displacement interpolation, the constructed embedding conforms to the geometry of the surface; Figure 3.13 shows sample embeddings generated using this



Linear interpolation

Convolutional barycenter

Convolutional barycenter (bounded entropy)

Figure 3.11: BRDF interpolation: BRDFs for the materials in the four corners of each image are fixed, and the rest are computed using bilinear weights. Linearly interpolating BRDFs (left) yields spurious highlights, the convolutional barycenter (center) moves highlights continuously but increases diffusion, and the entropy-bounded barycenter (right) moves highlights in a sharper fashion.

strategy.

Soft maps. A relaxation of the point-to-point correspondence problem replaces the unknown from a map $\phi : M_0 \to M$ to a measure-valued map $\mu_x : M_0 \to \text{Prob}(M)$. [SGB13] generalizes the Dirichlet energy of a map to the measure-valued case, but their discussion is limited to *analysis* rather than *computation* of maps because their discretization scales poorly.

Suppose M_0 and M are triangle meshes and H_t is the heat kernel matrix of M. A regularized discretization of the measure-valued map Dirichlet energy is provided by the Wasserstein propagation objective (3.17) from M_0 viewed as a graph $M_0 = (V, E)$ to distributions on M, with weights proportional to inverse squared edge lengths. Coupled with pointwise constraints, Algorithm 4 provides a way to recover a map minimizing the

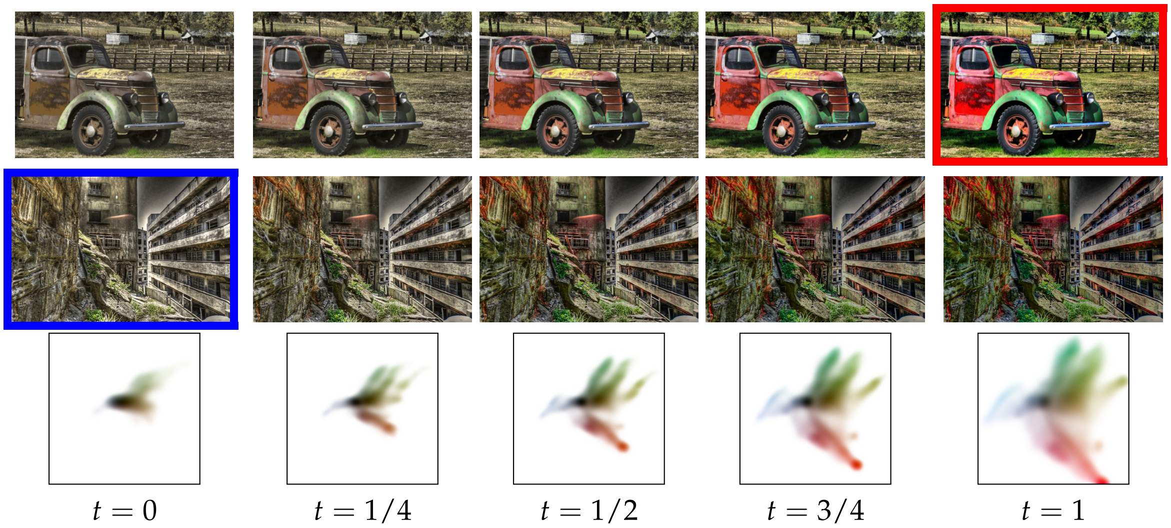


Figure 3.12: Color transfer with 2D convolutional transportation over the chrominance space. Top row: evolution of the color-corrected image f_1^α as a function of $\alpha = (1-t,t)$. Middle row: evolution of f_2^α. The red (resp. blue) framed image shows the input f_1 (resp. f_2) which is obtained for $t = 0$ (resp. $t = 1$). Bottom row: barycenter histogram μ as a function of t; colors encode the corresponding position x over the (a,b) domain while luminance corresponds to the amplitude of $\mu(x)$ (zero being white).

resulting energy; convergence can be slow, however, when the constraints are far apart.

To relax dependence on pointwise constraints and accelerate convergence, we introduce a *compatibility function* $c(x,y) : M_0 \times M \to \mathbb{R}_+$ expressing the geometric compatibility of $x \in M_0$ and $y \in M$; small $c(x,y)$ indicates that the geometry of M_0 near x is similar to that of M near y. Discretely, take c_v to sample the compatibility function $c(v,\cdot)$ on M associated with $v \in M_0$. We modify the objective (3.17) as follows:

$$\left[\sum_{(v,w)\in E} \frac{1}{\ell^2_{(v,w)}} \mathcal{W}^2_{2,H_t}(p_v,p_w)\right] + \tau \left[\sum_{v\in V} \omega_v a^\top (p_v \otimes c_v)\right]. \tag{3.18}$$

This objective favors distributions p_v with low compatibility cost; the weight ω_v is the area weight of $v \in M_0$.

Take $N(v)$ to be the valence of $v \in V$. In terms of transportation plans, (3.18) equals $\sum_{(v,w)\in E} \mathcal{W}^2_{2,\overline{H}_t}(p_v,p_w)/\ell^2_{(v,w)}$, where

$$\overline{H}_t \overset{\text{def.}}{=} \text{diag}\left[\exp\left(-\frac{\ell_e^2\tau\omega_v c_v}{\gamma N(v)}\right)\right] H_t \text{diag}\left[\exp\left(-\frac{\ell_e^2\tau\omega_w c_w}{\gamma N(w)}\right)\right].$$

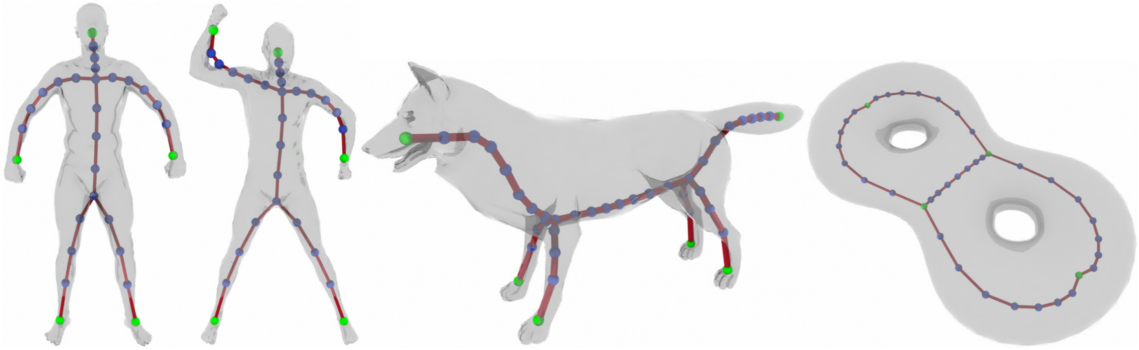


Figure 3.13: Embeddings of skeletons computed using Wasserstein propagation; the positions of the blue vertices are computed automatically using the fixed green vertices and topology of the graph.

This matrix is a diagonal rescaling of H_t, so we can still efficiently optimize (3.18) using Algorithm 4, slightly adjusted to use a different kernel on each edge.

Figure 3.14 shows maps between a pair of surfaces computed using this technique. Because the models are nearly isometric, we use the wave kernel signature (WKS) [ASC11] to determine the compatibility function $c(x,y)$. This signature is unable to distinguish between the orientation-preserving and left/right flipped maps between the two surfaces. Wasserstein propagation guided by this choice of $c(x,y)$ paired with a sparse set of fixed correspondences breaking the symmetry is enough to recover both maps. The resulting soft map matrices are of size 1024×1024, an order of magnitude larger than the maps generated in [SNB$^+$12], computed in less than a minute using similar hardware.

3.8 Discussion and Conclusion

Although optimal transportation has long been an attractive potential technique for graphics applications, optimization challenges hampered efforts to include it as part of the standard toolbox. Convolutional Wasserstein distances comprise a large step toward closing the gap between theory and practice. They are easily computable via the heat kernel—a well-studied and widely-implemented operator in graphics—and through the iterated projection algorithm can be incorporated into modeling problems with transportation terms.

We have demonstrated the breadth of applications enabled by this framework, from rendering to image processing to geometry. Modeling via probability distributions is natural for these and other problems, and we foresee applications across several additional

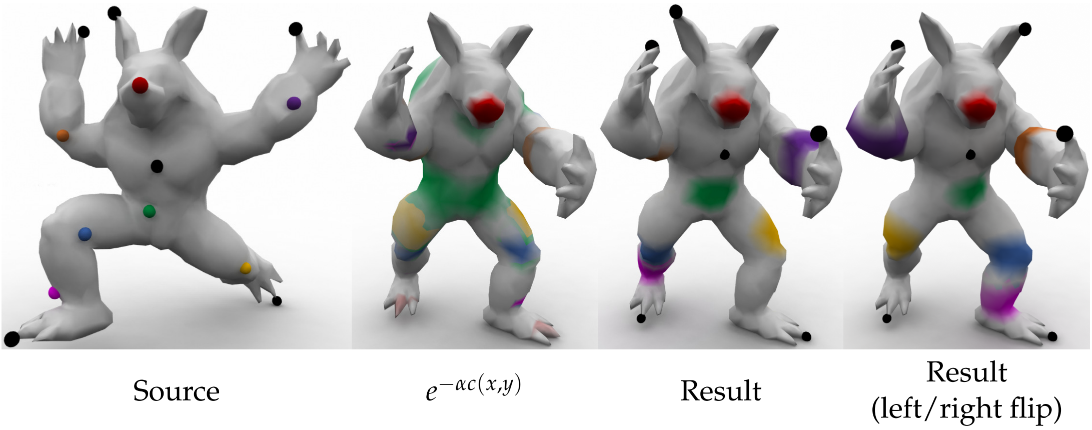


Figure 3.14: Soft maps: Colored points on the source are mapped to the colored distributions on the target, where black points are fixed input correspondences. Our method is able to extract *two* maps from the left-right symmetric descriptor $c(x, y)$, depending on whether the fixed correspondences preserve orientation or are flipped.

disciplines. Having reduced the cost of experimenting with transportation models, future studies now may incorporate transportation into graphics applications including processing of volumetric data, caustic design, dimensionality reduction, and simulation.

Several theoretical and numerical problems remain open. The regularization in convolutional transport enables scalable computation but introduces smoothing; imaging applications like those in [ZQC+14] require sharp edges that can get lost. As it stands now, while our technique outperforms existing methods for transportation in graphics, numerics degrade if γ is too small, similar to the heat kernel approximation in [CWW13]; this is the primary drawback of our transport approximation. Modeling with “true” quadratic Wasserstein distances remains a challenge on images and triangle meshes, and large-scale discretizations of flow models proposed by Benamou and Brenier [BB00a] remain to be formulated. Closer to the current discussion, the algorithm for propagation in §5.2.2 might benefit from preconditioners spreading information non-locally in each iteration; this would alleviate the need to iterate $|V|$ times to guarantee “communication” between every pair of vertices.

Optimal transportation provides an intuitive, foundational approach to geometric problems over many domains. Practical, easily-implemented optimization tools like the ones introduced here will enable its incorporation into graphics pipelines for countless tasks.

Chapter 4

Soft Maps Between Surfaces

Having considered algorithms for approximating both one- and two-Wasserstein distances primarily on two-dimensional domains like surfaces, we now proceed to *four* dimensions and study the problem of finding a map one surface to another. The dimensionality of such a problem might be considered fundamentally higher, in the sense that we are now optimizing over matchings between pairs of two-dimensional domains; this creates a four-dimensional problem total. Even so, the optimization advantages of optimal transportation coupled with the fact that we expect even probabilistic relaxations of mappings to be sparse will lead to computationally feasible algorithms for correspondence.

4.1 Introduction

A natural problem in geometry processing is that of finding a smooth map between two surfaces. A reliable algorithm for finding such a map can be used in pipelines for texture or annotation transfer, segmentation, morphing, and surface editing, among other applications within graphics. Outside of graphics, ongoing research in vision and other fields makes use of shape maps to create links between new inputs and previously-analyzed data; for instance, a robot navigating an unknown environment may try to map objects it encounters to ones in some given database of objects it can manipulate.

Unless shapes are rigid motions or isometric deformations of each other, it is difficult to define a single "best" map between most pairs of surfaces at point-to-point granularity. This difficulty arises because there are at least two geometric sources of ambiguity complicating the mapping problem:

Global ambiguity: Symmetric shapes may admit multiple geometrically equivalent maps; for instance, human models often have left-right symmetries that generate two equivalent maps in terms of the amount of geodesic distance distortion they induce. Note that the shapes might be symmetric under a rigid transformation of space or under an *intrinsic* symmetry (as in e.g. [OSG08, RBBK10]).

Local ambiguity: Shapes that are not exact isometric deformations of one another may admit an informative map at some level of coarseness but not at the point-to-point level due to scaling, slippage, or the absence of identical details. For instance, generating maps between a horse and a dog model makes sense at the segment level because both animals have similar limb structures, even if the structures within those limbs are different.

Additional ambiguities can result from a lack of context. Without knowledge of the process used to obtain the target from the source, it is impossible to know which maps are semantically relevant, regardless of geometric cues.

Given these fundamental problems, a limitation of many mapping algorithms is that they attempt to find a point-to-point map with no more than a geometric prior, whether it be rigidity, conformality, isometry, elasticity, or otherwise. These methods are forced to make somewhat arbitrary decisions as to the user's desired map or, worse, unsuccessfully attempt to combine often disjoint acceptable maps. Thus, mapping algorithms that process a variety of shapes should incorporate uncertainty when the mapping problem is itself ambiguous. These ambiguities can be resolved with domain-specific knowledge, semantic information, or other cues, or they can be used to prompt the user for guidance.

In this chapter, we propose *soft maps*, which generalize point-to-point maps by embracing uncertainty as a fundamental part of the mapping process. In this setup, maps are expressed as conditional distributions of a distribution on the product of the two surfaces, which we call a *soft correspondence*. In other words, we attach to each pair of regions on the two surfaces a probability indicating the likelihood that these regions should be mapped to one another. Soft maps and correspondences are easily discretized as matrices of probability values. This representation is amenable to convex optimization techniques, while still allowing for representation of point-to-point maps as permutation matrices.

The soft map framework is capable of handling both local and global ambiguities, as illustrated in Figure 4.1. Here, given only geometric information, we can compute a soft

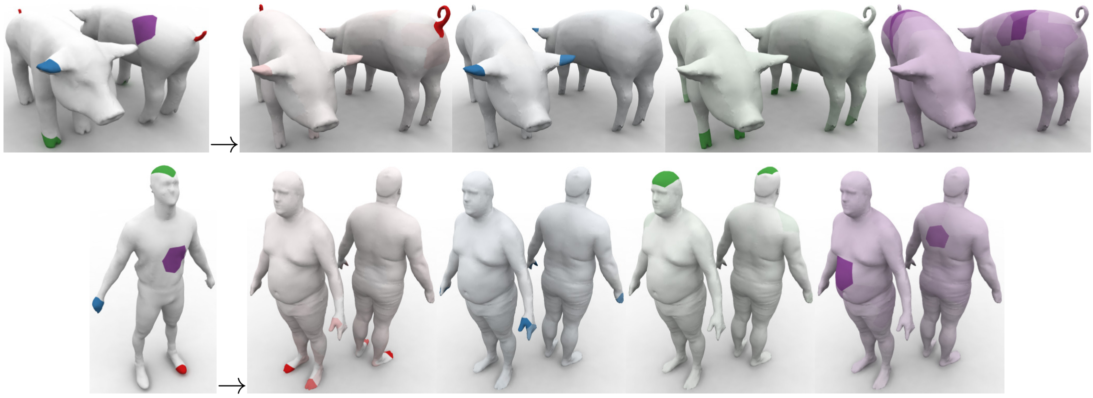


Figure 4.1: Soft maps from one model to another. The colored patches on the leftmost model are mapped to the colored distributions over the models on the right. These soft maps acknowledge discrete left-right and front-back symmetries as well as localized ambiguities including slippage along the pig's back.

mapping where the front hoof of a pig model is mapped $\sim$50% to each of the front hooves on a different pig; the back of the source pig is mapped to a larger region on the target pig's back, since the lack of distinguishing geometric features makes a more precise mapping impossible. Similarly, the map in Figure 4.1 between human models acknowledges their approximate left-right and front-back symmetries.

One important property of soft maps is *continuity*, which must be redefined probabilistically to ensure that nearby points on one surface yield nearby distributions on the other. We define infinitesimal notions of soft map continuity and show how Wasserstein distances can measure the discrete continuity of a soft map via a relaxation of the classical Dirichlet energy of a differentiable point-to-point map. With this metric and others describing a soft map's alignment with geometric features and bijectivity, we provide a convex optimization framework for analyzing and computing soft maps.

4.2 Related Work

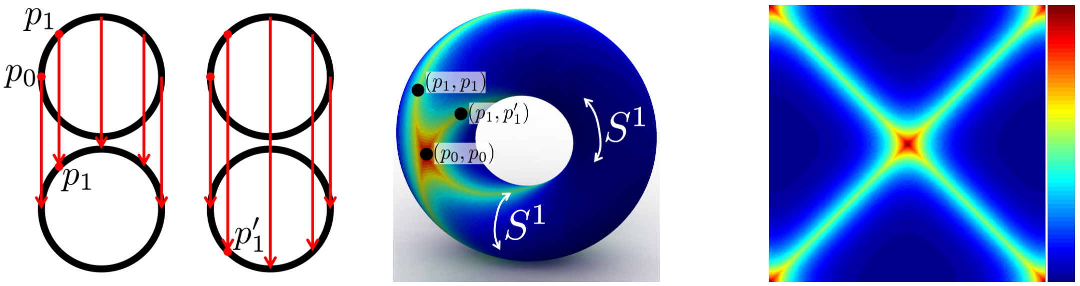
The literature on mapping between surfaces is vast, and we refer the reader to [vKZHCO11, CLM$^+$11, BBK08] for general summaries of previous work. The idea of computing a mapping by minimizing descriptor distances and preserving continuity is common to many of the works surveyed here.

