Convolutional Wasserstein Distances: Efficient Optimal Transportation on Geometric Domains

Justin Solomon
Stanford University

Fernando de Goes
Pixar Animation Studios

Gabriel Peyré
CNRS & Univ. Paris-Dauphine

Marco Cuturi
Kyoto University

Adrian Butscher
Autodesk, Inc.

Andy Nguyen
Stanford University

Tao Du
Stanford University

Leonidas Guibas
Stanford University

Figure 1: *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 §7.*

Abstract

This paper introduces a new class of algorithms for optimization problems involving optimal transportation over geometric domains. Our main contribution is to show that optimal transportation can be made tractable over large domains used in graphics, such as images and triangle meshes, improving performance by orders of magnitude compared to previous work. To this end, we approximate optimal transportation distances using entropic regularization. The resulting objective contains a geodesic distance-based kernel that can be approximated with the heat kernel. This approach leads to simple iterative numerical schemes with linear convergence, in which each iteration only requires Gaussian convolution or the solution of a sparse, pre-factored linear system. We demonstrate the versatility and efficiency of our method on tasks including reflectance interpolation, color transfer, and geometry processing.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry & Object Modeling—Geometric algorithms, languages, & systems

Keywords: Optimal transportation, Wasserstein distances, entropy, displacement interpolation.

1 Introduction

Probability distributions are ubiquitous objects in computer graphics, used to encapsulate possibly uncertain information associated with arbitrary geometric domains. Examples include image histograms, geometric features, relaxations of correspondence maps, and even physical quantities like BRDFs. To compare these objects, it is important to define an adequate notion of proximity or coverage quantifying the discrepancy or, equivalently, similarity between distributions. These computations are commonly posed and analyzed within the theory of *optimal transportation*.

The prototypical problem in optimal transportation is the evaluation of Wasserstein (also known as Earth Mover's) distances between distributions [Villani 2003; Rubner et al. 2000]. These distances quantify the geometric discrepancy between two distributions by measuring the minimal amount of "work" needed to move all the mass contained in one distribution onto the other. Recent developments show that incorporating these 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 is a Dirac at the midpoint $(x+y)/2$, thus offering a geometric notion of the midpoint of two distributions.

A limiting factor in optimal transportation is the complexity of the underlying minimization problem. The usual linear program describing optimal transportation is related to minimum-cost matching, with a quadratic number of variables and time complexity scaling at least cubically in the size of the domain [Burkard and Çela 1999]. This poor complexity is largely due to the use of coupling variables representing the amount of mass transported between every pair of samples. Hence, existing large-scale methods often resort to aggressive or ad-hoc approximations that can lose connections to transportation theory or compensate with alternative formulations that apply only to restricted cases.

This paper introduces a fast, scalable numerical framework for optimal transportation over geometric domains. Our work draws insight from recent advances in machine learning approximating optimal transportation distances using entropic regularization [Cuturi 2013]. 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 correspon-

dence 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.

2 Related Work

The original formulation of optimal transportation, introduced in [Kantorovich 1942], involves a linear program connecting a pair of distributions. The cost of moving density from one point to another is specified using a fixed matrix of pairwise costs. As outlined in [Burkard et al. 2009], 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 [Bonneel et al. 2011] and in [Lipman and Daubechies 2011]. 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.

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 [Mérigot 2011; Schwartzburg et al. 2014] or through Newton iterations on Euclidean spaces [de Goes et al. 2012; Zhao et al. 2013]. More recently, this Newton-based approach for optimal transportation was extended to discrete surfaces [de Goes et al. 2014]. 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 [de Goes et al. 2011].

Another line of work proposes a dynamical formulation for optimal transportation with an additional time variable. For squared distance costs, Benamou and Brenier [2000] compute transportation distances by minimizing the cost of advecting one distribution to another in time. For non-squared distance costs, Solomon et al. [2014a] solve for transportation maps as the flow of a vector field whose divergence matches the difference between the input densities.

Other methods use optimal transportation to aggregate and average information from multiple densities. Examples include barycenter computation [Agueh and Carlier 2011], density propagation over graphs [Solomon et al. 2014b], and computation of "soft" correspondence maps [Solomon et al. 2012]. These problems are typically solved via a multi-marginal linear program [Agueh and Carlier 2011; Kim and Pass 2013], which is infeasible for large-scale domains. One work-around approaches the dual of the linear program using L-BFGS with subgradient directions [Carlier et al. 2014], but this strategy suffers from poor conditioning and noisy results.

Regularization provides a promising way to approximate solutions of transportation problems. 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 [Cuturi 2013]. With entropic regularization, optimal transportation is solved using an iterative scaling method known as the iterative proportional fitting procedure (IPFP) or Sinkhorn-Knopp algorithm [Deming and Stephan 1940; Sinkhorn 1967], which can be implemented in parallel GPGPU architectures and used to compute e.g. the barycenter of thousands of distributions [Cuturi and Doucet 2014].

Our work leverages the efficiency of iterative scaling methods for entropy-regularized transport and related problems, principally [Cuturi 2013; Benamou et al. 2015]. 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 Preliminaries

We begin with background on optimal transportation. 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 $\mathrm{Prob}(M)$ to indicate the space of probability measures on M and $\mathrm{Prob}(M \times M)$ to refer to probability measures on the *product space* of M with itself. To avoid confusion, we will refer to elements $\mu_0, \mu_1, ... \in \mathrm{Prob}(M)$ as *marginals* and to joint probabilities $\pi_0, \pi_1, ... \in \mathrm{Prob}(M \times M)$ as *couplings*.

3.1 Optimal Transportation

A source marginal μ_0 can be transformed into a target marginal μ_1 by means of a *transportation plan* π, a coupling in $\mathrm{Prob}(M \times M)$ describing the amount of mass $\pi(x, y)$ to be displaced from μ_0 at x towards y to create μ_1 in aggregate. Mass conservation laws impose that such couplings are necessarily in the set

$$\Pi(\mu_0, \mu_1) \stackrel{\text{def.}}{=} \{\pi \in \mathrm{Prob}(M \times M) : \pi(\cdot, M) = \mu_0, \pi(M, \cdot) = \mu_1\}.$$

The optimal transportation problem from μ_0 to μ_1 seeks a coupling $\pi \in \Pi(\mu_0, \mu_1)$ with minimal cost, computed as the integral of squared distances d^2 against π. Formally, the 2-Wasserstein distance between μ_0 and μ_1 is thus defined as

$$\mathcal{W}_2(\mu_0, \mu_1) \stackrel{\text{def.}}{=} \left[\inf_{\pi \in \Pi(\mu_0, \mu_1)} \iint_{M \times M} d(x, y)^2 \, \mathrm{d}\pi(x, y)\right]^{1/2}. \quad (1)$$

The 2-Wasserstein distance satisfies all metric axioms and has several attractive properties—see [Villani 2003, §7] for details.

3.2 Kullback-Leibler Divergence

The modified transportation problems we consider involve quantities from information theory, whose definitions we recall below. We refer the reader to [Cover and Thomas 2006] for detailed discussions.