Recent work on mapping reveals several approaches incorporating geometric cues and matching strategies. [BBK06] embeds one surface into another using Generalized Multidimensional Scaling to minimize distortion. [LF09, KLCF10, KLF11] explore and combine maps from the group of Möbius transforms of a surface, which can be constructed efficiently and include isometries. [GBAL09, SOG09] introduces the *heat kernel signature* (HKS), assigning a pointwise signature based on heat flow, and [OMMG10] uses a related technique to find maps with guaranteed behavior for nearly-isometric surfaces. The HKS construction is applied to the wave equation in [ASC11] for experimentally more informative signatures. The algorithms in these and other papers find a *single* sparse or full map, whereas our new *representation* of a map can encode multiple correspondences.

The idea of a "fuzzy map" in terms of probability matrices was introduced in [WL78] using simple heuristics to construct and update maps. More recently, the Möbius transformations sampled in [LF09] generate a "fuzzy correspondence matrix" guiding point-to-point matching. Fuzzy schemes are also used to relax point-to-point mappings as in [BBM05, RCB97]. A probabilistic approach to mapping is taken in [ASP$^+$04, TBW$^+$11], though here distributions are *over* non-soft point-to-point mappings and thus subject to the rigidity of point-to-point schemes. Our optimization for finding soft maps has commonalities with theirs since their energy can be separated into unitary and binary terms, although ours is convex and thus not prone to local minima.

Some existing approaches use convex optimizations that are related to ours. The continuous relaxation of the integer program in [WSSC11] could be viewed as a soft map, although the output is harder to interpret. The relaxation of the graph isomorphism problem in [SU97] provides some analogous constructions to the constraints on and desired properties of soft maps for graphs; a related construction on hypergraphs is provided in [ZS08]. Wasserstein distances also have been applied to optimizations for several related vision and geometry problems. For instance, [LD11] uses them to construct a distance metric between surfaces invariant to Möbius transformations, and [HZTA04] uses them to guide image registration.

The work closest related to soft mapping, however, is that on measure couplings and Gromov-Wasserstein distances [Mém07, Mém09, Mém11]. Here, correspondences are discretizations of measure couplings, or probability distributions over the product of two surfaces whose marginalizations to the surfaces yield areas. The method is only acceptable for nearly-isometric surfaces admitting area-preserving correspondences. Furthermore, the optimization problem for finding measure couplings is non-convex with multiple local



(a) Two maps $S^1 \mapsto S^1$ (b) Soft correspondence (c) Soft correspondence matrix A

Figure 4.2: (a) Two maps from S^1 to itself, (b) a soft correspondence superposing the two maps on the torus $S^1 \times S^1$, and (c) the corresponding matrix A.

optima when either surface is symmetric.

Applications of soft maps overlap significantly with those of point-to-point mapping. Some overlapping applications are better suited to the probabilistic context. For instance, annotation transfer and other tasks operating on shapes at a coarse level can use soft maps directly. Additionally, since soft maps can encode multiple point-to-point maps, methods like [NBCW+11] for finding consistent maps within a collection may have a higher chance of success.

4.3 Definitions

Let M_0 and M be two surfaces embedded in $\mathbb{R}^3$; we will assume the surfaces are rescaled so that they both have area 1. We can view $M_0 \times M$ as a four-dimensional manifold.

The basic object we consider is a probability measure $P \in \text{Prob}(M_0 \times M)$, which we call a *soft correspondence* between M_0 and M. We view $P(U \times V)$ as the probability that a pair of points $p_0 \in U \subseteq M_0$ and $p \in V \subseteq M$ are related to each other. With this interpretation, the *uniform distribution* indicates a mapping in which all pairs of points (p_0, p) are deemed equally likely to be related, while the relationship $y = \phi(x)$ induced by the mapping $\phi : M_0 \to M$ is encoded by a δ-measure whose support is the surface $\{(x, \phi(x)) : x \in M_0\} \subseteq M_0 \times M$.

It can be difficult to visualize distributions on the four-dimensional product $M_0 \times M$. If M_0 and M are curves, however, the *two*-dimensional product $M_0 \times M$ can be visualized on the plane or using a toroidal topology. We thus show an example in Figure 4.2 of a soft correspondence between the circle S^1 and itself on the torus $S^1 \times S^1$. This distribution represents a convex combination of two point-to-point maps, illustrating the expressive power of soft maps.

Given a soft correspondence P between M_0 and M, we use conditional probabilities to derive *soft maps*. A soft map from M_0 to M is a function $\mu : x \mapsto \mu_x$ assigning a probability measure $\mu_x \in \text{Prob}(M)$ to each $x \in M_0$. Thus, if $\mathcal{U} \subseteq M$, we interpret the value $\mu_x(\mathcal{U})$ as the probability that a randomly sampled $y \in \mathcal{U}$ corresponds to x. In this way, correspondences can be expressed with a degree of uncertainty. For all $x \in M_0$ and $\mathcal{U} \subseteq M$, we require that $\mu_x(\mathcal{U}) \geq 0$ and $\mu_x(M) = 1$.

Soft maps generalize conventional point-to-point maps between surfaces. In particular, every map $\phi : M_0 \to M$ yields a soft map μ by requiring that $\mu_x(\mathcal{U}) = 1$ if and only if $\phi(x) \in \mathcal{U}$. That is, μ_x is a unit Dirac mass centered at $\phi(x)$.

Soft maps can encode a wider variety of mapping behavior than conventional maps. For example, the precise location of the point corresponding to $x \in M_0$ might not be known. Then, μ_x would have a peak at $\phi(x)$ with nonzero width representing uncertainty in the location of $\phi(x)$. If M admits self-symmetries, then x might correspond equally well to multiple points on M. Then, μ_x would be the convex combination of two or more peaked probability measures.

A second advantage of soft maps is that they can be represented by positive scalar functions, i.e. their *densities*. The density of the soft map μ is the function $\rho : M_0 \times M \to \mathbb{R}_+$ satisfying

$$\mu_x(\mathcal{U}) = \int_{\mathcal{U}} \rho(x, y)\, dy$$

for all $x \in M_0$ and $\mathcal{U} \subseteq M$. Here, dy is the area measure of M. Note $\int_M \rho(x, y)\, dy = 1$ must hold for all $x \in M_0$. Henceforth, we will use the abbreviation $d\mu_x(y) \overset{\text{def.}}{=} \rho(x, y)\, dy$. This scalar function representation makes it straightforward to discretize soft maps.

REMARK: Recall that not all probability distributions admit densities. But we consider only distributions that are at least *weak limits* of smooth densities. Our analysis remains valid with this assumption.

4.4 Variational Analysis of Soft Maps

A typical approach to optimizing for a soft map or analyzing a given soft map would be to minimize a convex potential measuring desirable properties like smoothness, bijectivity, and preservation of geometric features. In this section, we outline some functional quantifying these properties given a soft map $x \mapsto \mu_x$.

4.4.1 Dirichlet Energy

Let μ be a soft map from M_0 to M. We would like to quantify the degree of smoothness of the function $x \mapsto \mu_x$ in the x-variable. To do so, we construct a *Dirichlet energy* for soft maps in line with the general framework suggested in [Jos94] for Dirichlet energies in metric spaces. We choose a distance metric for $\mathrm{Prob}(M)$, namely the 2-Wasserstein distance $\mathcal{W}_2 : \mathrm{Prob}(M) \times \mathrm{Prob}(M) \to \mathbb{R}_+$.

To this end, we define the *Dirichlet energy density* of a soft map μ from M_0 to M at $x \in M_0$ as

$$e_\mu(x) \overset{\text{def.}}{=} \lim_{r\to 0} \frac{1}{\mathrm{Area}(B_r(x))} \int_{B_r(x)} \left[\frac{\mathcal{W}_2(\mu_x, \mu_{x'})}{\mathrm{dist}_0(x, x')} \right]^2 dx' \tag{4.1}$$

where dist_0 is the geodesic distance function on M_0. The *Dirichlet energy* of μ is then

$$\mathcal{E}_D(\mu) \overset{\text{def.}}{=} \frac{1}{\mathrm{Area}(M_0)} \int_{M_0} e_\mu(x)\, dx\,. \tag{4.2}$$

We will appeal to three important properties of the Dirichlet energy for soft maps. We state the first two here and give a more thorough discussion of the key third property in the next section.

Proposition 8. *The behavior of the Dirichlet energy in two special limiting cases is as follows.*

1. *Let μ be a soft map from M_0 to M with constant density, i.e. $\rho(x, y) = \rho(y)$ for all $x \in M_0$. Then $\mathcal{E}_D(\mu) = 0$.*

2. *Let $\phi : M_0 \to M$ be a map. The Dirichlet energy of the associated soft map equals the Dirichlet energy of ϕ.*

Proof. If μ is a soft map from M_0 to M with constant density then $\mu_x = \mu_{x'}$ for all $x, x' \in M_0$. Thus $\mathcal{W}_2(\mu_x, \mu_{x'}) = 0$ because the product distribution $\pi \overset{\text{def.}}{=} \mu_x \otimes \mu_{x'}$ is a measure coupling with zero cost. Next, if $\phi : M_0 \to M$ is a map, then the associated soft map is $\delta_{\phi(x)}(y)\, dy$ where δ_p is the Dirac δ-density centered at p. There is only on measure coupling of $\delta_{\phi(x)}(y)\, dy$ and $\delta_{\phi(x')}(y)\, dy$, namely the product distribution for which the cost is $\mathrm{dist}(\phi(x), \phi(x'))$. The Dirichlet energy density now reduces to the conventional Dirichlet energy density in the limit as $r \to 0$. □

Proposition 9. *The Dirichlet energy is convex under linear combination. That is, if μ_1 and μ_2 are soft maps from M_0 to M and $\alpha \in [0,1]$, then*

$$\mathcal{E}_D((1-\alpha)\mu_1 + \alpha\mu_2) \leq (1-\alpha)\mathcal{E}_D(\mu_1) + \alpha\mathcal{E}_D(\mu_2)\,.$$

Proof. Let μ_1, μ_2 be soft maps, π_k for $k = 1, 2$ be optimal measure couplings of $[\mu_k]_x$, $[\mu_k]_{x'}$, and $\alpha \in [0,1]$. Then $\pi_\alpha \overset{\text{def.}}{=} (1-\alpha)\pi_1 + \alpha\pi_2$ is a measure coupling of $[(1-\alpha)\mu_1 + \alpha\mu_2]_x$ and $[(1-\alpha)\mu_1 + \alpha\mu_2]_{x'}$. Since $\iint_{M\times M}[\text{dist}(y,\bar{y})]^2 d\pi_\alpha(y,\bar{y}) = (1-\alpha)W_2^2([\mu_1]_x, [\mu_1]_{x'}) + \alpha W_2^2([\mu_2]_x, [\mu_2]_{x'})$, the proposition follows. □

The expression in (4.1) for the Dirichlet energy density is unwieldy and does not adapt well to a discrete setting. In particular, the infimum introduces many auxiliary variables for representing measure couplings. Therefore, the discretization of the Dirichlet energy in this form scales poorly with problem size.

By exploiting the properties of the 2-Wasserstein distance, however, we can simplify the Dirichlet energy density (4.1) into a form that enables a tractable discretization. To do so, we must introduce a new mathematical object, which we call the *transportation potential* and denote by Q. This object takes as input a point $x \in M_0$ and a tangent vector $V \in T_x M_0$ and outputs a function on M, with linear dependence on V. Given (x, V) we write $y \mapsto Q(x,y)\cdot V$ for the output function.

Proposition 10. *Let $d\mu_x(y) \overset{\text{def.}}{=} \rho(x,y)dy$ be a soft map from M_0 to M satisfying a suitable regularity condition. Then its Dirichlet energy satisfies*

$$\mathcal{E}_D(\mu) = \int_{M_0}\int_M \rho(x,y)\|\nabla Q(x,y)\|^2\,dy\,dx\,, \tag{4.3}$$

where Q is the transportation potential of μ. It is found by solving the partial differential equation

$$\begin{aligned} \nabla\cdot\big(\rho(x,y)\nabla Q(x,y)\cdot V\big) &= -\langle\nabla_0\rho(x,y), V\rangle \\ \int_M \rho(x,y)Q(x,y)\cdot V\,dy &= 0 \end{aligned} \tag{4.4}$$

for every $x \in M_0$ and $V \in T_x M_0$.

Proof. Assume first that $\rho > 0$. We recall from the theory of optimal transportation that the solution of the optimal transportation problem for the 2-Wasserstein distance on a compact surface can be characterized as follows. The transport between two measures with positive density μ_1 and μ_2 is achieved by a map $\psi : M \to M$ of the form $\psi(y) \overset{\text{def.}}{=} \exp_y(\nabla q(y))$ where $q : M \to \mathbb{R}$ is a convex function and $\exp_y : T_y M \to M$ is the geodesic exponential map. Moreover, the transportation cost can be expressed as $[\mathcal{W}_2(\mu_1, \mu_2)]^2 = \int_M \|\nabla q\|^2 d\mu_1$. See [Vil03, Ch. 2] for details. We can use these ideas to simplify the Dirichlet density (4.1) by setting $\mu_1 \overset{\text{def.}}{=} \mu_x$ and $\mu_2 \overset{\text{def.}}{=} \mu_{x'}$, yielding $q \overset{\text{def.}}{=} q_{x,x'}$ that achieves the transport from μ_x to $\mu_{x'}$. When x and x' are sufficiently close, we can write $x' = \exp_x(\varepsilon V)$ where $\varepsilon = \text{dist}_0(x, x')$ and $V \in T_x M_0$. To first order, $q_{x,x'}(y) \approx \varepsilon Q(x, V, y)$ where $y \mapsto Q(x, V, y)$ is a function on M. Also Q is linear V. Substituting q into the expression for the cost given above and differentiating in ε leads to the PDE (4.4) for each $x \in M_0$ and $V \in T_x M_0$. Finally, taking the limit as $\varepsilon \to 0$ in the expression for the Dirichlet energy density (4.1) yields the desired simplification. Limiting arguments can be made to handle the cases of densities which fail to be everywhere non-zero and weak limits of smooth densities. □

REMARK: The differential operator in (4.4) is linear and has the constant functions in its kernel. The second equation in (4.4), however, ensures that solutions are transverse to the constant functions. Thus the solutions of (4.4) are unique.

Despite its involved mathematical definition, the intuition for Q is straightforward. For each $x \in M_0$, we visualize the distribution $d\mu_x(y) = \rho(x, y)\, dy$ as a collection of particles on M whose density near y is proportional to $\rho(x, y)$. If we choose a small vector $V \in T_x M_0$ and displace x to $x' \overset{\text{def.}}{=} \exp_x(V)$, we can track the motion of these particles on M as they rearrange themselves in a Wasserstein-optimal manner from μ_x to $\mu_{x'}$. The vector field $\rho(x, y) \nabla Q(x, y) \cdot V$ is the momentum of these particles as they move and the cost (4.3) is twice their kinetic energy. This interpretation aligns with the Benamou-Brenier formulation of optimal transportation in terms of fluid flow [BB00b], see also [Vil03, Ch. 8].

The partial differential equation (4.4) satisfied by Q is an anisotropic version of Poisson's equation. We can invoke standard theory when $\rho > 0$ to establish existence and uniqueness of Q. If this inequality does not hold, these properties can fail. Then, solvability of (4.4) can be restored by considering the $\varepsilon \to 0$ limit of the equation for $(1 - \varepsilon)\rho + \varepsilon \rho_0$ where $\rho_0(x, y)$ is uniform on M for all $x \in M_0$, see [AGS05, Ch. 8].

4.4.2 Bijectivity Energy

A bijectivity energy for soft maps should promote the equal distribution of probabilistic mass pushed forward under the soft map. To this end, let $d\mu_x(y) \overset{\text{def.}}{=} \rho(x,y)\,dy$ be a soft map between M_0 and M, and consider the quantity $b(y) \overset{\text{def.}}{=} \int_{M_0} \rho(x,y)\,dx$. Note that $b(y) \geq 0$ and $\int_M b(y)\,dy = 1$ so $b(y)\,dy$ is a probability measure on M. Indeed $\int_{\mathcal{U}} b(y)\,dy$ gives the probability that a randomly sampled $y \in \mathcal{U} \subseteq M$ receives mass from *somewhere* in M_0. We view b as a *bijectivity energy density* whose square integral yields the bijectivity energy, a convex quadratic function of μ.

We define the bijectivity energy of a soft map μ from M_0 to M, with $d\mu_x(y) \overset{\text{def.}}{=} \rho(x,y)dy$ as

$$\mathcal{E}_b(\mu) \overset{\text{def.}}{=} \int_M \left[\int_{M_0} \rho(x,y)\,dx\right]^2 dy\,. \tag{4.5}$$

We can see that μ has small bijectivity energy when $b(y)$ is nearly constant and $b(y)\,dy$ is as spread out as possible. Thus for such μ, most $y \in M$ receive mass from M_0 and no y receives a large amount of mass. A final desirable property of the bijectivity energy is that it has the "correct" limit for soft maps arising from conventional maps.