A coupling π is *absolutely continuous* with respect to the volume measure when it admits a density function p, so that $\pi(U) = \int_U p(x, y) \, \mathrm{d}x \, \mathrm{d}y \, , \forall U \subseteq M \times M$. 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

$$H(\pi) \stackrel{\text{def.}}{=} -\iint_{M \times M} \pi(x, y) \ln \pi(x, y) \, \mathrm{d}x \, \mathrm{d}y. \quad (2)$$

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

$$\mathrm{KL}(\pi | \mathcal{K}) \stackrel{\text{def.}}{=} \iint_{M \times M} \pi(x, y) \left[\ln \frac{\pi(x, y)}{\mathcal{K}(x, y)} - 1\right] \mathrm{d}x \, \mathrm{d}y. \quad (3)$$

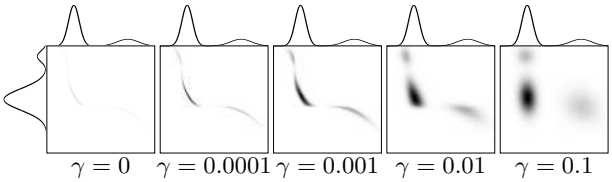


Figure 2: *Transportation plans with different values of γ, with 1D quadratic costs; $\mu_0, \mu_1 \in \mathrm{Prob}([0,1])$ are shown on the axes.*

4 Regularized Optimal Transportation

In this section, we present a modification of 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.

4.1 Entropy-Regularized Wasserstein Distance

Following e.g. [Cuturi 2013; Benamou et al. 2015], we modify the objective of the optimal transportation problem in (1) by adding an entropy term $H(\pi)$ promoting spread-out transportation plans π. The *entropy-regularized* 2*-Wasserstein distance* is then defined as:

$$\mathcal{W}^2_{2,\gamma}(\mu_0,\mu_1) \stackrel{\text{def.}}{=} \inf_{\pi\in\Pi}\left[\iint_{M\times M} d(x,y)^2\,\pi(x,y)\mathrm{d}x\mathrm{d}y - \gamma H(\pi)\right], \quad (4)$$

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 [Léonard 2012] for discussion of its connection to non-regularized transport, recovered as $\gamma \to 0$.

When $\gamma > 0$, the solution π to (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. Fig. 2 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:

$$\mathcal{K}_\gamma(x,y) = e^{-d(x,y)^2/\gamma}, \quad d(x,y)^2 = -\gamma \ln \mathcal{K}_\gamma(x,y). \quad (5)$$

By combining (3), (4) and (5) algebraically, the entropy-regularized Wasserstein distance can be computed from the smallest KL divergence from a coupling $\pi \in \Pi$ to the kernel $\mathcal{K}_\gamma$:

$$\mathcal{W}^2_{2,\gamma}(\mu_0,\mu_1) = \gamma\left[1 + \min_{\pi\in\Pi}\mathrm{KL}(\pi|\mathcal{K}_\gamma)\right]. \quad (6)$$

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 [Cuturi 2013; Benamou et al. 2015], 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.

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

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

Similar to [Crane et al. 2013], we associate the heat kernel $\mathcal{H}_t$ to the geodesic distance function $d(\cdot,\cdot)$ based on the Varadhan's formula [1967], which states that the distance $d(x,y)$ can be recovered by transferring heat from x to y over a short time interval:

$$d(x,y)^2 = \lim_{t\to 0}\left[-2t\ln\mathcal{H}_t(x,y)\right]. \quad (7)$$

Setting $t \stackrel{\text{def.}}{=} \gamma/2$ in (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.:

$$\mathcal{W}^2_{2,\mathcal{H}_{\gamma/2}}(\mu_0,\mu_1) \stackrel{\text{def.}}{=} \gamma\left[1 + \min_{\pi\in\Pi}\mathrm{KL}(\pi|\mathcal{H}_{\gamma/2})\right]. \quad (8)$$

Developing conditions for convergence of $\mathcal{W}^2_{2,\mathcal{H}_{\gamma/2}}$ as $\gamma \to 0$ is a challenging topic for future research. Note that while derivatives of distances from (7) can diverge near the cut locus [Malliavin and Stroock 1996], distance values are valid everywhere on M provided M is connected and compact; divergence of derivatives is not problematic for our method.

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 [Cuturi 2013, Theorem 1]). Hence, as in [Crane et al. 2013], 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.

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 $\mathbf{f} \in \mathbb{R}^n$.
- A vector $\mathbf{a} \in \mathbb{R}^n_+$ of "area weights," with $\mathbf{a}^\top\mathbf{1} = 1$, defined so that
$$\int_M f(x)\,\mathrm{d}x \approx \mathbf{a}^\top\mathbf{f}.$$
- A symmetric matrix $\mathbf{H}_t$ discretizing the kernel $\mathcal{H}_t$ such that
$$\int_M f(y)\mathcal{H}_t(\cdot,y)\mathrm{d}y \;\approx\; \mathbf{H}_t(\mathbf{a}\otimes\mathbf{f}).$$
It is sufficient to know how to *apply* $\mathbf{H}_t$ 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, $\mathbf{a} \overset{\text{def.}}{=} \mathbf{1}/n_1 n_2$ and $\mathbf{H}_t$ 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 $\mathbf{a}$ as lumped areas proportional to the sum of triangle areas adjacent to a given vertex. Given the cotangent Laplacian $\mathbf{L} \in \mathbb{R}^{n \times n}$ [MacNeal 1949] and a diagonal area matrix $\mathbf{D_a}$ ($\mathbf{D_v}$ denotes the diagonal matrix with elements in vector $\mathbf{v}$), we discretize the heat kernel by solving the diffusion equation via an implicit Euler integration [Desbrun et al. 1999] with time step $t = \gamma/2$, i.e.,

$$\mathbf{w} = \mathbf{H}_t(\mathbf{a} \otimes \mathbf{v}) \iff (\mathbf{D_a} + \gamma/2\mathbf{L})\,\mathbf{w} = \mathbf{a} \otimes \mathbf{v}.$$

$\mathbf{D_a} + \gamma/2\mathbf{L}$ 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 [Davis 2006] with γ proportional to the maximum edge length [Crane et al. 2013]; 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 $\mu \in \mathrm{Prob}(M)$ as a vector $\boldsymbol{\mu} \in \mathbb{R}^n_+$ with $\mathbf{a}^\top \boldsymbol{\mu} = 1$ and a distribution $\pi \in \mathrm{Prob}(M \times M)$ as $\boldsymbol{\pi} \in \mathbb{R}^{n \times n}_+$ with $\mathbf{a}^\top \boldsymbol{\pi} \mathbf{a} = 1$. The discrete KL divergence between a discrete distribution $\boldsymbol{\pi}$ and an arbitrary $\mathbf{H} \in \mathbb{R}^{n \times n}_{+,*}$ is then defined as

$$\mathrm{KL}(\boldsymbol{\pi}|\mathbf{H}) \overset{\text{def.}}{=} \sum_{ij} \boldsymbol{\pi}_{ij} \mathbf{a}_i \mathbf{a}_j \left[\ln \frac{\boldsymbol{\pi}_{ij}}{\mathbf{H}_{ij}} - 1 \right]. \tag{9}$$

Given discrete distributions $\boldsymbol{\mu}_0$ and $\boldsymbol{\mu}_1$, we model plans $\boldsymbol{\pi} \in \Pi(\boldsymbol{\mu}_0, \boldsymbol{\mu}_1)$ as matrices $\boldsymbol{\pi} \in \mathbb{R}^{n \times n}_+$ with $\boldsymbol{\pi}\mathbf{a} = \boldsymbol{\mu}_0$ and $\boldsymbol{\pi}^\top \mathbf{a} = \boldsymbol{\mu}_1$. Finally, the convolutional Wasserstein distance is computed via

$$\boxed{\mathcal{W}^2_{2,\mathbf{H}_t}(\boldsymbol{\mu}_0, \boldsymbol{\mu}_1) \overset{\text{def.}}{=} \gamma \left[1 + \min_{\boldsymbol{\pi} \in \Pi} \mathrm{KL}(\boldsymbol{\pi}|\mathbf{H}_t) \right].} \tag{10}$$

Similarly to the continuous case, the minimization in (10) is convex with linear constraints on $\boldsymbol{\pi}$. Its complexity is tied to the variable $\boldsymbol{\pi}$, which scales quadratically in n. As shown in the supplemental document, we overcome this issue using the following result:

Proposition 1. *The transportation plan $\boldsymbol{\pi} \in \Pi(\boldsymbol{\mu}_0, \boldsymbol{\mu}_1)$ minimizing* (10) *is of the form $\boldsymbol{\pi} = \mathbf{D_v} \mathbf{H}_t \mathbf{D_w}$, with unique vectors $\mathbf{v}, \mathbf{w} \in \mathbb{R}^n$ satisfying*

$$\begin{cases} \mathbf{D_v} \mathbf{H}_t \mathbf{D_w} \mathbf{a} = \boldsymbol{\mu}_0, \\ \mathbf{D_w} \mathbf{H}_t \mathbf{D_v} \mathbf{a} = \boldsymbol{\mu}_1. \end{cases} \tag{11}$$

Therefore, rather than computing a matrix $\boldsymbol{\pi}$, we can instead compute a pair of vectors $(\mathbf{v}, \mathbf{w})$, reducing the number of unknowns to $2n$. This proposition generalizes a result in [Cuturi 2013] with the introduction of area weights $\mathbf{a}$. We can find $(\mathbf{v}, \mathbf{w})$ by alternating projections onto the linear marginal constraints via an area-weighted version of *Sinkhorn's algorithm* [1964], detailed in Algorithm 1.

As in [Solomon et al. 2014a], $\mathcal{W}^2_{2,\mathbf{H}_t}$ between distributions centered at individual vertices can be used as point-to-point distances. Fig. 3 shows one example computed using our algorithm. The resulting pointwise distance squared is exactly the logarithm of $\mathbf{H}_t$. Since Crane et al. [2013] 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.

function CONVOLUTIONAL-WASSERSTEIN($\boldsymbol{\mu}_0, \boldsymbol{\mu}_1; \mathbf{H}_t, \mathbf{a}$)

// Sinkhorn iterations
$\mathbf{v}, \mathbf{w} \leftarrow \mathbf{1}$
for $i = 1, 2, 3, \ldots$
  $\mathbf{v} \leftarrow \boldsymbol{\mu}_0 \oslash \mathbf{H}_t(\mathbf{a} \otimes \mathbf{w})$
  $\mathbf{w} \leftarrow \boldsymbol{\mu}_1 \oslash \mathbf{H}_t(\mathbf{a} \otimes \mathbf{v})$

// KL divergence
return $\gamma\, \mathbf{a}^\top [(\boldsymbol{\mu}_0 \otimes \ln \mathbf{v}) + (\boldsymbol{\mu}_1 \otimes \ln \mathbf{w})]$

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

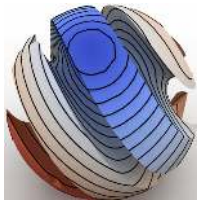
(a)   (b)

Figure 3: *$\mathcal{W}^2_{2,\mathbf{H}_t}$ between δ distributions (a) as a vertex-to-vertex distance (b; computed with $\gamma = 10^{-5}$ — slight smoothing).*

Timing & numerics. To evaluate efficiency, we compare three approaches to approximating $\mathcal{W}_2$: a linear program discretizing (1), regularized distances with a full distance-based kernel [Cuturi 2013], and convolutional Wasserstein distances $\mathcal{W}^2_{2,\mathbf{H}_t}$. The linear program is solved using state-of-the-art parallel optimization [MOSEK ApS 2014], with all-pairs distances along mesh edges from an $O(n^2 \log n)$ algorithm [Johnson 1977]. [Cuturi 2013] and our 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 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 [Cuturi 2013] 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 [Franklin and Lorenz 1989; Knight 2008], and similar guarantees exist for alternating projection methods [Escalante and Raydan 2011]. These bounds give a rough indicator of the number of iterations needed to compute convolutional distances and derived quantities used in §6. In practice, the convergence rate depends on the sharpness of the kernel and of the distributions $\boldsymbol{\mu}_0$ and $\boldsymbol{\mu}_1$. The experiments reported in Table 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.

$\|V\|$	$\|T\|$	PD	LP	[Cuturi 2013]	PF	$\mathcal{W}^2_{2,\mathbf{H}_t}$
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 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 [Cuturi 2013]; timing for this step is written separately. Cholesky pre-factorization (PF) is needed for convolutional distance and is similarly separated.*

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.

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 [Agueh and Carlier 2011], 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

$$\min_\mu \sum_{i=1}^k \alpha_i \mathcal{W}_2^2(\mu, \mu_i). \tag{12}$$

After discretization, we can pose the barycenter problem as

$$\min_{\boldsymbol{\mu}} \sum_{i=1}^k \alpha_i \mathcal{W}_{2,\mathbf{H}_t}^2(\boldsymbol{\mu}, \boldsymbol{\mu}_i). \tag{13}$$

Substituting transportation plans yields an equivalent problem:

$$\begin{array}{rl} \min_{\{\boldsymbol{\pi}_i\}} & \sum_{i=1}^k \alpha_i \mathrm{KL}(\boldsymbol{\pi}_i | \mathbf{H}_t) \\ \text{s.t.} & \boldsymbol{\pi}_i^\top \mathbf{a} = \boldsymbol{\mu}_i \; \forall i \in \{1, \dots, k\} \\ & \boldsymbol{\pi}_i \mathbf{a} = \boldsymbol{\pi}_1 \mathbf{a} \; \forall i \in \{1, \dots, k\} \end{array}$$

The first constraint enforces that $\boldsymbol{\pi}_i$ marginalizes to $\boldsymbol{\mu}_i$ in one direction, and the second constraint enforces that all the $\boldsymbol{\pi}_i$'s marginalize to the same $\boldsymbol{\mu}$ in the opposite direction.

As suggested by Benamou et al. [2015], the expanded problem can be viewed as a *projection* with respect to KL divergence from $\mathbf{H}_t$ (repeated k times) onto the constraint set $\mathcal{C}_1 \cap \mathcal{C}_2$, where

$$\mathcal{C}_1 \stackrel{\text{def.}}{=} \{(\boldsymbol{\pi}_i)_{i=1}^k : \boldsymbol{\pi}_i^\top \mathbf{a} = \boldsymbol{\mu}_i \; \forall i \in \{1, \dots, k\}\}$$
$$\mathcal{C}_2 \stackrel{\text{def.}}{=} \{(\boldsymbol{\pi}_i)_{i=1}^k : \boldsymbol{\pi}_i \mathbf{a} = \boldsymbol{\pi}_j \mathbf{a} \; \forall i, j \in \{1, \dots, k\}\}.$$

Problems of this form can be minimized using *iterated Bregman projection* [Bregman 1967], which initializes all the $\boldsymbol{\pi}_i$'s to $\mathbf{H}_t$ 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 2. *The KL projection of $(\boldsymbol{\pi}_i)_{i=1}^k$ onto $\mathcal{C}_1$ satisfies $\mathrm{proj}_{\mathcal{C}_1} \boldsymbol{\pi}_i = \boldsymbol{\pi}_i \mathbf{D}_{\boldsymbol{\mu}_i \oslash \boldsymbol{\pi}_i^\top \mathbf{a}}$ for each $i \in \{1, \dots k\}$.*

Proposition 3. *The KL projection of $(\boldsymbol{\pi}_i)_{i=1}^k$ onto $\mathcal{C}_2$ satisfies $\mathrm{proj}_{\mathcal{C}_2} \boldsymbol{\pi}_i = \mathbf{D}_{\boldsymbol{\mu} \oslash \mathbf{d}_i} \boldsymbol{\pi}_i$ for each $i \in \{1, \dots k\}$, where $\mathbf{d}_i = \boldsymbol{\pi}_i \mathbf{a}$ and $\boldsymbol{\mu} = \prod_i \mathbf{d}_i^{\alpha_i / \sum_\ell \alpha_\ell}$.*