Proposition 11. *Suppose $\phi : M_0 \to M$ is a diffeomorphism and let μ be the associated soft map. Then*

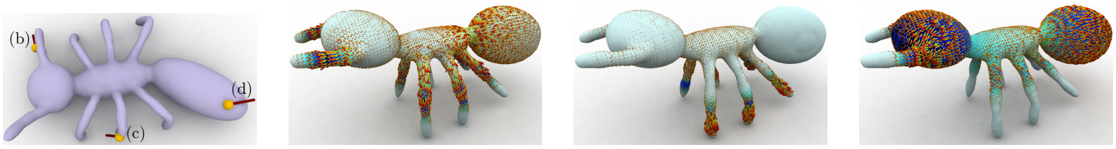
$$\mathcal{E}_b(\mu) = \int_M \left[\det(\nabla_0\phi)\right]^{-2} \circ \phi^{-1}(y)dy\,.$$

Proof. Let $d\mu_x(y) \overset{\text{def.}}{=} \delta_{\phi(x)}(y)\,dy$ and perform a simple change of variables in the inner integral of (4.5). □

Therefore a conventional map for which $\det(\nabla_0\phi)$ is too small will have large bijectivity energy, and so $\mathcal{E}_b$ penalizes the failure of local injectivity. Moreover, the formula

$$\det(\nabla_0\phi(x)) = \lim_{r\to 0} \frac{\text{Area}(\phi(B_r(x))}{\text{Area}(B_r(x))}\,,$$

where $B_r(x)$ is the ball of radius r centered at x, tells us that energy-optimal maps will be such that $\det(\nabla_0\phi(x))$ is as spread out as possible, which promotes non-zero relative values of $\text{Area}(\phi(B_r(x))$ for each x, i.e. local surjectivity.



(a) Source points (b) (c) (d)

Figure 4.3: (a) An ant model with three source points and directions; (b,c,d) corresponding momentum fields for a soft map from the heat kernel signature. The blue shading of the targets shows the magnitude of the soft map in each case.

4.4.3 Descriptor Matching

Given $x \in M_0$ and $y \in M$, we can often write a function $c(x, y)$ measuring the geometric compatibility of x and y. For example, in our tests we take $c(x, y)$ to be the L^1 difference between wave kernel signatures [ASC11]. Then, we can measure the agreement between a soft map μ and c as

$$\mathcal{E}_c(\mu) \overset{\text{def.}}{=} \int_{M_0} \int_M c(x, y)\, d\mu_x(y). \tag{4.6}$$

4.5 Applications for Soft Map Analysis and Synthesis

We now show how the transportation potential Q and the Dirichlet energy from §4.4.1 can be used for analyzing soft maps. The goal of this section and subsequent ones in this chapter is illustration and application of the constructions from §4.4. Many discretizations of these theoretical constructions are possible and must be tuned to the particular application and data type under consideration.

See [SGB13] for details of the discretization of soft maps on triangle mesh surfaces used in this section. In short, this paper constructs soft maps in the language of the finite element method (FEM). Because soft maps are functions on a *four*-dimensional manifold $M_0 \times M$, dimensionality is reduced by constructing localized partitions of unity on the source and target surfaces as linear combinations of piecewise-linear hat functions. Gradients, Laplacians, and the like are computed as the restriction of piecewise-linear FEM to this reduced basis.

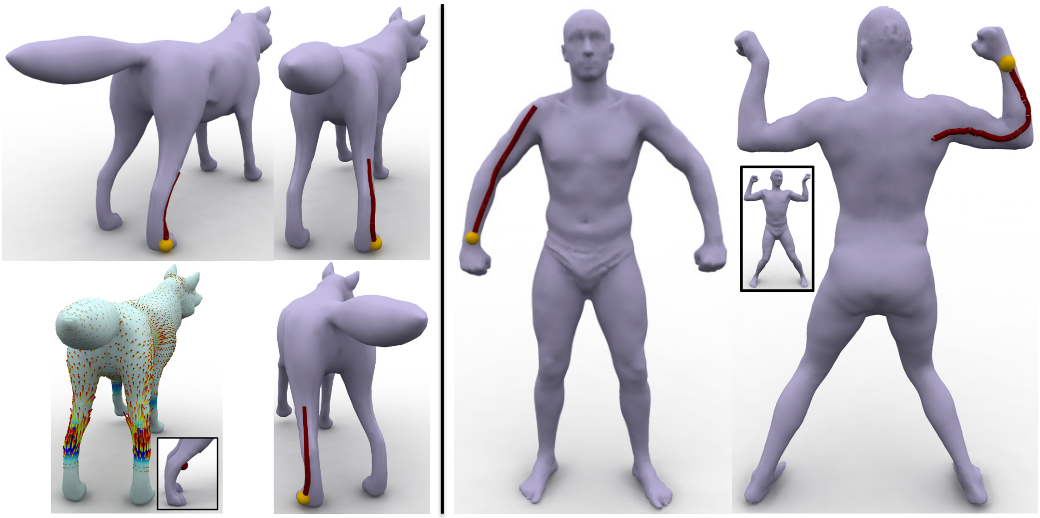


Figure 4.4: (left) The soft map (lower left) in blue does not distinguish between the two legs of the wolf, nor between radial points at the same height on a given leg. The velocity vector field ∇Q, however, deduces second-order information from the soft map. Given a path (upper left) on the source, corresponding paths (middle left) can be integrated along ∇Q from a single match; the soft map encodes forward and orientation-reversing maps, which can be traced depending on the initial match shown in yellow. (right) Following ∇Q transfers the path on the left model to the path on the right model. Path integration disambiguates the soft map, which does not differentiate points on rings around the arms.

4.5.1 Extracting Correspondences

The soft map density ρ is a function on the four-dimensional product space $M_0 \times M$ and is therefore hard to visualize. Picking a few salient source points $x \in M_0$ and showing the corresponding distributions on M gives some sense of the correspondence behavior suggested by ρ. But it remains difficult to see *dynamic* behavior in these correspondences as a source point is moved in a given direction.

The momentum field $\rho\nabla Q$ of ρ, however, captures exactly this type of behavior. In Figures 4.3, and 4.5, we illustrate how the momentum might be used to illustrate the dynamics of a soft map.

In particular, the fluid flow interpretation of optimal transportation views the probability density ρ as a collection of particles whose aggregate motion is by advection under the momentum field. We leverage this idea to extract pointwise correspondences from a soft map as follows. Suppose we are given a soft map density ρ from M_0 to M and a

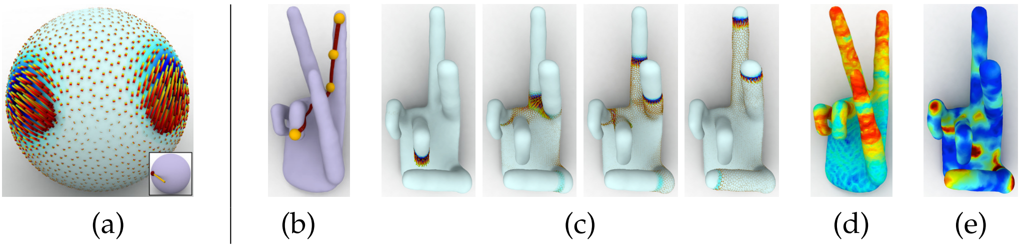


Figure 4.5: (a) Soft maps are capable of representing superpositions of uncertain point-to-point maps. We simulate such a soft map (the average two soft maps representing the identity and an orientation-reversing map of the sphere); the momentum field $\rho\nabla Q$ in red tracks the motion of both peaks of the map simultaneously (source point and direction boxed). (b) Four points marked along a geodesic on a source surface; (c) the corresponding images of a soft map from (b) constructed using the wave kernel signature [ASC11] on the target shaded from light to dark and with the momentum vector field in red; (d) the Dirichlet energy density of the map in log scale; (e) the reciprocal of the bijectivity energy density. Note in (c) that $\rho\nabla Q$ shows where mass of the soft map moves: up and down the target fingers and along the hand; the WKS cannot distinguish between the index and middle fingers. The Dirichlet energy density (c) is highest exactly where the map induces distortion at the joints.

single point-to-point correspondence $x \in M_0 \mapsto y \in M$. We can assume that y is near a peak of μ_x. Now given a path $\gamma_0(t) : [0, T) \to M_0$ starting at x, we can trace a path $\gamma : [0, T) \to M$ of corresponding points on M using the soft map velocity as a guide. In particular, we obtain an ordinary differential equation (ODE) for γ by substituting γ_0 for x and its derivative $\dot{\gamma}_0$ for V into the expression for the velocity:

$$\dot{\gamma} = \nabla Q(\gamma_0, \gamma) \cdot \dot{\gamma}_0 \quad \text{with} \quad \gamma(0) = y. \tag{4.7}$$

This ODE can be integrated in t using Euler's method. Figure 4.4 show two examples of this process. The left subfigure also shows that different choices of $y \in M$ yield different paths; this provides a strategy for isolating symmetries in ρ.

4.5.2 Analyzing Maps

The transportation potential, Dirichlet energy density, and bijectivity energy density all can be used to visualize and analyze characteristics of soft maps. We demonstrate our visualization techniques on maps from two sources.

First, when we are given a point-to-point map $\phi : M_0 \to M$. Proposition 8 shows that the traditional Dirichlet energy of ϕ is approximated by the Dirichlet energy of any

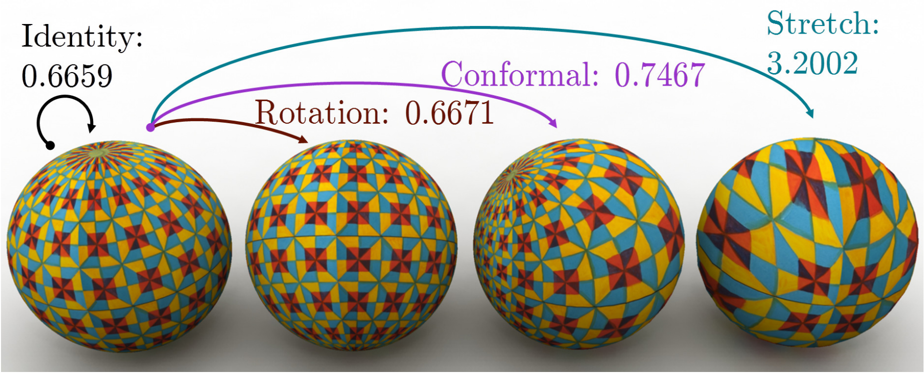
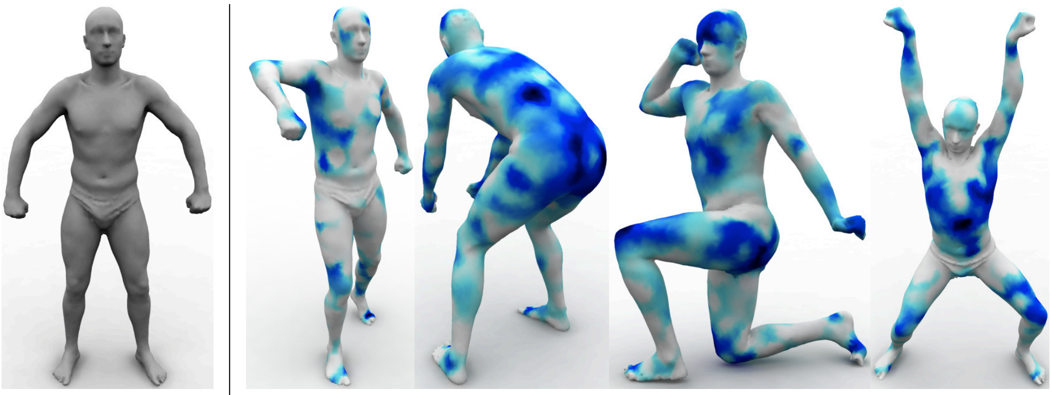


Figure 4.6: Four sphere maps illustrated by transferring textures. The Dirichlet energies of the approximating soft maps is indicated. The identity, rotation, and conformal maps yield nearly the same energy while the stretched map is larger. We expect low values for the first three maps because they are critical points of the point-to-point Dirichlet energy; the only map with exactly zero Dirichlet energy in the point-to-point case sends the entire source to a single point on the target.

sufficiently "close" soft map. This suggests that a strategy for analyzing the traditional Dirichlet energy of ϕ is to construct a soft map approximating ϕ and computing its Dirichlet energy. To do so, we represent ϕ at the finest level using a density ρ^{fine} that is sparse in the basis of hat functions on M_0 and per-triangle densities on M. For efficiency, we then project into a coarser basis in which we can evaluate the soft Dirichlet energy feasibly. Figure 4.6 shows how the behavior of the traditional Dirichlet energy for maps aligns with that of the Dirichlet energy of the projected soft maps.

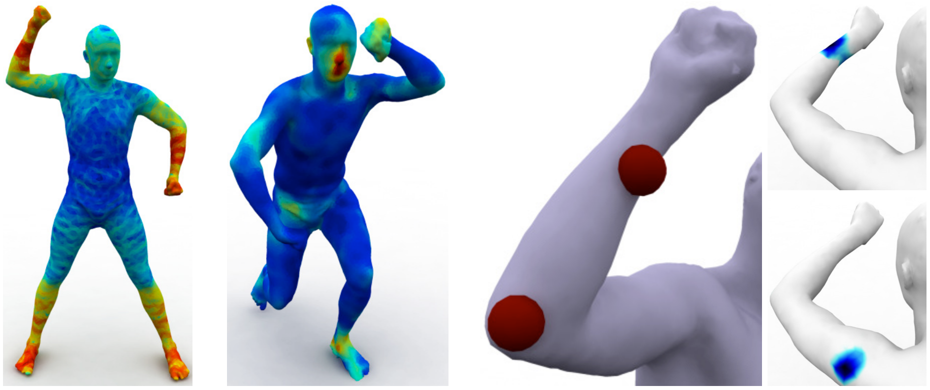
Second, soft maps can be obtained by constructing the maximum-entropy probability distribution derived from descriptor differences. That is, suppose $f_0 : M_0 \to \mathbb{R}^d$ and $f : M \to \mathbb{R}^d$ are descriptors for M_0 and M. Then the distribution we have in mind is $\rho(x,y) = Z(x)^{-1} \exp(-\|f_0(x) - f(y)\|^2/\sigma^2)$, where $Z(x)$ is a normalization factor. We use this as our soft map and project it onto the bases as above. In our examples, we use the heat kernel and wave kernel signatures [SOG09, ASC11].

In Figures 4.6 and 4.7, we use Dirichlet and bijectivity energy densities to visualize and quantify point-to-point map continuity; the remaining figures are constructed using descriptors. Figure 4.5 shows the Dirichlet density as a function on a mesh. Figure 4.8 shows how it can be used to find poorly-generated soft maps and points where distortion is undesirably high.



$\mathcal{E}_D$	0.75	0.77	0.80	0.81
$\mathcal{E}_b$	1.01	1.03	1.03	1.03

Figure 4.7: Four deformations of the mesh on the left with ground-truth correspondences colored by the reciprocal of the bijectivity energy density. Below the meshes are the Dirichlet energy $\mathcal{E}_D$ and bijectivity energy $\mathcal{E}_b$.



(a) Source (Dirichlet) $\mathcal{E}_D = 1102.9$

(b) Target (Bijectivity) $\mathcal{E}_b = 1.19$

(c) Stretching

Figure 4.8: Constructing a soft map from (a) to (b) using the wave kernel signature unexpectedly yields a large Dirichlet energy $\mathcal{E}_D$. Examining the energy density (a; in log scale) shows large distortion on the model's forearms; examining the map more closely in (c) reveals that the map inadvertently stretches the source's forearm to the target's entire arm.

4.6 Analysis and Decomposition of Soft Maps

Unlike point-to-point maps, the probability matrices for soft maps are amenable to linear-algebraic analysis to understand the information they encode. We illustrate this type of analysis for individual maps and collections of maps between surfaces.

Contrasting with the previous section, this section uses the discretization of soft maps proposed in [SNB+12]. This paper provides optimization techniques for computing coarse approximations of soft maps by minimizing weighted sums of analogs of the variational objectives from §4.4, yielding matrices A of probability values. For this discussion, we do not need differential interpretations of soft maps, and hence we can use a coarser approximation in which surfaces are partitioned into Voronoi cells; then, a soft map is represented as a piecewise constant set of probability values on products of these cells. The resulting optimizations are carried using standard linear programming packages.

4.6.1 Bases for Mapping

Suppose we are given a soft correspondence matrix $A = (a_{ij})$. If our correspondence is written in the identity basis, each row or column of A encodes a map from a single patch on M_0 or M, resp. This representation is not necessarily the most compact. For instance, if a surface admits a symmetry that cannot be resolved by a given descriptor, then symmetric patches always should be coupled. Similarly, if there is not enough evidence to distinguish nearby patches, they also can be coupled.

Once we have computed A, however, we can seek bases on M_0 and M that better respect such couplings *a posteriori*. In particular, projecting the uniform vector of ones out of the columns of A and performing a singular value decomposition (SVD) mimics the steps of principal component analysis (PCA), yielding an orthogonal basis (including the uniform vector) for the column space of A. This basis provides a simple representation of the couplings exhibited in the column marginals of A, and the singular values provide an indication of the importance of each basis vector. Explicitly including the uniform vector allows us to guarantee that our basis can represent probability distributions. A similar process can be carried out on the rows of A for a basis on the target surface.

Figure 4.9 shows eight members of the basis M resulting from SVD analysis of a map. The basis reveals patterns on M that should be mapped together, respecting symmetries

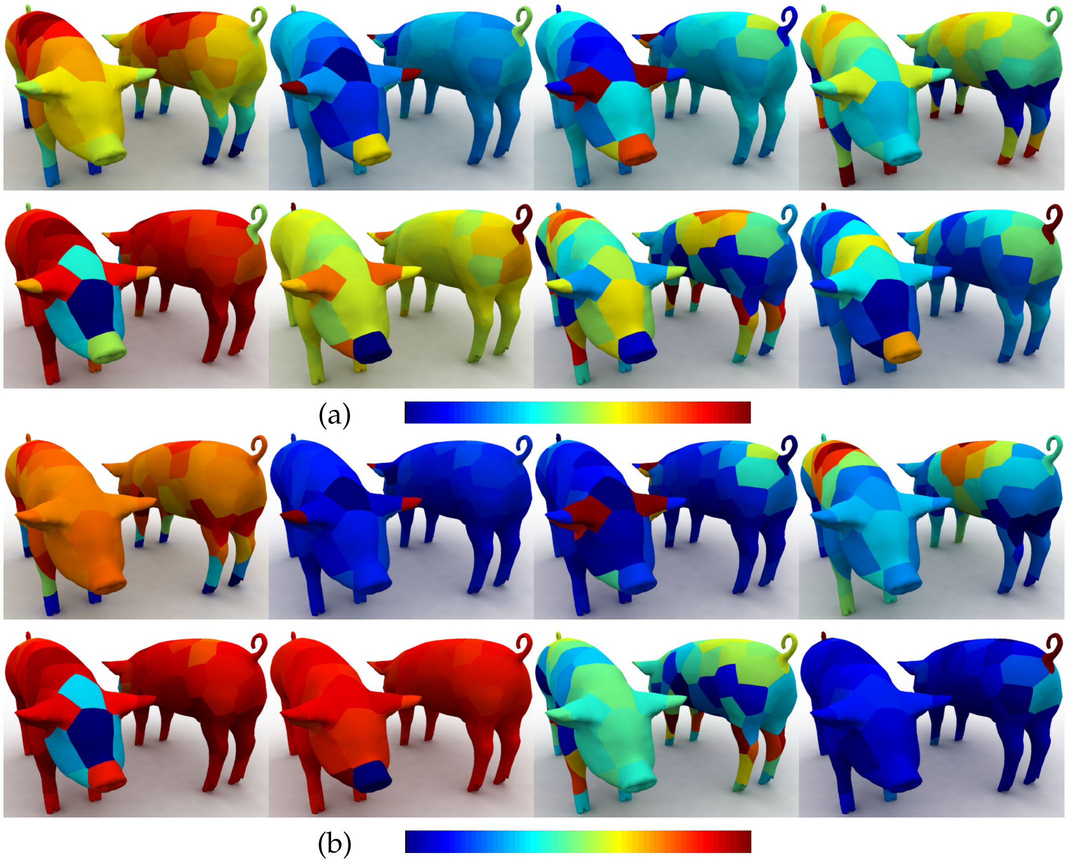


Figure 4.9: (a) The first eight SVD basis vectors from the first map in Figure 4.1, sorted by decreasing singular value, and (b) the same vectors "untangled" using [SBCBG11] to better show their support. Bases are colored using the scale below the images. These respect symmetries and are spread depending on the usefulness of ϕ for mapping each patch.

that are not disambiguated by ϕ; the basis on M_0 is similar. Such bases indicate the mapping resolution and couplings that should be expected for continuous maps respecting a given ϕ. Figure 4.10 shows a plot of the singular values from our decomposition. These singular values have a relatively long tail, so low-rank approximations of A can be obtained by projecting onto a restricted basis; Figure 4.11 shows such a projection onto the first nine basis vectors.

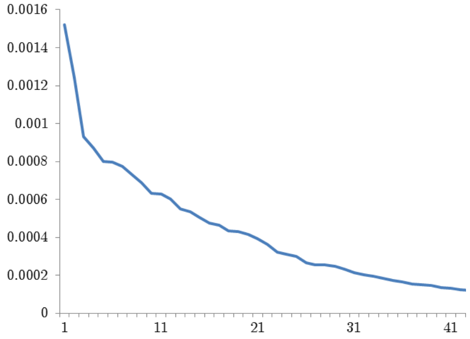


Figure 4.10: Singular values for the basis in Figure 4.9.

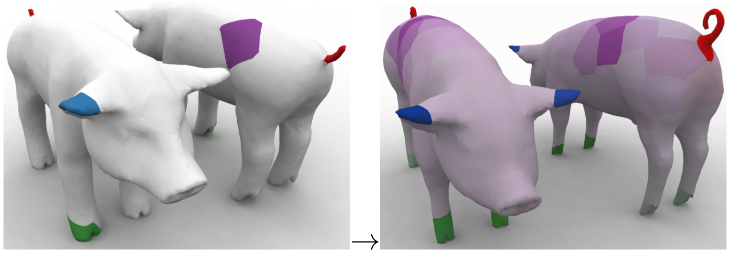


Figure 4.11: A low-rank map using the basis from Figure 4.9; it is nearly indistinguishable from the original map.

4.6.2 Understanding Collections of Shapes

Using SVD bases as in Section 4.6.1 makes it possible to express a single map using a few basis vectors and a smaller correspondence matrix. In some sense, the compactness here is not surprising, since the bases are tailor-made for the map in question. The two bases, however, live on M_0 and M *individually*, so in some sense the map is only expressed in the reduced correspondence matrix.

Consider now a *collection* of shapes $M_1, \ldots, M_n$, each with its own patch decomposition. We seek a "probabilistic basis" on each shape that captures its maps to all the others. Computing such a basis is a simple extension of our previous method: we simply concatenate the outgoing maps to *all* other shapes, project out the uniform distribution, and perform SVD. Figure 4.12 shows the results of an experiment in which a database of twenty shapes is mapped pairwise and analyzed using this technique. The resulting bases, illustrated for one shape in Figure 4.12(b), are more robust than those from Figure 4.9 since they are not subject to the particularities of a single map; they can be "untangled" using the method in [SBCBG11] (shown in Figure 4.12(c)) to illustrate their support more clearly,

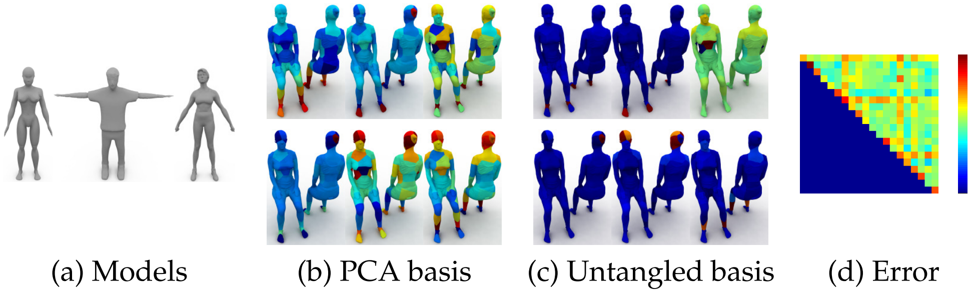


Figure 4.12: Pairwise maps between the models in (a) yield bases for mapping on the shapes including those in (b), which can be untangled to yield bases (c). A subset of the models used in the paper is shown. The untangled basis reveals the power of the reduced-rank representation, directly coupling symmetric patches that cannot be disambiguated using ϕ or continuity. For instance, feet and legs are coupled in the untangled basis and thus always are mapped together. L^2 soft correspondence matrix approximation error using the truncated twenty-vector basis is shown in (d); the color bar scales between 0% and 25% error.

although this process does not affect their span. Figure 4.12(d) shows that approximation using these non-map-specific bases remains relatively effective, demonstrating their generality.

Some of the highest "errors" in Figure 4.12(d) are along the diagonal, which represents self maps. The fact that descriptors match exactly can make the identity map dominate soft mapping output regardless of symmetries, see Figure 4.13(a); thus we leave them out of the SVD computation. Projecting onto the shape's mapping basis can alleviates this issue and makes for more symmetric self mappings as in Figure 4.13(b).

4.7 Discussion

There are many ways to view soft correspondences within the larger context of mapping algorithms. Primarily, they serve as a new map representation acknowledging uncertainty in the mapping problem, improving upon dissimilarity matrices using continuity to cull false matches. They can also be viewed as superpositions of symmetric or slippage-prone point-to-point maps whose spread reflects potential mapping quality latent in a given descriptor. Regardless of interpretation, soft correspondences deal with local and global ambiguities gracefully, admit straightforward analysis, and can be computed using convex optimization.

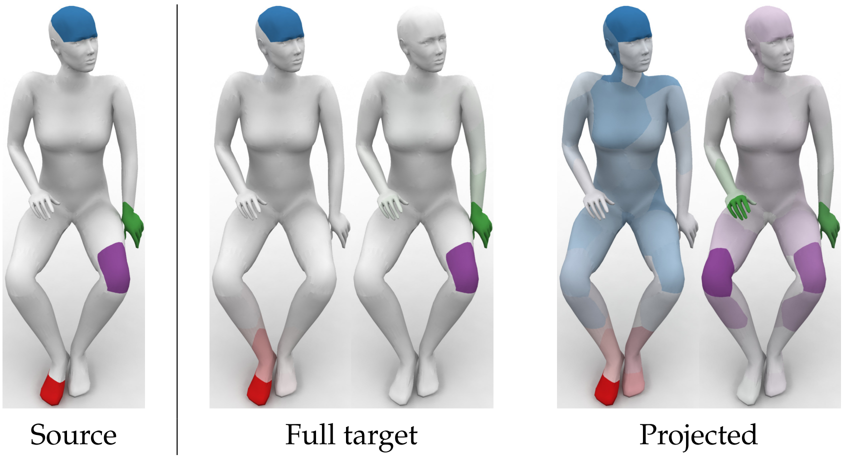


Figure 4.13: Map from a shape to itself before and after projecting onto the reduced basis. The maps on the right are more symmetric at the cost of being more spread out.

Together, these methods provide a toolkit for exploring and generating soft maps and indicate promising avenues for future research in surface mapping and analysis. As suggested earlier, many geometry processing algorithms implicitly make use of soft maps through descriptor differencing or by accumulating potential matches, and our proposed techniques can be used to understand the quality and structure of these constructions, including their discontinuities and locations where additional mapping evidence might increase bijectivity or sharpness. They also provide methods for displaying local variations of soft maps using momenta rather than small differences between probability distributions. With more specialized optimizations, it also may be possible to compute dense, continuous soft maps in analogy to the coarse maps illustrated here. In the end, this work represents a considerable step toward the design of a pipeline for generating and understanding soft maps backed by a convergent theory characterizing discrete and continuous behavior.

Chapter 5

Wasserstein Propagation Along Graphs

We conclude our consideration of problems in geometric data processing with an example in which the geometric aspect is more abstract, while the computational machinery remains similar to that introduced in Chapter 4. Similar to the mapping problem, we will couple the geometry of *two* domains. The source domain in this case, however, will be fundamentally discrete—a graph—while the target can be any metric space over which optimal transportation problems can be solved. Switching to a machine learning context, the shortest-path structure of the graph will provide notions proximity or similarity between nodes to be labeled with probability distributions over a target domain, leading to algorithms for semi-supervised and manifold-valued learning.

5.1 Introduction

Graph-based semi-supervised learning is an effective approach for learning problems involving a limited amount of labeled data [SNZ08]. Methods in this class typically propagate labels from a subset of nodes of a graph to the rest of the nodes. Usually each node is associated with a real number, but in many applications labels are more naturally expressed as histograms or probability distributions. For instance, the traffic density at a given location can be seen as a histogram over the 24-hour cycle; these densities may be known only where a service has cameras installed but need to be propagated to the entire map. Product ratings, climatic measurements, and other data sources exhibit similar structure.

While methods for numerical labels, such as [BN01, ZGL03, BNS06, ZB11, JYL$^+$12] (also see the survey by [Zhu08] and references therein), can be applied bin-by-bin to propagate normalized frequency counts, this strategy does not model interactions between histogram bins. As a result, a fundamental aspect of this type of data is ignored, leading to artifacts even when propagating Gaussian distributions.

Among first works directly addressing semi-supervised learning of probability distributions is [SB11], which propagates distributions representing class memberships. Their loss function, however, is based on Kullback-Leibler divergence, which cannot capture interactions between histogram bins. [TC09] allow interactions between bins by essentially modifying the underlying graph to its tensor product with a prescribed bin interaction graph; this approach loses probabilistic structure and tends to oversmooth. Similar issues have been encountered in the mathematical literature [McC97, AC11] and in vision/graphics applications [BvdPPH11, RPDB12] involving interpolating probability distributions. Their solutions attempt to find weighted barycenters of distributions, which is insufficient for propagating distributions along graphs.

The goal of this chapter is to provide an efficient and theoretically sound approach to graph-based semi-supervised learning of probability distributions. Similar to algorithms we have developed for mapping between surfaces, our strategy uses the machinery of optimal transportation. In particular, leveraging intuition from the previous chapter, we employ the two-Wasserstein distance between distributions to construct a regularizer measuring the “smoothness” of an assignment of a probability distribution to each graph node. The final assignment is produced by optimizing this energy while fitting the histogram predictions at labeled nodes.

Our technique has many notable properties. As certainty in the known distributions increases, it reduces to the method of label propagation via harmonic functions [ZGL03]. Also, the moments and other characteristics of the propagated distributions are characterized by those of the labeled nodes at minima of our smoothness energy. Our approach does not restrict the class of the distributions provided at labeled nodes, allowing for bimodality and other non-Gaussian properties. Finally, we prove that under an appropriate change of variables our objective can be minimized using a fast linear solve.

5.2 Preliminaries and Motivation

5.2.1 Label Propagation on Graphs

We consider generalization of the problem of label propagation on a graph $G = (V, E)$. Suppose a label function f is known on a subset of vertices $V_0 \subseteq V$, and we wish to extend f to the remainder $V \backslash V_0$. The classical approach of [ZGL03] minimizes the Dirichlet energy $\mathcal{E}_D[f] \overset{\text{def.}}{=} \sum_{(v,w)\in E} \omega_e (f_v - f_w)^2$ over the space of functions taking the prescribed values on V_0. Here ω_e is the weight associated to the edge $e = (v, w)$. $\mathcal{E}_D$ is a measure of smoothness; therefore the minimizer matches the prescribed labels with minimal variation in between. Minimizing this quadratic objective is equivalent to solving $\Delta f = 0$ on $V \backslash V_0$ for an appropriate positive definite Laplacian matrix Δ [CY00]. Solutions of this system are well-known to enjoy many regularity properties, making it a sound choice for smooth label propagation.

5.2.2 Propagating Probability Distributions

Suppose, however, that each vertex in V_0 is decorated with a probability distribution rather than a real number. That is, for each $v \in V_0$, we are given a probability distribution $\rho_v \in \text{Prob}(\mathbb{R})$. Our goal now is to propagate these distributions to the remaining vertices, generating a *distribution-valued map* $\rho : v \in V \mapsto \rho_v \in \text{Prob}(\mathbb{R})$ associating a probability distribution with every vertex $v \in V$. It must satisfy $\rho_v(x) \geq 0$ for all $x \in \mathbb{R}$ and $\int_{\mathbb{R}} \rho_v(x)\, dx = 1$. In §5.4 we consider the generalized case $\rho : V \to \text{Prob}(\Gamma)$ for alternative domains Γ including subsets of $\mathbb{R}^n$; most of the statements we prove about maps into $\text{Prob}(\mathbb{R})$ extend naturally to this setting with suitable technical adjustments.

In the applications we consider, such a propagation process should satisfy a number of properties:

- The spread of the propagated distributions should be related to the spread of the prescribed distributions.
- As the prescribed distributions in V_0 become peaked (concentrated around the mean), the propagated distributions should become peaked around the values obtained by propagating means of prescribed distributions via label propagation (e.g. [ZGL03]).
- The computational complexity of distribution propagation should be similar to that

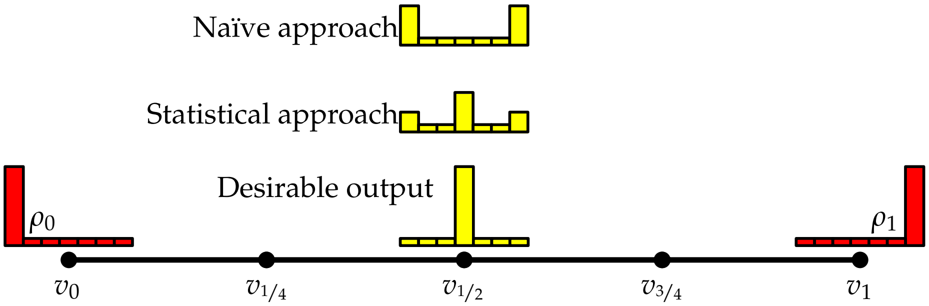


Figure 5.1: Propagating prescribed probability distributions (in red) to interior nodes of path graph identified with the interval $[0,1]$.

of scalar propagation.