The propositions, originally presented without area weights in [Benamou et al. 2015] and proved similarly in our supplemental document, 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 $\mathbf{v}_i, \mathbf{w}_i \in \mathbb{R}^n$ so that $\boldsymbol{\pi}_i = \mathbf{D}_{\mathbf{v}_i} \mathbf{H}_t \mathbf{D}_{\mathbf{w}_i}$. If M is represented using n samples, this reduces storage and algorithmic runtime by a factor of n.

function WASSERSTEIN-BARYCENTER($\{\boldsymbol{\mu}_i\}, \{\alpha_i\}; \mathbf{H}_t, \mathbf{a}$)
  // Initialization
  $\mathbf{v}_1, \dots, \mathbf{v}_k \leftarrow \mathbf{1}$
  $\mathbf{w}_1, \dots, \mathbf{w}_k \leftarrow \mathbf{1}$
  // Iterate over $\mathcal{C}_i$'s
  for $j = 1, 2, 3, \dots$
    $\boldsymbol{\mu} \leftarrow \mathbf{1}$
    for $i = 1, \dots, k$
      // Project onto $\mathcal{C}_1$
      $\mathbf{w}_i \leftarrow \boldsymbol{\mu}_i \oslash \mathbf{H}_t(\mathbf{a} \otimes \mathbf{v}_i)$
      $\mathbf{d}_i \leftarrow \mathbf{v}_i \otimes \mathbf{H}_t(\mathbf{a} \otimes \mathbf{w}_i)$
      $\boldsymbol{\mu} \leftarrow \boldsymbol{\mu} \otimes \mathbf{d}_i^{\alpha_i}$
    // Optional
    $\boldsymbol{\mu} \leftarrow$ ENTROPIC-SHARPENING($\boldsymbol{\mu}, H_0; \mathbf{a}$)
    // Project onto $\mathcal{C}_2$
    for $i = 1, \dots, k$
      $\mathbf{v}_i \leftarrow \mathbf{v}_i \otimes \boldsymbol{\mu} \oslash \mathbf{d}_i$
  return $\boldsymbol{\mu}$

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

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

$$H(\boldsymbol{\mu}) \stackrel{\text{def.}}{=} -\sum_i \mathbf{a}_i \boldsymbol{\mu}_i \ln \boldsymbol{\mu}_i.$$

We expect the non-regularized Wasserstein barycenter of a set of distributions to have entropy bounded by that of the input distributions $(\boldsymbol{\mu}_i)_{i=1}^k$. Hence, take $H_0 \stackrel{\text{def.}}{=} \max_i H(\boldsymbol{\mu}_i)$ (or a user-specified bound). Then, we can modify the barycenter problem slightly:

$$\begin{array}{rl} \min_{\boldsymbol{\mu}} & \sum_{i=1}^k \alpha_i \mathcal{W}_{2,\mathbf{H}_t}^2(\boldsymbol{\mu}, \boldsymbol{\mu}_i) \\ \text{s.t.} & H(\boldsymbol{\mu}) \leq H_0. \end{array} \tag{14}$$

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

The problem in (14) is not convex, but we apply Bregman projections nonetheless. We augment $\mathcal{C}_2$ with an entropy constraint:

$$\overline{\mathcal{C}}_2 \stackrel{\text{def.}}{=} \mathcal{C}_2 \cap \{(\boldsymbol{\pi}_i)_{i=1}^k : H(\boldsymbol{\pi}_i \mathbf{a}) + \mathbf{a}^\top \boldsymbol{\pi}_i \mathbf{a} \leq H_0 + 1 \; \forall i \in \{1, \dots, k\}\}$$

function ENTROPIC-SHARPENING($\boldsymbol{\mu}, H_0; \mathbf{a}$)
  if $H(\boldsymbol{\mu}) + \mathbf{a}^\top \boldsymbol{\mu} > H_0 + 1$ **then**
    $\beta \leftarrow$ FIND-ROOT($\mathbf{a}^\top \boldsymbol{\mu}^\beta + H(\boldsymbol{\mu}^\beta) - (1 + H_0); \beta \geq 0$)
  else $\beta \leftarrow 1$
  return $\boldsymbol{\mu}^\beta$

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

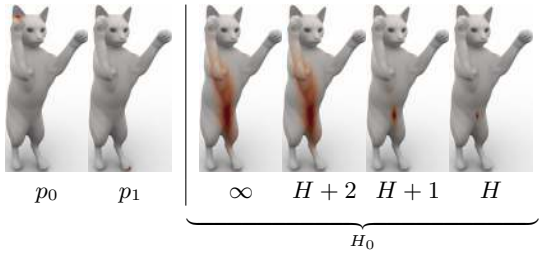


Figure 4: *Barycenters with different levels of entropic sharpening, controlled by H_0. Here, $H \stackrel{\text{def.}}{=} \max\{H(\boldsymbol{\mu}_1), H(\boldsymbol{\mu}_2)\} \approx -2.569$.*

The $\mathbf{a}^\top \boldsymbol{\pi}_i \mathbf{a}$ term is for algebraic convenience in proving the proposition below; at convergence, $\mathbf{a}^\top \boldsymbol{\pi}_i \mathbf{a} = 1$ 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, as proved in the supplemental document:

Proposition 4. *There exists $\beta \in \mathbb{R}$ such that the KL projection of $(\boldsymbol{\pi}_i)_{i=1}^k$ onto $\overline{\mathcal{C}}_2$ satisfies $\text{proj}_{\overline{\mathcal{C}}_2} \boldsymbol{\pi}_i = \mathbf{D}_{\boldsymbol{\mu} \oslash \mathbf{d}_i} \boldsymbol{\pi}_i$ for all $i \in \{1, \ldots, k\}$, where $\mathbf{d}_i = \boldsymbol{\pi}_i \mathbf{a}$ and $\boldsymbol{\mu} = \left(\prod_i \mathbf{d}_i^{\alpha_i}\right)^\beta$.*

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 $\boldsymbol{\mu}_i$'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.

Fig. 4 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.

6.2 Displacement Interpolation

The 2-Wasserstein distance $\mathcal{W}_2$ has a distinguishing *displacement interpolation* property [McCann 1997]. $\mathcal{W}_2(\mu_0, \mu_1)$ 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 paths on M. As a point of comparison, Solomon et al. [2014a] use 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 [2011] prove under suitable regularity that the interpolating path μ_t from μ_0 to μ_1 satisfies

$$\mu_t = \inf_{\mu \in \text{Prob}(M)} \left[(1 - t)\mathcal{W}_2^2(\mu_0, \mu) + t\mathcal{W}_2^2(\mu, \mu_1)\right], \qquad (15)$$

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 $\boldsymbol{\mu}_0, \boldsymbol{\mu}_1 \in \text{Prob}(M)$ we wish to find a time-varying $\boldsymbol{\mu}_t$ interpolating between the two. To do so, we define

$$\boldsymbol{\mu}_t \stackrel{\text{def.}}{=} \min_{\boldsymbol{\mu} \in \text{Prob}(M)} \left[(1 - t)\mathcal{W}_{2,\mathbf{H}_t}^2(\boldsymbol{\mu}_0, \boldsymbol{\mu}) + t\mathcal{W}_{2,\mathbf{H}_t}^2(\boldsymbol{\mu}, \boldsymbol{\mu}_1)\right]. \qquad (16)$$

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

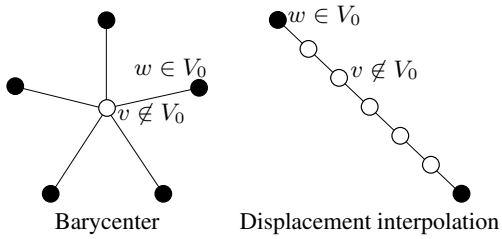


Figure 5: *Wasserstein propagation can be used to model barycenter problems and displacement interpolation. Here, we show the corresponding graph $G = (V, E)$; vertices in V_0 have solid shading.*

Fig. 6 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.