The simplest method for propagating probability distributions is to extend [ZGL03] naïvely. For each $x \in \mathbb{R}$, we can view $\rho_v(x)$ as a label at $v \in V$ and solve the Dirichlet problem $\Delta\rho_v(x) = 0$ with $\rho_{v_0}(x)$ prescribed for all $v \in V_0$. The resulting functions $\rho_v(x)$ are distributions because the maximum principle guarantees $\rho_v(x) \geq 0$ for all x and $\int_{\mathbb{R}} \rho_v(x)\,dx = 1$ for all $v \in V$ since these properties hold at the boundary [CCK07].

It is easy to see, however, that this method has shortcomings. For instance, consider the case where G is a path graph representing the segment $[0,1]$ and the labeled vertices are the endpoints, $V_0 = \{0,1\}$. In this case, the naïve approach results in the linear interpolation $\rho_t(x) \overset{\text{def.}}{=} (1-t)\rho_0(x) + t\rho_1(x)$ at all intermediate graph vertices for $t \in (0,1)$. The propagated distributions are thus *bimodal* as in Figure 5.1a. Given our criteria, however, we would prefer an interpolation result closer to Figure 5.1c, which causes the peak in the boundary data simply to slide from left to right without introducing variance as t changes.

An alternative strategy for propagating probability distributions over V given boundary data on V_0 is to use a statistical approach. We could repeatedly draw an independent sample from each distribution in $\{\rho_v : v \in V_0\}$ and propagate the resulting scalars using a classical approach; binning the results of these repeated experiments provides a histogram-style distribution at each vertex in V. This strategy has a similar shortcomings to the naïve approach above. For instance, in the path graph example, the interpolated distribution is *trimodal* as in Figure 5.1b, with nonzero probability at both endpoints and for some v in the interior of V.

Of course, the desiderata above are application-specific. One key assumption is that the spread of the distributions is preserved, which differs from existing approaches which tend to blur the distributions. While this property is not intrinsically superior, in a way the experiments in §5.6 validate not only the algorithmic effectiveness of our technique but also this assumption about probabilistic data on graphs.

5.3 Wasserstein Propagation

Ad hoc methods for propagating distributions based on methods for scalar functions tend to have a number of drawbacks. Therefore, we tackle this problem using a technique designed explicitly for the probabilistic setting. To this end, we formulate the semi-supervised problem at hand as the optimization of a Dirichlet energy for distribution-valued maps generalizing the classical Dirichlet energy.

Similar to the construction in [SB11], we replace the square distance between scalar function values appearing in the classical Dirichlet energy (namely the quantity $|f_v - f_w|^2$) with an appropriate distance between the distributions ρ_v and ρ_w. Rather than using the bin-by-bin KL divergence, however, we use the Wasserstein distance with quadratic cost between probability distributions with finite second moment on $\mathbb{R}$, defined in this case as

$$\mathcal{W}_2(\rho_v, \rho_w) \overset{\text{def.}}{=} \inf_{\pi \in \Pi(\rho_v, \rho_w)} \left(\iint_{\mathbb{R}^2} |x-y|^2 \, d\pi(x,y) \right)^{1/2},$$

where $\Pi(\rho_0, \rho_1) \subseteq \mathrm{Prob}(\mathbb{R}^2)$ is the set of probability distributions π on $\mathbb{R}^2$ satisfying the marginal constraints

$$\int_0^1 \pi(x,y)\, dx = \rho_w(y) \quad \text{and} \quad \int_0^1 \pi(x,y)\, dy = \rho_v(x)\,.$$

In machine learning, the Wasserstein distance already has shown promise for search and clustering techniques [IVdAdC11, ADKU11].

With these ideas in place, we define a Dirichlet energy for a distribution-valued map from a graph into $\mathrm{Prob}(\mathbb{R})$ by

$$\mathcal{E}_D[\rho] \overset{\text{def.}}{=} \sum_{(v,w) \in E} \mathcal{W}_2^2(\rho_v, \rho_w)\,, \tag{5.1}$$

along with the notion of *Wasserstein propagation* of distribution-valued maps given prescribed boundary data:

WASSERSTEIN PROPAGATION

Minimize $\mathcal{E}_D[\rho]$ in the space of distribution-valued maps with prescribed distributions at all $v \in V_0$.

5.3.1 Theoretical Properties

Solutions of the Wasserstein propagation problem satisfy many desirable properties that we will establish below. Before proceeding, however, we recall a fact about the Wasserstein distance. Let $\rho \in \text{Prob}(\mathbb{R})$ be a probability distribution. Then its cumulative distribution function (CDF) is given by $F(x) \overset{\text{def.}}{=} \int_{-\infty}^{x} \rho(y)\, dy$, and the *generalized inverse* of the its CDF is given by $F^{-1}(s) \overset{\text{def.}}{=} \inf\{x \in \mathbb{R} : F(x) > s\}$. Then the following result, suggested informally in §1.4.3, holds.

Proposition 12. *[[Vil03], Theorem 2.18] Let $\rho_0, \rho_1 \in \text{Prob}(\mathbb{R})$ with CDFs F_0, F_1. Then*

$$\mathcal{W}_2^2(\rho_0, \rho_1) = \int_0^1 (F_1^{-1}(s) - F_0^{-1}(s))^2\, ds\,. \tag{5.2}$$

By applying (5.2) to the minimization problem (5.1), we obtain a *linear* strategy for our propagation problem.

Proposition 13. *Wasserstein propagation can be characterized in the following way. For each $v \in V_0$ let F_v be the CDF of the distribution ρ_v. Now suppose that for each $s \in [0, 1]$ we determine $g_s : V \to \mathbb{R}$ as the solution of the classical Dirichlet problem*

$$\begin{aligned} \Delta g_s &= 0 \quad \forall\, v \in V \setminus V_0 \\ g_s(v) &= F_v^{-1}(s) \quad \forall\, v \in V_0\,. \end{aligned} \tag{5.3}$$

Then for each v, the function $s \mapsto g_s(v)$ is the inverse CDF of a probability distribution ρ_v. Moreover, the distribution-valued map $v \mapsto \rho_v$ minimizes the Dirichet energy (5.1).

Proof. Let $\mathcal{X}$ be the set of functions $g : V \times [0, 1] \to \mathbb{R}$ satisfying the constraints $g_s(v) = F_v^{-1}(s)$ for all $s \in [0, 1]$ and all $v \in V_0$. Consider the minimization problem

$$\min_{g \in \mathcal{X}} \hat{\mathcal{E}}_D(g) \overset{\text{def.}}{=} \sum_{(u,v) \in E} \int_0^1 (g_s(u) - g_s(v))^2\, ds\,.$$

The solution of this optimization for each s is exactly a solution of the classical Dirichlet problem (5.3) on G. Moreover, the maximum principle implies that $g_s(v) \leq g_{s'}(v)$ whenever $s < s'$, which holds by definition for all $v \in V_0$, can be extended to all $v \in V$ [CCK07]. Hence $g_s(v)$ can be interpreted as an inverse CDF for each $v \in V$ form which we can define a distribution-valued map $\rho : v \mapsto \rho_v$. Since $\hat{\mathcal{E}}_D$ takes on its minimum value in the subset of $\mathcal{X}$ consisting of inverse CDFs, and $\hat{\mathcal{E}}_D$ coincides with $\mathcal{E}_D$ on this set, ρ is a solution of the Wasserstein propagation problem. □

Distribution-valued maps $\rho : V \to \mathrm{Prob}(\mathbb{R})$ propagated by optimizing (5.1) satisfy many analogs of functions extended using the classical Dirichlet problem. Two results of this kind concern the *mean* $m(v)$ and the *variance* $\sigma(v)$ of the distributions ρ_v as functions of V. These are defined as

$$m(v) \overset{\text{def.}}{=} \int_{-\infty}^{\infty} x\rho_v(x)\, dx$$
$$\sigma^2(v) \overset{\text{def.}}{=} \int_{-\infty}^{\infty} (x - m(v))^2 \rho_v(x)\, dx\,.$$

Proposition 14. *Suppose the distribution-valued map $\rho : V \to \mathrm{Prob}(\mathbb{R})$ is obtained using Wasserstein propagation. Then for all $v \in V$ the following estimates hold.*

- $\inf_{v_0 \in V_0} m(v_0) \leq m(v) \leq \sup_{v_0 \in V_0} m(v_0)$.
- $0 \leq \sigma(v) \leq \sup_{v_0 \in V_0} \sigma(v_0)$.

Proof. Both estimates can be derived from the following formula. Let $\rho \in \mathrm{Prob}(\mathbb{R})$ and let $\phi : \mathbb{R} \to \mathbb{R}$ be any integrable function. If we apply the change of variables $s = F(x)$ where F is the CDF of ρ in the integral defining the expectation value of ϕ with respect to ρ, we get

$$\int_{-\infty}^{\infty} \phi(x)\rho(x)\, dx = \int_0^1 \phi(F^{-1}(s))\, ds\,.$$

Thus $m(v) = \int_0^1 F_v^{-1}(s)\, ds$ and $\sigma^2(v) = \int_0^1 (F_v^{-1}(s) - m(v))^2\, ds$ where F_v is the CDF of ρ_v for each $v \in V$.

Assume ρ minimizes (5.1) with fixed boundary constraints on V_0. By Proposition 13, we then have $\Delta F_v^{-1} = 0$ for all $v \in V$. Therefore $\Delta m(v) = \int_0^1 \Delta F_v^{-1}(s)\, ds = 0$, so m is a harmonic function on V. The estimates for m follow by the maximum principle for harmonic functions. Also,

$$\begin{aligned}\Delta[\sigma^2(v)] &= \int_0^1 \Delta(F_v^{-1}(s) - m(v))^2\, ds \\ &= \sum_{(v,v')\in E} \int_0^1 \left(a(v,s) - a(v',s)\right)^2 ds \\ &\geq 0, \qquad \text{where } a(v,s) \overset{\text{def.}}{=} F_v^{-1}(s) - m(v),\end{aligned}$$

since $\Delta F_v^{-1}(s) = \Delta m(v) = 0$. Thus σ^2 is a subharmonic function and the upper bound for σ^2 follows by the maximum principle for subharmonic functions. □

Finally, we check that if we encode a classical interpolation problem using Dirac delta distributions, we recover the classical solution. The essence of this result is that if the boundary data for Wasserstein propagation has zero variance, then the solution must also have zero variance.

Proposition 15. *Suppose that there exists $u : V_0 \to \mathbb{R}$ such that $\rho_v(x) = \delta(x - u(v))$ for all $v \in V_0$. Then, the solutions of the classical Dirichlet problem and the Wasserstein propagation problem coincide in the following way. Suppose that $f : V \to \mathbb{R}$ satisfies the classical Dirichlet problem with boundary data u. Then $\rho_v(x) \overset{\text{def.}}{=} \delta(x - f(v))$ minimizes* (5.1) *subject to the fixed boundary constraints.*

Proof. The boundary data for ρ given here yields the boundary data $g_s(v) = u(v)$ for all $v \in V_0$ and $s \in [0,1)$ in the Dirichlet problem (5.3). The solution of this Dirichlet problem is thus also constant in s, let us say $g_s(v) = f(v)$ for all $s \in [0,1)$ and $v \in V$. The only distributions whose inverse CDFs are of this form are δ-distributions; hence $\rho_v(x) = \delta(x - f(v))$ as desired. □

5.3.2 Application to Smoothing

Using the connection to the classical Dirichlet problem in Proposition 13 we can extend our treatment to other differential equations. There is a large space of differential equations that have been adapted to graphs via the discrete Laplacian Δ; here we focus on the heat equation, considered e.g. in [CCK07].

The heat equation for scalar functions is applied to smoothing problems; for example, in $\mathbb{R}^n$ solving the heat equation is equivalent to Gaussian convolution. Just as the

Dirichlet equation on F^{-1} is equivalent to Wasserstein propagation, heat diffusion on F^{-1} is equivalent to gradient flows of the energy $\mathcal{E}_D$ in (5.1), providing a straightforward way to understand and implement such a diffusive process.

Proposition 16. *Let $\rho : V \to \mathrm{Prob}(\mathbb{R})$ be a distribution-valued map and let $F_v : [0,1] \to \mathbb{R}$ be the CDF of ρ_v for each $v \in V$. Then these two procedures are equivalent:*

- *Mass-preserving flow of ρ in the direction of steepest descent of the Dirichlet energy.*
- *Heat flow of the inverse CDFs.*

Proof. A mass-preserving flow of ρ is a family of distribution-valued maps $\rho_\varepsilon : V \to \mathrm{Prob}(\mathbb{R})$ with $\varepsilon \in (-\varepsilon_0, \varepsilon_0)$ that satisfies the equations

$$\left.\begin{aligned} \frac{\partial \rho_{v,\varepsilon}(t)}{\partial \varepsilon} + \frac{\partial}{\partial t}\big(Y_v(\varepsilon,t)\rho_{v,\varepsilon}(t)\big) = 0 \\ \rho_{v,0}(t) = \rho_v(t) \end{aligned}\right\} \ \forall\, v \in V,$$

where $Y_v : (-\varepsilon_0, \varepsilon_0) \times \mathbb{R} \to \mathbb{R}$ is an arbitrary function that governs the flow. By applying the change of variables $t = F_{v,\varepsilon}^{-1}(s)$ using the inverse CDFs of the $\rho_{v,\varepsilon}$, we find that this flow is equivalent to the equations

$$\left.\begin{aligned} \frac{\partial F_{v,\varepsilon}^{-1}(s)}{\partial \varepsilon} = Y_v(\varepsilon, F_{v,\varepsilon}^{-1}(s)) \\ F_{v,0}^{-1}(s) = F_v^{-1}(s) \end{aligned}\right\} \ \forall\, v \in V.$$

A short calculation starting from (5.1) now leads to the derivative of the Dirichlet energy under such a flow, namely

$$\frac{d\mathcal{E}_D(\rho_\varepsilon)}{d\varepsilon} = -2 \sum_{v \in V} \int_0^1 \Delta(F_{v,\varepsilon}^{-1}) \cdot Y_v(\varepsilon, F_{v,\varepsilon}^{-1}(s))\, ds\,.$$

Thus, steepest descent for the Dirichlet energy is achieved by choosing $Y_v(\varepsilon, F_{v,\varepsilon}^{-1}(s)) \overset{\text{def.}}{=} \Delta(F_{v,\varepsilon}(s))$ for each v, ε, s. As a result, the equation for the evolution of $F_{v,\varepsilon}^{-1}$ becomes

$$\left.\begin{aligned} \frac{\partial F_{v,\varepsilon}^{-1}(s)}{\partial \varepsilon} = \Delta(F_{v,\varepsilon}^{-1}(s)) \\ F_{v,0}^{-1}(s) = F_v^{-1}(s) \end{aligned}\right\} \ \forall\, v \in V,$$

which is exactly heat flow of $F_{v,\varepsilon}^{-1}$. □

5.4 Generalization

Our preceding discussion involves distribution-valued maps into Prob($\mathbb{R}$), but in a more general setting we might wish to replace Prob($\mathbb{R}$) with Prob(Γ) for an alternative domain Γ carrying a distance metric d. Our original formulation of Wasserstein propagation easily handles such an extension by replacing $|x-y|^2$ with $d(x,y)^2$ in the definition of $\mathcal{W}_2$. Furthermore, although proofs in this case are considerably more involved, some key properties proved above for Prob($\mathbb{R}$) extend naturally.

In this case, we no longer can rely on the computational benefits of Propositions 13 and 16 but can solve the propagation problem directly. If Γ is discrete, then Wasserstein distances between ρ_v's can be computed using a linear program. Suppose we represent two histograms as $\{a_1,\ldots,a_m\}$ and $\{b_1,\ldots,b_m\}$ with $a_i,b_i \geq 0\ \forall i$ and $\sum_i a_i = \sum_i b_i = 1$. Then, the definition of $\mathcal{W}_2$ yields the optimization:

$$\begin{aligned} \mathcal{W}_2^2(\{a_i\},\{b_j\}) = \min \sum_{ij} d_{ij}^2 x_{ij} \\ \text{s.t.} \sum_j x_{ij} = a_i\ \forall i \\ \sum_i x_{ij} = b_j\ \forall j \\ x_{ij} \geq 0\ \forall i,j. \end{aligned} \tag{5.4}$$

Here d_{ij} is the distance from bin i to bin j, which need not be proportional to $|i-j|$.

From this viewpoint, the energy $\mathcal{E}_D$ from (5.1) remains convex in ρ and can be optimized using a linear program simply by summing terms of the form (5.4) above:

$$\begin{aligned} &\min_{\rho,x} \sum_{e\in E}\sum_{ij} d_{ij}^2 x_{ij}^{(e)} \\ &\text{s.t.} \sum_j x_{ij}^{(e)} = \rho_{vi}\ \forall e=(v,w)\in E, i\in S \\ &\sum_i x_{ij}^{(e)} = \rho_{wj}\ \forall e=(v,w)\in E, j\in S \\ &\sum_i \rho_{vi} = 1\ \forall v\in V \qquad \rho_{vi} \text{ fixed } \forall v\in V_0 \\ &\rho_{vi} \geq 0\ \forall v\in V, i\in S \quad x_{ij} \geq 0\ \forall i,j\in S, \end{aligned}$$

where $S = \{1, \ldots, m\}$.

5.5 Algorithm Details

We handle the general case from §5.4 by optimizing the linear programming formulation directly. Given the size of these linear programs, we use large-scale barrier method solvers.

The characterizations in Propositions 13 and 16, however, suggest a straightforward discretization and accompanying set of optimization algorithms in the linear case. In fact, we can recover propagated distributions by inverting the graph Laplacian Δ via a sparse linear solve, leading to near-real-time results for moderately-sized graphs G.

For a given graph $G = (V, E)$ and subset $V_0 \subseteq V$, we discretize the domain $[0, 1]$ of F_v^{-1} for each v using a set of evenly-spaced samples $s_0 = 0, s_1, \ldots, s_m = 1$. This representation supports any ρ_v provided it is possible to sample the inverse CDF from Proposition 12 at each s_i. In particular, when the underlying distributions are histograms, we model ρ_v using δ functions at evenly-spaced bin centers, which have piecewise constant CDFs; we model continuous ρ_v using piecewise linear interpolation. Regardless, in the end we obtain a non-decreasing set of samples $(F^{-1})_v^1, \ldots, (F^{-1})_v^m$ with $(F^{-1})_v^1 = 0$ and $(F^{-1})_v^m = 1$.

Now that we have sampled F_v^{-1} for each $v \in V_0$, we can propagate to the remainder $V \backslash V_0$. For each $i \in \{1, \ldots, m\}$, we solve the system from (5.3):

$$\begin{aligned} \Delta g = 0 \quad \forall\, v \in V \setminus V_0 \\ g(v) = (F^{-1})_v^i \quad \forall\, v \in V_0 \,. \end{aligned} \tag{5.5}$$

In the diffusion case, we replace this system with implicit time stepping for the heat equation, iteratively applying $(I - t\Delta)^{-1}$ to g for diffusion time step t. In either case, the linear solve is sparse, symmetric, and positive definite; we apply Cholesky factorization to solve the systems directly.

This process propagates F^{-1} to the entire graph, yielding samples $(F^{-1})_v^i$ for all $v \in V$. We invert once again to yield samples ρ_v^i for all $v \in V$. Of course, each inversion incurs some potential for sampling and discretization error, but in practice we are able to oversample sufficiently to overcome most potential issues. When the inputs ρ_v are discrete histograms, we return to this discrete representation by integrating the resulting $\rho_v \in \text{Prob}([0, 1])$ over the width of the bin about the center defined above.

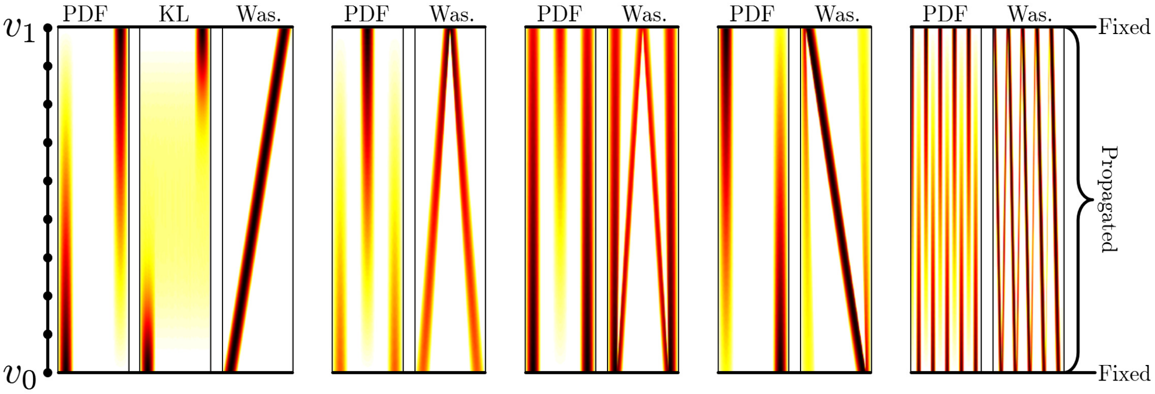


Figure 5.2: Comparison of propagation strategies on a linear graph (coarse version on left); each horizontal slice represents a vertex $v \in V$, and the colors from left to right in a slice show ρ_v. [SB11] (KL) is shown only in one example because it has qualitatively similar behavior to the PDF strategy.

This algorithm is efficient even on large graphs and is easily parallelizable. For instance, the initial sampling steps for obtaining F^{-1} from ρ are parallelizable over $v \in V_0$, and the linear solve (5.5) can be parallelized over samples i. Direct solvers can be replaced with iterative solvers for particularly large graphs G; regardless, the structure of such a solve is well-understood and studied, e.g. in [KFS13].

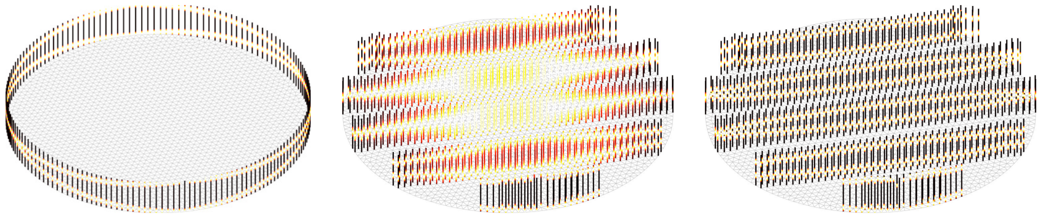
5.6 Experiments

We run our scheme through a number of tests demonstrating its strengths and weaknesses compared to other potential methods for propagation. We compare Wasserstein propagation with the strategy of propagating probability distribution functions (PDFs) directly, as described in §5.2.2.

5.6.1 Synthetic Tests

We begin by considering the behavior of our technique on synthetic data designed to illustrate its various properties.

One-Dimensional Examples. Figure 5.2 shows "displacement interpolation" properties inherited by our propagation technique from the theory of optimal transportation. The



Boundary conditions PDF propagation Wasserstein propagation

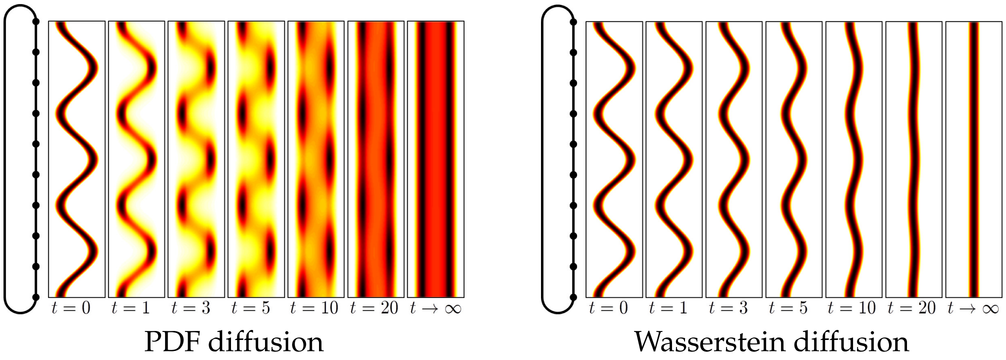
Figure 5.3: PDF and Wasserstein propagation on a meshed circle with prescribed boundary distributions. The underlying graph is shown in grey, and probability distributions at vertices $v \in V$ are shown as vertical bars colored by the density ρ_v; we invert the color scheme of Figures 5.2 and 5.4 to improve contrast. Propagated distributions are computed for *all* vertices but for clarity are shown at representative slices of the circle.

underlying graph is a line as in Figure 5.1, along the vertical axis. Horizontally, each image is colored by values in ρ_v.

The bottom and top vertices v_0 and v_1 have fixed distributions ρ_{v_0} and ρ_{v_1}, and the remaining vertices receive ρ_v via one of two propagation techniques. The left of each pair propagates distributions by solving a classical Dirichlet problem independently for each bin of the probability distribution function (PDF) ρ_v, whereas the right of each pair propagates inverse CDFs using our method in §5.5.

By examining the propagation behavior from the bottom to the top of this figure, it is easy to see how the naïve PDF method varies from Wasserstein propagation. For instance, in the leftmost example both ρ_{v_0} and ρ_{v_1} are unimodal, yet when propagating PDFs all the intermediate vertices have bimodal distributions; furthermore, no relationship is determined between the two peaks. Contrastingly, our technique identifies the modes of ρ_{v_0} and ρ_{v_1}, linearly moving the peak from one side to the other.

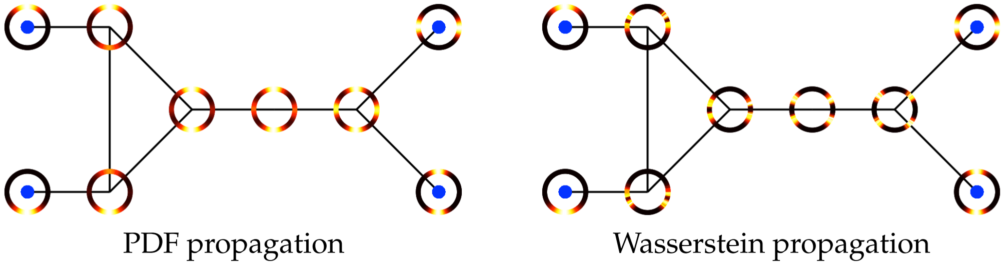
Boundary Value Problems. Figure 5.3 illustrates our algorithm on a less trivial graph G. To mimic a typical test case for classical Dirichlet problems, our graph is a mesh of the unit circle, and we propagate ρ_v from fixed distributions on the boundary. Unlike the classical case, however, our prescribed boundary distributions ρ_v are multimodal. Once again, Wasserstein propagation recovers a smoothly-varying set of distributions whose peaks behave like solutions to the classical Dirichlet problem. Propagating probability directions rather than inverse CDFs yields somewhat similar modes, but with much higher entropy and variance especially at the center of the circle.



PDF diffusion

Wasserstein diffusion

Figure 5.4: Comparison of PDF diffusion and Wasserstein diffusion; in both cases the leftmost distribution comprises the initial conditions, and several time steps of diffusion are shown left-to-right. The underlying graph G is the circle on the left.



PDF propagation

Wasserstein propagation

Figure 5.5: Interpolation of distributions on $\mathbb{S}^1$ via PDF propagation and Wasserstein propagation; in these figures, the vertices with valence 1 have prescribed distributions ρ_v and the remaining vertices have distributions from propagation.

Diffusion. Figure 5.4 illustrates the behavior of Wasserstein diffusion compared with simply diffusing distribution values directly. When PDF values are diffused directly, as time t increases the distributions simply become more and more smooth until they are uniform not only along G but also as distributions on $\text{Prob}([0,1])$. Contrastingly, Wasserstein diffusion preserves the uncertainty from the initial distributions but does not increase it as time progresses.

Alternative Target Domain. Figure 5.5 shows an example in which the target is $\text{Prob}(\mathbb{S}^1)$, where $\mathbb{S}^1$ is the unit circle, rather than $\text{Prob}([0,1])$. We optimize the $\mathcal{E}_D$ using the linear program in §5.4 rather than the linear algorithm for $\text{Prob}([0,1])$. Conclusions from this example are similar to those from Figure 5.3: Wasserstein propagation identifies peaks from different prescribed boundary distributions without introducing variance, while PDF

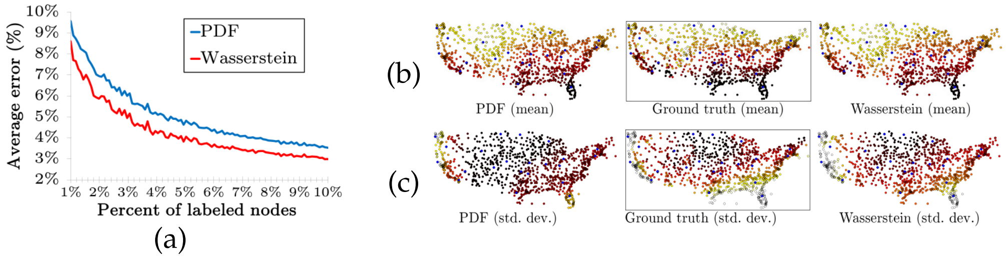


Figure 5.6: We propagate histograms of temperatures collected over time to a map of the United States: (a) Average error at propagated sites as a function of the number of nodes with labeled distributions; (b) means of the histograms at the propagated sites from a typical trial in (a); (c) standard deviations at the propagated sites. Vertices with prescribed distributions are shown in blue and comprise $\sim 2\%$ of V.

propagation exhibits much higher variance in the interpolated distributions and does not "move" peaks from one location to another.

5.6.2 Real-World Data

We now evaluate our techniques on real-world input. To evaluate the quality of our approach relative to ground truth, we will use the *one*-Wasserstein distance, or Earth Mover's Distance [RTG00], formulated by removing the square in the formula for $\mathcal{W}_2^2$. We use this distance, given on $\mathrm{Prob}(\mathbb{R})$ by the L^1 distance between (non-inverted) CDFs, because it does not favor the $\mathcal{W}_2$ distance used in Wasserstein propagation while taking into account the ground distances. We consider weather station coordinates as defining a point cloud on the plane and compute the point cloud Laplacian using the approach of [CL06].

Temperature Data. Figure 5.6 illustrates the results of a series of experiments on weather data on a map of the United States.[1] Here, we have $|V| = 1113$ sites each collecting daily temperature measurements, which we classify into 100 bins at each vertex. In each experiment, we choose a subset $V_0 \subseteq V$ of vertices, propagate the histograms from these vertices to the remainder of V, and measure the error between the propagated and ground-truth histograms.

Figure 5.6a shows quantitative results of this experiment. Here we show the average histogram error per vertex as a function of the percent of nodes in V with fixed labels; the fixed vertices are chosen randomly, and errors are averaged over 20 trials for each

[1] National Climatic Data Center

percentage. The Wasserstein strategy consistently outperforms naïve PDF interpolation with respect to our error metric and approaches relatively small error with as few as 5% of the labels fixed.

Figures 5.6b and 5.6c show results for a single trial. We color the vertices $v \in V$ by the mean (b) and standard deviation (c) of ρ_v from PDF and Wasserstein propagation. Both yield similar mean temperatures on $V \backslash V_0$, which agree with the means of the ground truth data. The standard deviations, however, better illustrate differences between the approaches. In particular, the standard deviations of the Wasserstein-propagated distributions approximately follow those of the ground truth histograms, whereas the PDF strategy yields high standard deviations nearly everywhere on the map due to undesirable smoothing effects.

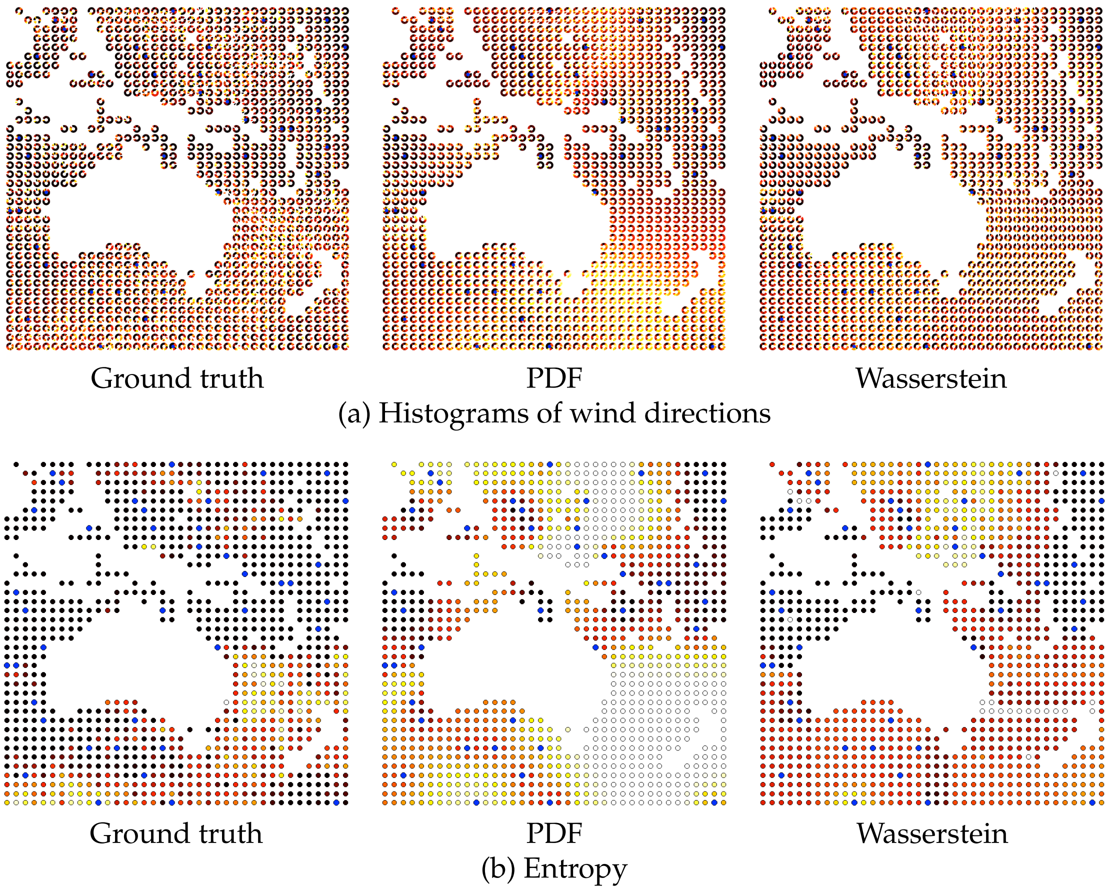
Wind Directions. We apply the general formulation in §5.4 to propagating distributions on the unit circle $\mathbb{S}^1$ by considering histograms of wind *directions* collected over time by nodes on the ocean outside of Australia.[2]

In this experiment, we keep approximately 4% of the data points and propagate to the remaining vertices. Both the PDF and Wasserstein propagation strategies score similarly with respect to our error metric; in the experiment shown, Wasserstein propagation exhibits 6.6% average error per node and PDF propagation exhibits 6.1% average error per node. Propagation results are illustrated in Figure 5.7a.