6.3 Wasserstein Propagation

Generalizing the barycenter problem, we consider the "Wasserstein propagation" problem posed by Solomon et al. [2014b]. 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 $\mu_v \in \text{Prob}(M)$, 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 $v \in V_0$.

function WASSERSTEIN-PROPAGATION($V, E, V_0, \boldsymbol{\mu}(V_0); \mathbf{H}_t, \mathbf{a}$)
  // Initialization
  $\mathbf{v}_1, \ldots, \mathbf{v}_{|E|} \leftarrow \mathbf{1}$
  $\mathbf{w}_1, \ldots, \mathbf{w}_{|E|} \leftarrow \mathbf{1}$
  // Iterate over $\mathcal{C}_i$'s
  for $j = 1, 2, 3, \ldots$
    for $v \in V$
      if $v \in V_0$ **then**
        $\boldsymbol{\mu} \leftarrow \boldsymbol{\mu}_0(v)$
        // Project adjacent $\boldsymbol{\pi}_e$'s
        for $e \in N(v)$
          if $e = (w, v)$ **then** $\mathbf{w}_e \leftarrow \boldsymbol{\mu} \oslash \mathbf{H}_t(\mathbf{a} \otimes \mathbf{v}_e)$
          if $e = (v, w)$ **then** $\mathbf{v}_e \leftarrow \boldsymbol{\mu} \oslash \mathbf{H}_t(\mathbf{a} \otimes \mathbf{w}_e)$
      else if $v \notin V_0$ **then**
        // Estimate distribution
        $\omega \leftarrow \sum_{v \in e} \alpha_e$
        $\boldsymbol{\mu}_v \leftarrow \mathbf{1}$
        for $e \in N(v)$
          if $e = (w, v)$ **then** $\mathbf{d}_e \leftarrow \mathbf{w}_e \otimes \mathbf{H}_t(\mathbf{a} \otimes \mathbf{v}_e)$
          if $e = (v, w)$ **then** $\mathbf{d}_e \leftarrow \mathbf{v}_e \otimes \mathbf{H}_t(\mathbf{a} \otimes \mathbf{w}_e)$
          $\boldsymbol{\mu}_v \leftarrow \boldsymbol{\mu}_v \otimes \mathbf{d}_e^{\alpha_e / \omega}$
        for $e \in N(v)$
          if $e = (w, v)$ **then** $\mathbf{w}_e \leftarrow \mathbf{w}_e \otimes \boldsymbol{\mu}_v \oslash \mathbf{d}_e$
          if $e = (v, w)$ **then** $\mathbf{v}_e \leftarrow \mathbf{v}_e \otimes \boldsymbol{\mu}_v \oslash \mathbf{d}_e$
  return $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_{|V|}$

Algorithm 4: *Wasserstein propagation via Bregman projection.*

As an example, as proposed in [Solomon et al. 2012], 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. Given ground-truth correspondences for a few vertices, the optimization above fills in missing data.

Propagation can be modeled using convolutional distances as

$$\begin{array}{rl} \min_{\boldsymbol{\mu}_v} & \sum_{(v,w)\in E} \alpha_{(v,w)} \mathcal{W}^2_{2,\mathbf{H}_t}(\boldsymbol{\mu}_v, \boldsymbol{\mu}_w) \\ \text{s.t.} & \boldsymbol{\mu}_v \text{ fixed } \forall v \in V_0. \end{array} \tag{17}$$

Following the optimizations in previous sections, we instead optimize over transportation matrices $\boldsymbol{\pi}_e$ for each $e \in E$:

$$\begin{array}{rl} \min_{\boldsymbol{\pi}_e} & \sum_{e\in E} \alpha_{(v,w)} \mathrm{KL}(\boldsymbol{\pi}_e | \mathbf{H}_t) \\ \text{s.t.} & \boldsymbol{\pi}_e \mathbf{a} = \boldsymbol{\mu}_v \ \forall e = (v,w) \\ & \boldsymbol{\pi}_e^\top \mathbf{a} = \boldsymbol{\mu}_w \ \forall e = (v,w) \\ & \boldsymbol{\mu}_v \text{ fixed } \forall v \in V_0. \end{array}$$

The interpolated $\boldsymbol{\mu}$'s will be distributions because they must have the same integrals as the $\boldsymbol{\mu}$'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 2 and 3, we can write $\boldsymbol{\pi}_e = \mathbf{D}_{\mathbf{v}_e} \mathbf{H}_t \mathbf{D}_{\mathbf{w}_e}$ and update the $\mathbf{v}_e$'s and $\mathbf{w}_e$'s using simple rules.

Propagation encapsulates many other optimizations in Wasserstein space. Fig. 5 illustrates two examples. The convolutional barycenter problem (§6.1) is exactly propagation where G is a star graph, with vertices in V_0 on the spokes and the unknown distribution $\boldsymbol{\mu}$ associated with the center. An alternative model for displacement interpolation (§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 (15), which *directly* predicts the interpolation result at time t rather than computing the entire interpolation simultaneously.

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 (near-)indicator function of an averaged shape as the minimizer $\mu \in \mathrm{Prob}([-1,1]^2)$ of $\sum_i \alpha_i \mathcal{W}^2_{2,\mathbf{H}_t}(\mu, \chi_i)$; this indicator easily can be sharpened if a true binary function is desired.

Fig. 12 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. Fig. 13 provides a 1D interpolation example, with simple post-processing (thresholding and coloring) to recover boundaries. Figs. 1, 7, and 14 show analogous examples in three dimensions. We represent a surface volumetrically using the normalized indicator function of its interior. 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).

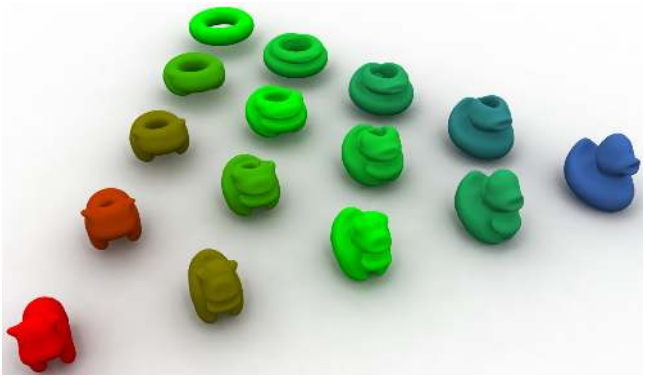


Figure 7: *Shape interpolation in 3D, expanded from Fig. 1.*

BRDF design. The BRDF $f(\omega_\mathbf{i}, \omega_\mathbf{o})$ of a material defines how much light it reflects from each incoming direction $\omega_\mathbf{i}$ to each outgoing direction $\omega_\mathbf{o}$. If $\omega_\mathbf{i}$ is fixed, all the outgoing directions fall on a hemisphere defined by the surface normal. After scaling, the BRDF values for $\omega_\mathbf{o}$ form a probability distribution over the hemisphere. Hence, displacement interpolation can be applied to interpolate between materials, as in [Bonneel et al. 2011].

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 $\mathbf{H}_t$ is approximated by the fast approximate Gaussian convolution from [Deriche 1993]. Spherical geometry is accounted for by modulating the width of this separable filter. We render images using the interpolated BRDFs using PBRT [Pharr and Humphreys 2010].

Fig. 8 shows interpolation between four BRDFs using our technique, yielding continuously-moving highlights. The corner BRDFs are sampled from closed-form materials [Blinn 1977; Ashikhmin and Shirley 2000]; the remaining BRDFs are interpolated.