The nature of the error from the two strategies, however, is quite different. In particular, Figure 5.7b shows the same map colored by the entropy of the propagated distributions. PDF propagation exhibits high entropy away from the prescribed vertices, reflecting the fact that the propagated distributions at these points approach uniformity. Wasserstein propagation, on the other hand, has a more similar pattern of entropy to that of the ground truth data, reflecting structure like that demonstrated in Proposition 14.

Non-Euclidean Interpolation. Proposition 15 suggests an application outside histogram propagation. In particular, if the vertices of V_0 have prescribed distributions that are δ functions encoding individual points as mapping targets, all propagated distributions also will be δ functions. Thus, one strategy for interpolation is to encode the problem probabilistically using δ distributions, interpolate using Wasserstein propagation, and then extract

[2] WindSat Remote Sensing Systems



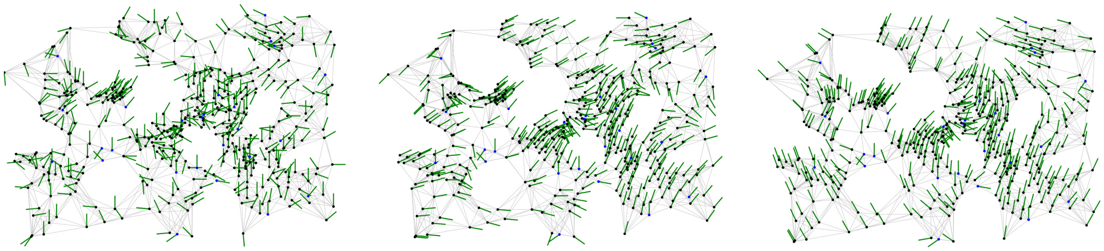
Ground truth PDF Wasserstein

(a) Histograms of wind directions

Ground truth PDF Wasserstein

(b) Entropy

Figure 5.7: (a) Interpolating histograms of wind directions using the PDF and Wasserstein propagation methods, illustrated using the same scheme as Figure 5.5; (b) entropy values from the same distributions.



Ground truth PDF (19%) Wasserstein (15%)

Figure 5.8: Learning wind directions on the unit circle $\mathbb{S}^1$.

peaks of the propagated distributions. Experimentally we find that optima of the linear program in §5.4 with peaked prescribed distributions yield peaked distributions ρ_v for all $v \in V$ even when the target is not $\mathrm{Prob}(\mathbb{R})$; we leave a proof for future work.

In Figure 5.8, we apply this strategy to interpolating angles on $\mathbb{S}^1$ from a single day of wind data on a map of Europe.[3] Classical Dirichlet interpolation fails to capture the identification of angles 0 and 2π. Contrastingly, if we encode the boundary conditions as peaked distributions on $\mathrm{Prob}(\mathbb{S}^1)$, we can interpolate using Wasserstein propagation without losing structure. The resulting distributions are peaked about a single maximum, so we extract a direction field as the mode of each ρ_v. Despite noise in the dataset we achieve 15% error rather than the 19% error obtained by classical Dirichlet interpolation of angles disregarding periodicity.

5.7 Discussion

It is easy to formulate strategies for histogram propagation by applying methods for propagating scalar functions bin-by-bin. Here, however, we have shown that propagating instead inverse CDFs has a deep connections to the theory of optimal transportation and provides superior results, making it a strong yet still efficient choice. This basic connection gives our method theoretical and practical soundness that is difficult to guarantee otherwise.

While our algorithms show promise as practical techniques, we leave many avenues for future study. Most prominently, the generalization in §5.4 can be applied to many problems, such as the surface mapping problem in [SGB13]. Such an optimization, however,

[3] Carbon Dioxide Information Analysis Center

has $O(m^2|E|)$ variables, which is intractable for dense or large graphs. An open theoretical problem might be to reduce the number of variables asymptotically. Some simplifications may also be afforded using approximations like [PW09], which simplify the form of d_{ij} at the cost of complicating theoretical analysis and understanding of optimal distributions ρ_v. Alternatively, work such as [RDG11] suggests the potential to formulate efficient algorithms when replacing $\mathrm{Prob}([0,1])$ with $\mathrm{Prob}(\mathbb{S}^1)$ or other domains with special structure.

In the end, our proposed algorithms are equally as lightweight as less principled alternatives, while exhibiting practical performance, theoretical soundness, and the possibility of extension into several alternative domains.

Chapter 6

Conclusion

Optimal transportation provides a remarkably powerful language for posing problems in geometric data processing. Previous chapters have written optimization problems involving classical quantities from differential geometry—including geodesic distances, Dirichlet energies, and derivatives of maps—completely within this convex framework. This interaction between mass transportation and classical geometry continues to be explored. For instance, recent work shows how Wasserstein distances are related to measures of curvature [Oll09, LLY11]; this theoretical construction has already been applied to computing the curvature of abstract discrete domains like networks [NLG$^+$15].

From a computational standpoint, even though optimal transportation problems are linear programs, their scaling necessitates development of specialized optimization machinery. Thankfully, problems in this domain are highly structured, and careful study of their interaction with the shortest-path connectivity of the underlying geometric domain helps bring them into the realm of practical computability. In particular, we have leveraged connections between evaluation of and optimization over Wasserstein distances and compressive flow and heat diffusion, when the cost function comes from shortest paths on a manifold. Although these algorithms are marginally less generic than the linear programming formulation, they represent a formidable gain in efficiency for arguably the most practical use cases.

The examples in this thesis show how optimal transportation can be used as a *practical modeling tool* for geometric data rather than simply as a source of theoretically challenging problems. With large-scale techniques for transportation between distributions on geometric domains encountered in computer graphics, geometry processing, learning, and other applications, optimal transportation can be incorporated into a variety of pipelines with confidence that the resulting model will be tractable. This development, suggested not only here but also in works cited throughout our discussion, suggests potential for many additional applications in areas like network analysis, graph theory, robotics, and other fields.

At the same time, it is important to acknowledge problems for which optimal transportation is poorly-suited. Most prominently, the language of mass transportation does not easily express problems that are primarily topological in nature. One intuition for this issue is that the space of probability distributions is path-connected: Given $\mu_0, \mu_1 \in \mathrm{Prob}(M)$, for any $t \in [0,1]$ the measure $(1-t)\mu_0 + t\mu_1$ is also a distribution. Hence, topological invariants like the degree of a map may be difficult to preserve in transportation algorithms.

Beyond additional modeling applications, many algorithmic problems remain for future consideration to explore fully the possibility of transportation-based geometry processing. Even on very structured domains like images and triangle meshes, efficient algorithms for optimal transportation with quadratic costs remain challenging to formulate, without resorting to regularization (e.g. the entropic regularization suggested in Chapter 3). Even more difficult are algorithms for models like propagation, in which nontrivial global structure emerges from local relationships expressed using transportation distances.

With or without these additional developments, transportation remains a practical tool for robust geometry processing, even in the presence of uncertainty and symmetry. By coupling modeling and optimization with fine-grained understanding of geometric structure, optimal transportation provides a natural and intuitive methodology for approaching challenging tasks over curved domains.

Bibliography

[AC11] M. Agueh and G. Carlier, *Barycenters in the Wasserstein space*, SIAM J. Math. Anal. **43** (2011), no. 2, 904–924.

[ADKU11] D. Applegate, T. Dasu, S. Krishnan, and S. Urbanek, *Unsupervised clustering of multidimensional distributions using earth mover distance*, KDD, 2011, pp. 636–644.

[AGS05] L. Ambrosio, N. Gigli, and G. Savaré, *Gradient flows: in metric spaces and in the space of probability measures*, Springer, 2005.

[Arn03] V. Arnold, *Lectures on partial differential equations*, Universitext, Springer, 2003.

[AS00] M. Ashikhmin and P. Shirley, *An anisotropic Phong BRDF model*, J. of Graph. Tools **5** (2000), no. 2, 25–32.

[ASC11] M. Aubry, U. Schlickewei, and D. Cremers, *The wave kernel signature: A quantum mechanical approach to shape analysis*, Proc. ICCV Workshops, Nov 2011, pp. 1626–1633.

[ASP$^+$04] D. Anguelov, P. Srinivasan, H.-C. Pang, D. Koller, S. Thrun, and J. Davis, *The correlated correspondence algorithm for unsupervised registration of nonrigid surfaces*, NIPS, 2004.

[Bar05] S. Bartels, *Stability and convergence of finite-element approximation schemes for harmonic maps*, SIAM journal on numerical analysis **43** (2005), no. 1, 220–238.

[BB00a] J.-D. Benamou and Y. Brenier, *A computational fluid mechanics solution of the Monge-Kantorovich mass transfer problem*, Numerische Mathematik **84** (2000), no. 3, 375–393.

[BB00b] J.-D. Benamou and Y. Brenier, *A computational fluid mechanics solution to the Monge-Kantorovich mass transfer problem*, Numerische Mathematik **84** (2000), no. 3, 375–393.

[BBK06] A. Bronstein, M. Bronstein, and R. Kimmel, *Generalized multidimensional scaling: A framework for isometry-invariant partial surface matching*, PNAS **103** (2006), no. 5, 1168–1172.

[BBK08] ______, *Numerical geometry of non-rigid shapes*, Springer, 2008.

[BBM05] A. Berg, T. Berg, and J. Malik, *Shape matching and object recognition using low distortion correspondences*, CVPR, vol. 1, 2005, pp. 26–33.

[Bc99] R. Burkard and E. Çela, *Linear assignment problems and extensions*, Handbook of Combinatorial Optimization: Supplement **1** (1999), 75.

[BCC$^+$15] J.-D. Benamou, G. Carlier, M. Cuturi, L. Nenna, and G. Peyré, *Iterative Bregman projections for regularized transportation problems*, SIAM J. Sci. Comp., to appear (2015).

[BDM09] R. Burkard, M. Dell'Amico, and S. Martello, *Assignment problems*, SIAM, 2009.

[Bec52] M. Beckmann, *A continuous model of transportation*, Econometrica (1952), 643–660.

[Bli77] J. F. Blinn, *Models of light reflection for computer synthesized pictures*, Proc. SIGGRAPH, vol. 11, 1977, pp. 192–198.

[BN01] M. Belkin and P. Niyogi, *Laplacian eigenmaps and spectral techniques for embedding and clustering*, NIPS, 2001, pp. 585–591.

[BNS06] M. Belkin, P. Niyogi, and V. Sindhwani, *Manifold regularization: A geometric framework for learning from labeled and unlabeled examples*, JMLR **7** (2006), 2399–2434.

[BPC$^+$11] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, *Distributed optimization and statistical learning via the alternating direction method of multipliers*, Found. Trends Mach. Learn. **3** (2011), no. 1, 1–122.

[Bre67] L. M. Bregman, *The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming*, USSR comp. math. and math. physics **7** (1967), no. 3, 200–217.

[BRPP14] N. Bonneel, J. Rabin, G. Peyré, and H. Pfister, *Sliced and Radon Wasserstein barycenters of measures*, J. Math. Imaging and Vision, to appear (2014).

[BvdPPH11] N. Bonneel, M. van de Panne, S. Paris, and W. Heidrich, *Displacement interpolation using Lagrangian mass transport*, ACM Trans. Graph. **30** (2011), no. 6, 158:1–158:12.

[CA14] M. Cuturi and D. Avis, *Ground metric learning*, J. Mach. Learn. Res. **15** (2014), no. 1, 533–564.

[CBW90] C. Connolly, J. B. Burns, and R. Weiss, *Path planning using Laplace's equation*, Proc. Conf. on Robotics and Automation, vol. 3, 1990, pp. 2102–2106.

[CCK07] S.-Y. Chung, Y.-S. Chung, and J.-H. Kim, *Diffusion and elastic equations on networks*, Pub. RIMS **43** (2007), no. 3, 699–726.

[CD14] M. Cuturi and A. Doucet, *Fast computation of Wasserstein barycenters*, Proc. ICML, vol. 32, 2014.

[CHK13] M. Campen, M. Heistermann, and L. Kobbelt, *Practical anisotropic geodesy*, Proc. SGP **32** (2013), no. 5, 63–71.

[Chu96] F. Chung, *Spectral graph theory*, Regional Conference Series, no. 92, Conference Board of the Mathematical Sciences, 1996.

[CL06] R. R. Coifman and S. Lafon, *Diffusion maps*, Applied and Computational Harmonic Anal. **21** (2006), no. 1, 5–30.

[CLL$^+$05] R. R. Coifman, S. Lafon, A. B. Lee, M. Maggioni, B. Nadler, F. Warner, and S. W. Zucker, *Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps*, PNAS **102** (2005), no. 21, 7426–7431.

[CLM$^+$11] W. Chang, H. Li, N. Mitra, M. Pauly, S. Rusinkiewicz, and M. Wand, *Computing correspondences in geometric data sets*, Eurographics Workshop, 2011.

[COO14] G. Carlier, A. Oberman, and E. Oudet, *Numerical methods for matching for teams and Wasserstein barycenters*, Preprint, Ceremade, 2014.

[CR87] J. Canny and J. Reif, *New lower bound techniques for robot motion planning problems*, Found. Comp. Sci., no. 28, 1987, pp. 49–60.

[CT06] T. Cover and J. Thomas, *Elements of information theory*, Wiley, 2006.

[Cut13] M. Cuturi, *Sinkhorn distances: Lightspeed computation of optimal transportation*, Proc. NIPS, vol. 26, 2013, pp. 2292–2300.

[CWW13] K. Crane, C. Weischedel, and M. Wardetzky, *Geodesics in heat: A new approach to computing distance based on heat flow*, ACM Trans. Graph. **32** (2013), no. 5, 152:1–152:11.

[CY00] F. Chung and S.-T. Yau, *Discrete Green's functions*, J. Combinatorial Theory **91** (2000), no. 1–2, 191–214.

[Dav06] T. A. Davis, *Direct methods for sparse linear systems*, SIAM, 2006.

[Del06] J. Delon, *Movie and video scale-time equalization application to flicker reduction*, IEEE Trans. on Image Proc. **15** (2006), no. 1, 241–248.

[Der93] R. Deriche, *Recursively implementing the Gaussian and its derivatives*, Tech. Report RR-1893, 1993.

[dGBOD12] F. de Goes, K. Breeden, V. Ostromoukhov, and M. Desbrun, *Blue noise through optimal transport*, ACM Trans. Graph. **31** (2012), no. 6, 171:1–171:11.

[dGCSAD11] F. de Goes, D. Cohen-Steiner, P. Alliez, and M. Desbrun, *An optimal transport approach to robust reconstruction and simplification of 2d shapes*, Computer Graph. Forum, vol. 30, 2011, pp. 1593–1602.

[dGMMD14] F. de Goes, P. Memari, P. Mullen, and M. Desbrun, *Weighted triangulations for geometry processing*, ACM Trans. Graph. **33** (2014), no. 3.

[DL09] M. M. Deza and M. Laurent, *Geometry of cuts and metrics*, Springer, 2009.

[DMSB99] M. Desbrun, M. Meyer, P. Schröder, and A. H. Barr, *Implicit fairing of irregular meshes using diffusion and curvature flow*, Proc. SIGGRAPH, 1999, pp. 317–324.

[DS40] W. E. Deming and F. F. Stephan, *On a least squares adjustment of a sampled frequency table when the expected marginal totals are known*, Annals Math. Stat. **11** (1940), no. 4, 427–444.

[DT10] A. Dominitz and A. Tannenbaum, *Texture mapping via optimal mass transport*, TVCG **16** (2010), no. 3, 419–433.

[ER11] R. Escalante and M. Raydan, *Alternating projection methods*, Fundamentals of Algorithms, SIAM, 2011.

[FL89] J. Franklin and J. Lorenz, *On the scaling of multidimensional matrices*, Linear Algebra and its Applications **114** (1989), 717–735.

[FM02] M. Feldman and R. McCann, *Monge's transport problem on a Riemannian manifold*, Trans. AMS **354** (2002), no. 4, 1667–1697.

[FPPA14] S. Ferradans, N. Papadakis, G. Peyré, and J.-F. Aujol, *Regularized discrete optimal transport*, SIAM J. Imaging Sci. **7** (2014), no. 3, 1853–1882.

[FPRS07] F. Fouss, A. Pirotte, J.-M. Renders, and M. Saerens, *Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation*, Trans. Knowledge and Data Eng. **19** (2007), no. 3, 355–369.

[GBAL09] K. Gebal, J. Bærentzen, H. Aanæs, and R. Larsen, *Shape analysis using the auto diffusion function*, Comp. Graph. Forum, vol. 28, 2009, pp. 1405–1413.

[HSS08] T. Hofmann, B. Schölkopf, and A. J. Smola, *Kernel methods in machine learning*, The Annals of Statistics **36** (2008), no. 3, 1171–1220.

[HYW00] B. He, H. Yang, and S. Wang, *Alternating direction method with self-adaptive penalty parameters for monotone variational inequalities*, J. Optim. Theory and App. **106** (2000), no. 2, 337–356.

[HZTA04] S. Haker, L. Zhu, A. Tannenbaum, and S. Angenent, *Optimal mass transport for registration and warping*, Int. J. Comp. Vision **60** (2004), no. 3, 225–240.