Color histogram manipulation. In image processing, optimal transportation has proven useful for color palette manipulations like contrast adjustment [Delon 2006] and color transfer [Pitié et al. 2007] 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 [Pitié et al. 2007; Bonneel et al. 2014; Papadakis et al. 2011] or can support only coarse histograms [Ferradans et al. 2014]. 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 requires 1D transport, which is computable in closed form [Villani 2003]; 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 $\boldsymbol{\mu}_i$ on an uniform rectangular grid over M.

For a given set of weights $\alpha \in \mathbb{R}_+^k$, we solve the barycenter problem (12) using Algorithm 2. This provides the weighted barycenter $\mu \in \mathrm{Prob}(M)$, discretized as a vector $\boldsymbol{\mu}$. The algorithm further-

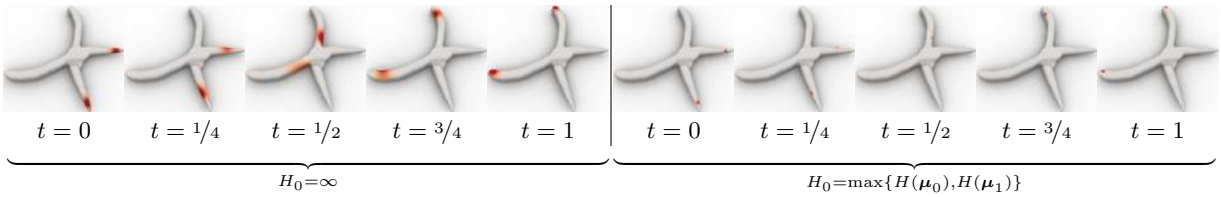


Figure 6: *Displacement interpolation without (left) and with (right) entropy limits. The optimization implicitly matches the two peaks at $t = 0$ and $t = 1$ and moves mass smoothly from one distribution to the other.*

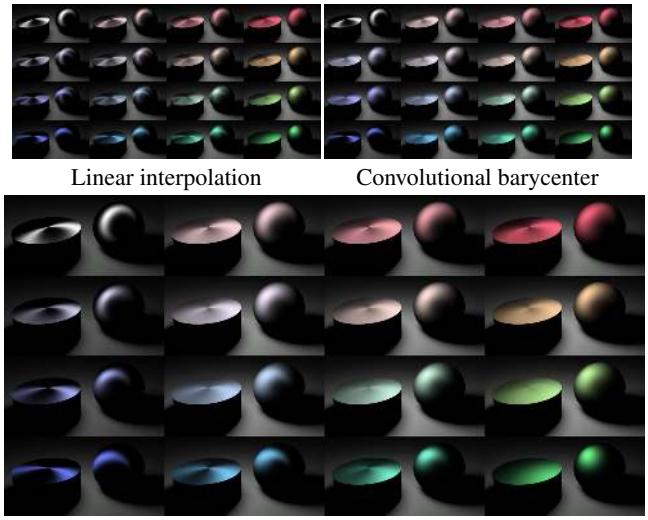


Figure 8: *BRDF interpolation: BRDFs for the materials in the four corners of each image are fixed, and the rest are computed using bilinear weights. Linearly interpolating BRDFs (left) yields spurious highlights, the convolutional barycenter (center) moves highlights continuously but increases diffusion, and the entropy-bounded barycenter (right) moves highlights in a sharper fashion.*

more provides the scaling factors $(\mathbf{v}_i, \mathbf{w}_i)$ for each $i = 1, \ldots, k$, which define the transport maps $\boldsymbol{\pi}_i = D_{\mathbf{v}_i} K D_{\mathbf{w}_i}$ between each input histogram $\boldsymbol{\mu}_i$ and the barycenter $\boldsymbol{\mu}$. This discrete coupling $\boldsymbol{\pi}_i$ 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

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

This integral is computed numerically as a sum over the grid, where $\boldsymbol{\pi}_i$ 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 Fig. 2. Thus, as $\gamma \to 0$, T_i converges to the optimal transport map. It can thus be used to define a corrected image $f_i^\alpha \overset{\text{def.}}{=} T_i \circ f_i$ whose chrominance histogram matches μ. Fig. 11 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

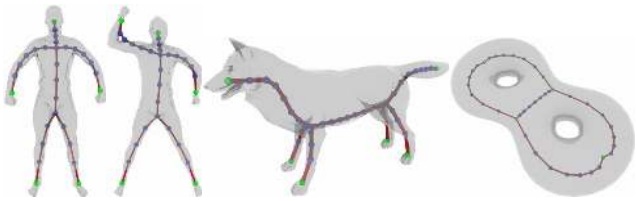


Figure 9: *Embeddings of skeletons computed using Wasserstein propagation; the positions of the blue vertices are computed automatically using the fixed green vertices and topology of the graph.*

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 (§6.3). We take as input the positions of vertices in a small subset $V_0 \subseteq V$. As suggested by Solomon et al. [2014a], we express the position of each $v \in V_0$ as a distribution $\boldsymbol{\mu}_v \in \mathrm{Prob}(M)$ using barycentric coordinates computed using the algorithm by Ju et al. [2005]. Distributions $\boldsymbol{\mu}_v \in \mathrm{Prob}(M)$ can be interpolated along G to the remaining $v \notin V_0$ via Wasserstein propagation with uniform edge weights. The computed $\boldsymbol{\mu}_v$'s serve as barycentric coordinates to embed the unlabeled vertices. Thanks to displacement interpolation, the constructed embedding conforms to the geometry of the surface; Fig. 9 shows sample embeddings generated using this 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 $\mu_x : M_0 \to \mathrm{Prob}(M)$. Solomon et al. [2013] generalize 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 $\mathbf{H}_t$ is the heat kernel matrix of M. A regularized discretization of the measure-valued map Dirichlet energy is provided by the Wasserstein propagation objective (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 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 $\mathbf{c}_v$ to sample the compatibility function $c(v, \cdot)$ on M associated with $v \in M_0$. We modify the objective (17) as follows:

$$\left[\sum_{(v,w) \in E} \frac{1}{\ell^2_{(v,w)}} \mathcal{W}^2_{2,\mathbf{H}_t}(\boldsymbol{\mu}_v, \boldsymbol{\mu}_w) \right] + \tau \left[\sum_{v \in V} \omega_v \mathbf{a}^\top (\boldsymbol{\mu}_v \otimes \mathbf{c}_v) \right]. \tag{18}$$

Source $e^{-\alpha c(x,y)}$ Result Result (left/right flip)

Figure 10: *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.*

This objective favors distributions $\boldsymbol{\mu}_v$ 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, (18) equals $\sum_{(v,w)\in E} \mathcal{W}^2_{2,\overline{\mathbf{H}}_t}(\boldsymbol{\mu}_v, \boldsymbol{\mu}_w)/\ell^2_{(v,w)}$, where

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

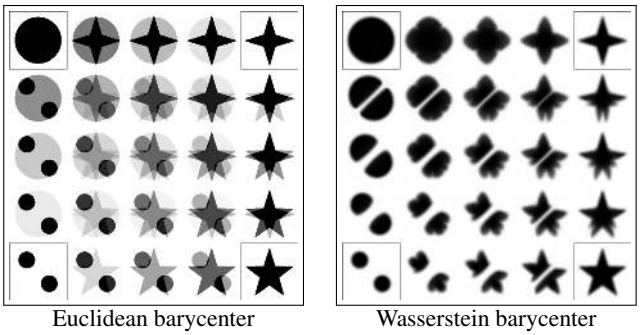
This matrix is a diagonal rescaling of $\mathbf{H}_t$, so we can still efficiently optimize (18) using Algorithm 4, slightly adjusted to use a different kernel on each edge. Fig. 10 shows maps between a pair of surfaces computed using this technique. Because the models are nearly isometric, we use the wave kernel signature (WKS) [Aubry et al. 2011] 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 [Solomon et al. 2012], computed in less than a minute using similar hardware.

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 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 [Zhu et al. 2014] 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 [Crane et al. 2013]; this is the primary drawback of our transport approximation. Modeling with "true" quadratic



Euclidean barycenter Wasserstein barycenter

Figure 12: *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.*

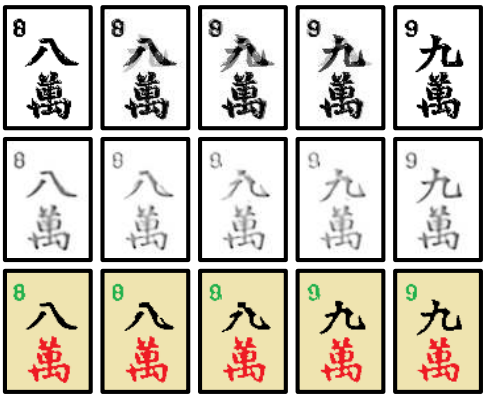


Figure 13: *"Generalized Mahjong:" Linear (top) and displacement (middle) interpolation between two images; while it is less sharp, the displacement interpolation result can be post-processed using simple image filters to generate a nontrivial interpolation (bottom; see e.g. the tip of the "9" character rotating outward).*

Wasserstein distances remains a challenge on images and triangle meshes, and large-scale discretizations of flow models proposed by Benamou and Brenier [2000] remain to be formulated. Closer to the current discussion, the algorithm for propagation in §6.3 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.

Acknowledgments

J. Solomon acknowledges the support of the Hertz Foundation Fellowship and the NSF GRFP. The work of G. Peyré has been supported by the European Research Council (ERC project SIGMA-Vision). M. Cuturi acknowledges the support of JSPS young researcher A grant 26700002. L. Guibas acknowledges the support of ONR MURI grant N00014-13-1-0341, NSF CCF grant 1161480, and a Google Research Award.

Cow model courtesy K. Crane. Cat and wolf models from TOSCA dataset. Human model from SCAPE dataset. Pliers and armadillo models from SHREC 2007 benchmark. Duck and sharp sphere models from AIM@Shape dataset, with owners IMATI AND MPII.

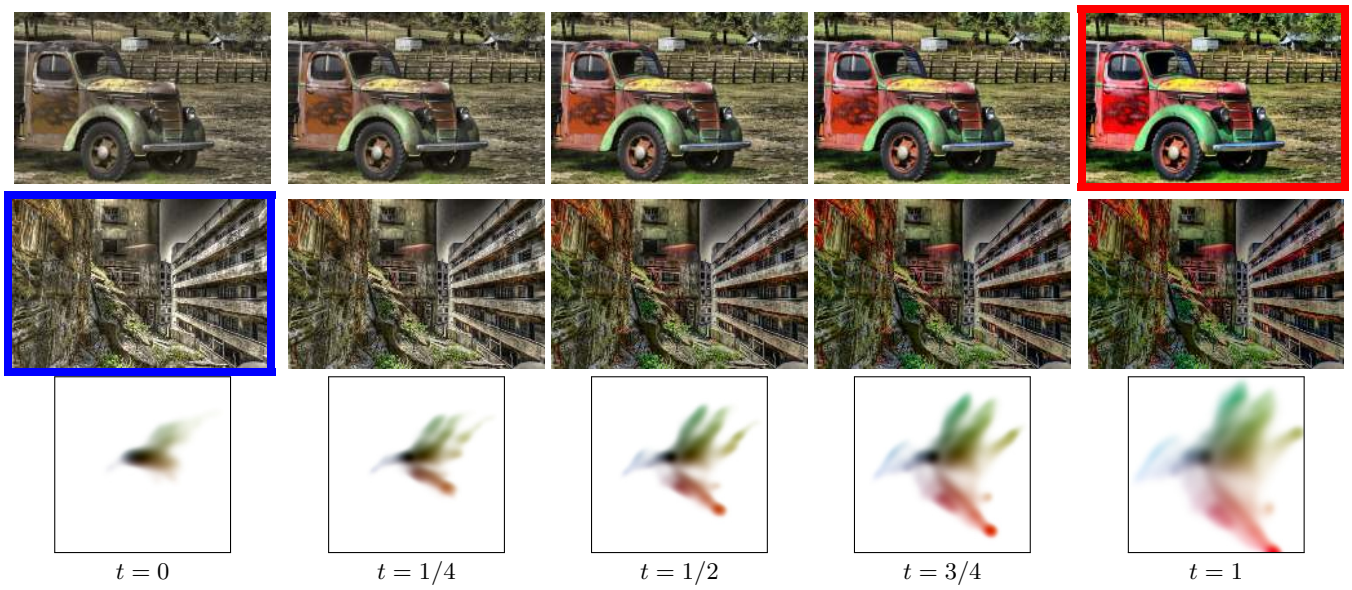


Figure 11: *Color transfer with 2D convolutional transportation over the chrominance space. Top row: evolution of the color-corrected image f_1^α as a function of $\alpha = (1-t, t)$. Middle row: evolution of f_2^α. The red (resp. blue) framed image shows the input f_1 (resp. f_2) which is obtained for $t = 0$ (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).*

References

AGUEH, M., AND CARLIER, G. 2011. Barycenters in the Wasserstein space. *SIAM J. Math. Anal. 43*, 2, 904–924.

ASHIKHMIN, M., AND SHIRLEY, P. 2000. An anisotropic Phong BRDF model. *J. of Graph. Tools 5*, 2, 25–32.

AUBRY, M., SCHLICKEWEI, U., AND CREMERS, D. 2011. The wave kernel signature: A quantum mechanical approach to shape analysis. In *Proc. ICCV Workshops*, 1626–1633.

BENAMOU, J.-D., AND BRENIER, Y. 2000. A computational fluid mechanics solution of the Monge-Kantorovich mass transfer problem. *Numerische Mathematik 84*, 3, 375–393.

BENAMOU, J.-D., CARLIER, G., CUTURI, M., NENNA, L., AND PEYRÉ, G. 2015. Iterative Bregman projections for regularized transportation problems. *SIAM J. Sci. Comp., to appear*.

BLINN, J. F. 1977. Models of light reflection for computer synthesized pictures. In *Proc. SIGGRAPH*, vol. 11, 192–198.

BONNEEL, N., VAN DE PANNE, M., PARIS, S., AND HEIDRICH, W. 2011. Displacement interpolation using Lagrangian mass transport. *ACM Trans. Graph. 30*, 6 (Dec.), 158:1–158:12.

BONNEEL, N., RABIN, J., PEYRÉ, G., AND PFISTER, H. 2014. Sliced and Radon Wasserstein barycenters of measures. *J. Math. Imaging and Vision, to appear*.

BREGMAN, L. 1967. 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*, 3, 200–217.

BURKARD, R., AND ÇELA, E. 1999. Linear assignment problems and extensions. *Handbook of Combinatorial Optimization: Supplement 1*, 75.

BURKARD, R., DELL'AMICO, M., AND MARTELLO, S. 2009. *Assignment Problems*. SIAM.

CARLIER, G., OBERMAN, A., AND OUDET, E. 2014. Numerical methods for matching for teams and Wasserstein barycenters. Preprint, Ceremade.

COVER, T., AND THOMAS, J. 2006. *Elements of Information Theory*. Wiley.

CRANE, K., WEISCHEDEL, C., AND WARDETZKY, M. 2013. Geodesics in heat: A new approach to computing distance based on heat flow. *ACM Trans. Graph. 32*, 5 (Oct.), 152:1–152:11.