[IVdAdC11] A. Irpino, R. Verde, and F. de A.T. de Carvalho, *Dynamic clustering of histogram data based on adaptive squared Wasserstein distances*, CoRR **abs/1110.1462** (2011).

[JKO98] R. Jordan, D. Kinderlehrer, and F. Otto, *The variational formulation of the Fokker–Planck equation*, SIAM Journal on Mathematical Analysis **29** (1998), no. 1, 1–17.

[JLYY11] X. Jiang, L.-H. Lim, Y. Yao, and Y. Ye, *Statistical ranking and combinatorial Hodge theory*, Mathematical Programming **127** (2011), no. 1, 203–244.

[Joh77] D. B. Johnson, *Efficient algorithms for shortest paths in sparse networks*, J. ACM **24** (1977), no. 1, 1–13.

[Jos94] J. Jost, *Equilibrium maps between metric spaces*, Calc. Var. and PDE **2** (1994), 173–204.

[JSW05] T. Ju, S. Schaefer, and J. Warren, *Mean value coordinates for closed triangular meshes*, ACM Trans. Graph. **24** (2005), no. 3, 561–566.

[JYL$^+$12] M. Ji, T. Yang, B. Lin, R. Jin, and J. Han, *A simple algorithm for semi-supervised learning with improved generalization error bound*, ICML, 2012.

[Kan42] L. Kantorovich, *On the transfer of masses (in Russian)*, Doklady Akademii Nauk **37** (1942), no. 2, 227–229.

[KFS13] D. Krishnan, R. Fattal, and R. Szeliski, *Efficient preconditioning of Laplacian matrices for computer graphics*, Trans. Graph. **32** (2013), no. 4, 142:1–142:15.

[Kir00] M. Kirby, *Geometric data analysis: An empirical approach to dimensionality reduction and the study of patterns*, Wiley, 2000.

[KLCF10] V. G. Kim, Y. Lipman, X. Chen, and T. Funkhouser, *Möbius transformations for global intrinsic symmetry analysis*, Proc. SGP (2010).

[KLF11] V. Kim, Y. Lipman, and T. Funkhouser, *Blended intrinsic maps*, Proc. SIGGRAPH (2011).

[Kni08] P. Knight, *The Sinkhorn–Knopp algorithm: Convergence and applications*, SIAM J. on Matrix Anal. and Applications **30** (2008), no. 1, 261–275.

[KP13] Y.-H. Kim and B. Pass, *Multi-marginal optimal transport on Riemannian manifolds*, arXiv:1303.6251 (2013).

[KS98] R. Kimmel and J. A. Sethian, *Computing geodesic paths on manifolds*, PNAS, 1998, pp. 8431–8435.

[LD11] Y. Lipman and I. Daubechies, *Conformal Wasserstein distances: Comparing surfaces in polynomial time*, Adv. Math. **227** (2011), no. 3, 1047–1077.

[Léo12] C. Léonard, *From the Schrödinger problem to the Monge-Kantorovich problem*, J. Funct. Anal. **262** (2012), no. 4, 1879–1920.

[LF09] Y. Lipman and T. Funkhouser, *Möbius voting for surface correspondence*, TOG **28** (2009), no. 3, 72:1–72:12.

[LGNL10] G. Lia, L. Guoa, J. Niea, and T. Liu, *An automated pipeline for cortical sulcal fundi extraction*, Medical Image Analysis **14** (2010), no. 3, 343–359.

[Li98] Y. Li, *A Newton acceleration of the Weiszfeld algorithm for minimizing the sum of Euclidean distances*, Comp. Optim. and App. **10** (1998), no. 3, 219–242.

[LLY11] Y. Lin, L. Lu, and S.-T. Yau, *Ricci curvature of graphs*, Tohoku Math. J. (2) **63** (2011), no. 4, 605–627.

[LLZ13] R. Lai, J. Liang, and H.-K. Zhao, *A local mesh method for solving PDEs on point clouds*, Inverse Prob. and Imaging **7** (2013), no. 3, 737–755.

[Low99] D. G. Lowe, *Object recognition from local scale-invariant features*, Proc. ICCV, 1999, pp. 1150–1157.

[LPD11] Y. Lipman, J. Puente, and I. Daubechies, *Conformal Wasserstein distance: II. computational aspects and extensions*, arXiv preprint:1103.4681 (2011).

[LRF10] Y. Lipman, R. Rustamov, and T. Funkhouser, *Biharmonic distance*, TOG **29** (2010), no. 3.

[LZ13] J. Liang and H. Zhao, *Solving partial differential equations on point clouds*, J. Sci. Comp. **35** (2013), no. 3, A1461–A1486.

[Mac49] R. MacNeal, *The solution of partial differential equations by means of electrical networks*, Ph.D. thesis, Caltech, 1949.

[McC97] R. J. McCann, *A convexity principle for interacting gases*, Adv. Math. **128** (1997), no. 1, 153–179.

[Mém07] F. Mémoli, *On the use of Gromov-Hausdorff distances for shape comparison*, Point Based Graphics, 2007, pp. 81–90.

[Mém09] F. Mémoli, *Spectral Gromov-Wasserstein distances for shape matching*, Workshop on Non-Rigid Shape Analysis and Deformable Image Alignment, 2009.

[Mém11] F. Mémoli, *A spectral notion of Gromov-Wasserstein distance and related methods*, Applied and Computational Harmonic Analysis **30** (2011), no. 3, 363–401.

[Mér11] Q. Mérigot, *A multiscale approach to optimal transport*, Comp. Graph. Forum **30** (2011), no. 5, 1583–1592.

[MMdGD11] P. Mullen, P. Memari, F. de Goes, and M. Desbrun, *HOT: Hodge-optimized triangulations*, TOG **30** (2011), no. 4, 103:1–103:12.

[MMP87] J. S. B. Mitchell, D. M. Mount, and C. H. Papadimitriou, *The discrete geodesic problem*, SIAM J. Comput. **16** (1987), no. 4, 647–668.

[MOS14] MOSEK ApS, *Mosek version 7*, `https://mosek.com`, 2014.

[NBCW$^+$11] A. Nguyen, M. Ben-Chen, K. Welnicka, Y. Ye, and L. Guibas, *An optimization approach to improving collections of shape maps*, Comp. Graph. Forum **30** (2011), no. 5, 1481–1491.

[Nea11] R. Neal, *MCMC using Hamiltonian dynamics*, Handbook of Markov Chain Monte Carlo (S. Brooks, A. Gelman, G. Jones, and X.-L. Meng, eds.), CRC Press, 2011.

[NLG$^+$15] C. Ni, Y. Lin, J. Gao, X. D. Gu, and E. Saucan, *Ricci curvature of the internet topology*, CoRR **abs/1501.04138** (2015).

[Oll09] Y. Ollivier, *Ricci curvature of markov chains on metric spaces*, Journal of Functional Analysis **256** (2009), no. 3, 810–864.

[OMMG10] M. Ovsjanikov, Q. Mérigot, F. Mémoli, and L. Guibas, *One point isometric matching with the heat kernel*, CGF, vol. 29, 2010, pp. 1555–1564.

[OSG08] M. Ovsjanikov, J. Sun, and L. Guibas, *Global intrinsic symmetries of shapes*, Comp. Graph. Forum, vol. 27, 2008, pp. 1341–1348.

[PBDSH13] D. Panozzo, I. Baran, O. Diamanti, and O. Sorkine-Hornung, *Weighted averages on surfaces*, TOG **32** (2013), no. 4, 60:1–60:12.

[PH10] M. Pharr and G. Humphreys, *Physically based rendering, second edition: From theory to implementation*, Morgan Kaufmann, July 2010.

[PJ08] S. Pan and Y. Jiang, *Smoothing Newton method for minimizing the sum of p-norms*, J. Optim. Theory and App. **137** (2008), no. 2, 255–275.

[PKD07] F. Pitié, A. C. Kokaram, and R. Dahyot, *Automated colour grading using colour distribution transfer*, Comp. Vision and Image Understanding **107** (2007), 123–137.

[Pla11] F. Plastria, *The Weiszfeld algorithm: Proof, amendments, and extensions*, Found. of Location Anal. (H. A. Eiselt and V. Marianov, eds.), Operations Research & Management Science, vol. 155, 2011, pp. 357–389.

[PP03] K. Polthier and E. Preuß, *Identifying vector field singularities using a discrete Hodge decomposition*, Vis. and Math. III (H.-C. Hege and K. Polthier, eds.), Springer, 2003, pp. 113–134.

[PPC11] N. Papadakis, E. Provenzi, and V. Caselles, *A variational model for histogram transfer of color images*, IEEE Trans. Image Proc. **20** (2011), no. 6, 1682–1695.

[PW09] O. Pele and M. Werman, *Fast and robust earth mover's distances*, ICCV, 2009, pp. 460–467.

[QSZ02] L. Qi, D. Sun, and G. Zhou, *A primal–dual algorithm for minimizing a sum of Euclidean norms*, J. Comp. Applied Math. **138** (2002), no. 1, 127–150.

[RBBK10] D. Raviv, A. Bronstein, M. Bronstein, and R. Kimmel, *Full and partial symmetries of non-rigid shapes*, IJCV **89** (2010), no. 1, 18–39.

[RCB97] A. Rangarajan, H. Chui, and F. Bookstein, *The softassign procrustes matching algorithm*, Information Processing in Medical Imaging, Springer, 1997, pp. 29–42.

[RDG11] J. Rabin, J. Delon, and Y. Gousseau, *Transportation distances on the circle*, J. Math. Imaging Vis. **41** (2011), no. 1–2, 147–167.

[RLF09] R. M. Rustamov, Y. Lipman, and T. Funkhouser, *Interior distance using barycentric coordinates*, Proc. SGP, 2009, pp. 1279–1288.

[RPDB12] J. Rabin, G. Peyre, J. Delon, and M. Bernot, *Wasserstein barycenter and its application to texture mixing*, LNCS, vol. 6667, Springer, 2012, pp. 435–446.

[RR06] B. Roux and H. Rouanet, *Geometric data analysis: From correspondence analysis to structured data analysis*, Kluwer, 2006.

[RTG00] Y. Rubner, C. Tomasi, and L. J. Guibas, *The earth mover's distance as a metric for image retrieval*, Int. J. Comput. Vision **40** (2000), no. 2, 99–121.

[San09] F. Santambrogio, *Absolute continuity and summability of transport densities: simpler proofs and new estimates*, Calc. Var. PDE **36** (2009), no. 3, 343–354.

[San13] ———, *Prescribed-divergence problems in optimal transportation*, MSRI lecture notes, October 2013, 2013.

[Say08] F.-J. Sayas, *A gentle introduction to the finite element method*, 2008.

[SB11] A. Subramanya and J. Bilmes, *Semi-supervised learning with measure propagation*, JMLR **12** (2011), 3311–3370.

[SBCBG11] J. Solomon, M. Ben-Chen, A. Butscher, and L. Guibas, *Discovery of intrinsic primitives on triangle meshes*, Comp. Graph. Forum **30** (2011), no. 2, 365–374.

[SCF10] J. Sun, X. Chen, and T. A. Funkhouser, *Fuzzy geodesics and consistent sparse correspondences for deformable shapes*, Proc. SGP **29** (2010), no. 5, 1535–1544.

[Sch95] G. Schwarz, *Hodge decomposition: a method for solving boundary value problems*, Lecture notes in mathematics, Springer, 1995.

[SCV14] J. Solomon, K. Crane, and E. Vouga, *Laplace-Beltrami: The Swiss army knife of geometry processing*, Symposium on Geometry Processing Graduate School, 2014.

[SD06] A. Stern and M. Desbrun, *Discrete geometric mechanics for variational time integrators*, ACM SIGGRAPH 2006 Courses, 2006, pp. 75–80.

[SdGP$^+$15] J. Solomon, F. de Goes, G. Peyré, M. Cuturi, A. Butscher, A. Nguyen, T. Du, and L. Guibas, *Convolutional Wasserstein distances: Efficient optimal transportation on geometric domains*, ACM Transactions on Graphics (Proc. SIGGRAPH), to appear (2015).

[SGB13] J. Solomon, L. Guibas, and A. Butscher, *Dirichlet energy for analysis and synthesis of soft maps*, Comp. Graph. Forum **32** (2013), no. 5, 197–206.

[Sin64] R. Sinkhorn, *A relationship between arbitrary positive matrices and doubly stochastic matrices*, Annals of Math. Stat. **35** (1964), no. 2, 876–879.

[Sin67] ______, *Diagonal equivalence to matrices with prescribed row and column sums*, American Math. Monthly **74** (1967), no. 4, 402–405.

[SNB$^+$12] J. Solomon, A. Nguyen, A. Butscher, M. Ben-Chen, and L. Guibas, *Soft maps between surfaces*, Comp. Graph. Forum **31** (2012), no. 5, 1617–1626.

[SNZ08] A. Singh, R. D. Nowak, and X. Zhu, *Unlabeled data: Now it helps, now it doesn't*, NIPS, 2008, pp. 1513–1520.

[SOG09] J. Sun, M. Ovsjanikov, and L. Guibas, *A concise and provably informative multiscale signature based on heat diffusion*, SGP, 2009, pp. 1383–1392.

[Sol15] J. Solomon, *Numerical algorithms: Methods for computer vision, machine learning, and graphics*, CRC Press, 2015.

[SRGB14] J. Solomon, R. Rustamov, L. Guibas, and A. Butscher, *Earth mover's distances on discrete surfaces*, ACM Transactions on Graphics (Proc. SIGGRAPH) **33** (2014), no. 4, 67:1–67:12.

[SRLB14] J. Solomon, R. Rustamov, G. Leonidas, and A. Butscher, *Wasserstein propagation for semi-supervised learning*, Proc. ICML, 2014, pp. 306–314.

[SSK$^+$05] V. Surazhsky, T. Surazhsky, D. Kirsanov, S. J. Gortler, and H. Hoppe, *Fast exact and approximate geodesics on meshes*, TOG **24** (2005), no. 3, 553–560.

[STTP14] Y. Schwartzburg, R. Testuz, A. Tagliasacchi, and M. Pauly, *High-contrast computational caustic design*, ACM Trans. Graph. **33** (2014), no. 4, 74:1–74:11.

[SU97] E. Scheinerman and D. Ullman, *Fractional graph theory*, Wiley, 1997.

[TACSD06] Y. Tong, P. Alliez, D. Cohen-Steiner, and M. Desbrun, *Designing quadrangulations with discrete harmonic forms*, Proc. SGP, 2006, pp. 201–210.

[Tak10] Y. Takano, *Metric-preserving reduction of earth mover's distance*, Asia-Pac. J. Op. Res. **27** (2010), no. 1, 39–54.

[TBW$^+$11] A. Tevs, A. Berner, M. Wand, I. Ihrke, and H.-P. Seidel, *Intrinsic shape matching by planned landmark sampling*, CGF **30** (2011), no. 2, 543–552.

[TC09] P. P. Talukdar and K. Crammer, *New regularized algorithms for transductive learning*, ECML-PKDD **5782** (2009), 442–457.

[Var67] S. R. S. Varadhan, *On the behavior of the fundamental solution of the heat equation with variable coefficients*, Comm. on Pure and Applied Math. **20** (1967), no. 2, 431–455.

[Vil03] C. Villani, *Topics in optimal transportation*, Graduate Studies in Mathematics, AMS, 2003.

[vKZHCO11] O. van Kaick, H. Zhang, G. Hamarneh, and D. Cohen-Or, *A survey on shape correspondence*, CGF **30** (2011), no. 6, 1681–1707.

[Wei37] E. Weiszfeld, *Sur le point pour lequel la somme des distances de n points donnés est minimum*, Tôhoku Math. J. **43** (1937), 355–386.

[WL78] F. P. Witte and D. Lucas, *Probabilistic tracking in a multitarget environment*, IEEE Conf. on Decision and Control, vol. 17, 1978, pp. 1212–1216.

[WSSC11] T. Windheuser, U. Schlickwei, F. R. Schimdt, and D. Cremers, *Large-scale integer linear programming for orientation preserving 3D shape matching*, CGF **30** (2011), no. 5, 1471–1480.

[ZB11] X. Zhou and M. Belkin, *Semi-supervised learning by higher order regularization*, ICML **15** (2011), 892–900.

[ZGL03] X. Zhu, Z. Ghahramani, and J. D. Lafferty, *Semi-supervised learning using Gaussian fields and harmonic functions*, 2003, pp. 912–919.

[Zhu08] X. Zhu, *Semi-supervised learning literature survey*, Tech. Report 1530, Computer Sciences, University of Wisconsin-Madison, 2008.

[ZQC$^+$14] B. Zhu, E. Quigley, M. Cong, J. Solomon, and R. Fedkiw, *Codimensional surface tension flow on simplicial complexes*, ACM Transactions on Graphics (Proc. SIGGRAPH) **33** (2014), no. 4, 111:1–111:11.

[ZS08] R. Zass and A. Shashua, *Probabilistic graph and hypergraph matching*, CVPR, 2008, pp. 1–8.

[ZSG$^+$13] X. Zhao, Z. Su, X. D. Gu, A. Kaufman, J. Sun, J. Gao, and F. Luo, *Area-preservation mapping using optimal mass transport*, IEEE Trans. Vis. and Comp. Graphics **19** (2013), no. 12.

[ZTS03] G. Zhou, K. Toh, and D. Sun, *Globally and quadratically convergent algorithm for minimizing the sum of Euclidean norms*, J. Optim. Theory and App. **119** (2003), no. 2, 357–377.