CUTURI, M., AND DOUCET, A. 2014. Fast computation of Wasserstein barycenters. In *Proc. ICML*, vol. 32.

CUTURI, M. 2013. Sinkhorn distances: Lightspeed computation of optimal transportation. In *Proc. NIPS*, vol. 26. 2292–2300.

DAVIS, T. A. 2006. *Direct Methods for Sparse Linear Systems*. SIAM.

DE GOES, F., COHEN-STEINER, D., ALLIEZ, P., AND DESBRUN, M. 2011. An optimal transport approach to robust reconstruction and simplification of 2d shapes. In *Computer Graph. Forum*, vol. 30, 1593–1602.

DE GOES, F., BREEDEN, K., OSTROMOUKHOV, V., AND DESBRUN, M. 2012. Blue noise through optimal transport. *ACM Trans. Graph. 31*, 6 (Nov.), 171:1–171:11.

DE GOES, F., MEMARI, P., MULLEN, P., AND DESBRUN, M. 2014. Weighted triangulations for geometry processing. *ACM Trans. Graph. 33*, 3.

DELON, J. 2006. Movie and video scale-time equalization application to flicker reduction. *IEEE Trans. on Image Proc. 15*, 1, 241–248.

DEMING, W. E., AND STEPHAN, F. F. 1940. On a least squares adjustment of a sampled frequency table when the expected marginal totals are known. *Annals Math. Stat. 11*, 4, 427–444.

DERICHE, R. 1993. Recursively implementing the Gaussian and its derivatives. Tech. Rep. RR-1893.

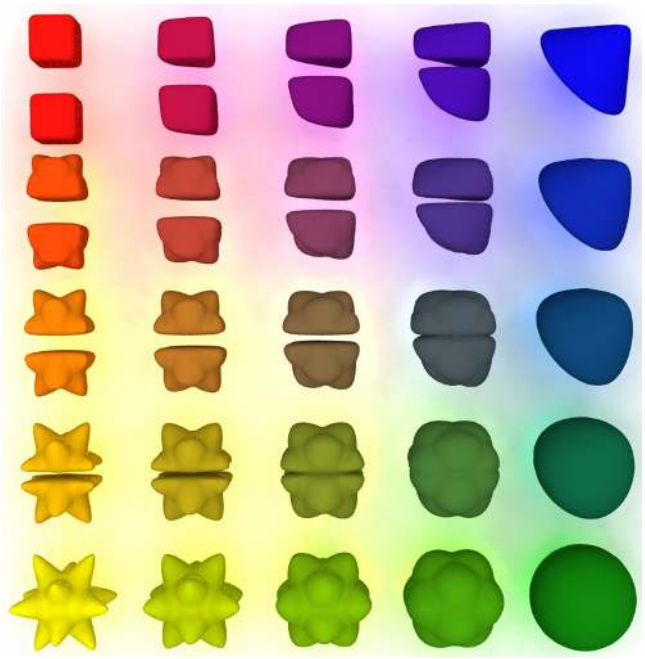


Figure 14: *Three-dimensional shape interpolation. The four corner shapes are represented using normalized indicator functions on a* $60 \times 60 \times 60$ *volumetric grid; barycenters of the distributions are computed using bilinear weights.*

DESBRUN, M., MEYER, M., SCHRÖDER, P., AND BARR, A. H. 1999. Implicit fairing of irregular meshes using diffusion and curvature flow. In *Proc. SIGGRAPH*, 317–324.

ESCALANTE, R., AND RAYDAN, M. 2011. *Alternating Projection Methods*. Fundamentals of Algorithms. SIAM.

FERRADANS, S., PAPADAKIS, N., PEYRÉ, G., AND AUJOL, J.-F. 2014. Regularized discrete optimal transport. *SIAM J. Imaging Sci. 7*, 3, 1853–1882.

FRANKLIN, J., AND LORENZ, J. 1989. On the scaling of multidimensional matrices. *Linear Algebra and its Applications 114*, 717–735.

JOHNSON, D. B. 1977. Efficient algorithms for shortest paths in sparse networks. *J. ACM 24*, 1 (Jan.), 1–13.

JU, T., SCHAEFER, S., AND WARREN, J. 2005. Mean value coordinates for closed triangular meshes. *ACM Trans. Graph. 24*, 3 (July), 561–566.

KANTOROVICH, L. 1942. On the transfer of masses (in Russian). *Doklady Akademii Nauk 37*, 2, 227–229.

KIM, Y.-H., AND PASS, B. 2013. Multi-marginal optimal transport on Riemannian manifolds. *arXiv:1303.6251*.

KNIGHT, P. 2008. The Sinkhorn–Knopp algorithm: Convergence and applications. *SIAM J. on Matrix Anal. and Applications 30*, 1, 261–275.

LÉONARD, C. 2012. From the Schrödinger problem to the Monge-Kantorovich problem. *J. Funct. Anal. 262*, 4, 1879–1920.

LIPMAN, Y., AND DAUBECHIES, I. 2011. Conformal Wasserstein distances: Comparing surfaces in polynomial time. *Adv. Math. 227*, 3, 1047–1077.

MACNEAL, R. 1949. *The Solution of Partial Differential Equations by means of Electrical Networks*. PhD thesis, Caltech.

MALLIAVIN, P., AND STROOCK, D. W. 1996. Short time behavior of the heat kernel and its logarithmic derivatives. *J. Differential Geom. 44*, 3, 550–570.

MCCANN, R. J. 1997. A convexity principle for interacting gases. *Adv. Math. 128*, 1, 153–179.

MÉRIGOT, Q. 2011. A multiscale approach to optimal transport. *Comp. Graph. Forum 30*, 5, 1583–1592.

MOSEK APS, 2014. Mosek version 7. `https://mosek.com`.

PAPADAKIS, N., PROVENZI, E., AND CASELLES, V. 2011. A variational model for histogram transfer of color images. *IEEE Trans. Image Proc. 20*, 6, 1682–1695.

PHARR, M., AND HUMPHREYS, G. 2010. *Physically Based Rendering, Second Edition: From Theory To Implementation*. Morgan Kaufmann, July.

PITIÉ, F., KOKARAM, A. C., AND DAHYOT, R. 2007. Automated colour grading using colour distribution transfer. *Comp. Vision and Image Understanding 107* (July), 123–137.

RUBNER, Y., TOMASI, C., AND GUIBAS, L. J. 2000. The earth mover's distance as a metric for image retrieval. *Int. J. Comput. Vision 40*, 2 (Nov.), 99–121.

SCHWARTZBURG, Y., TESTUZ, R., TAGLIASACCHI, A., AND PAULY, M. 2014. High-contrast computational caustic design. *ACM Trans. Graph. 33*, 4 (July), 74:1–74:11.

SINKHORN, R. 1964. A relationship between arbitrary positive matrices and doubly stochastic matrices. *Annals of Math. Stat. 35*, 2, 876–879.

SINKHORN, R. 1967. Diagonal equivalence to matrices with prescribed row and column sums. *American Math. Monthly 74*, 4, 402–405.

SOLOMON, J., NGUYEN, A., BUTSCHER, A., BEN-CHEN, M., AND GUIBAS, L. 2012. Soft maps between surfaces. *Comp. Graph. Forum 31*, 5 (Aug.), 1617–1626.

SOLOMON, J., GUIBAS, L., AND BUTSCHER, A. 2013. Dirichlet energy for analysis and synthesis of soft maps. *Comp. Graph. Forum 32*, 5, 197–206.

SOLOMON, J., RUSTAMOV, R., GUIBAS, L., AND BUTSCHER, A. 2014. Earth mover's distances on discrete surfaces. *ACM Trans. Graph. 33*, 4 (July), 67:1–67:12.

SOLOMON, J., RUSTAMOV, R., LEONIDAS, G., AND BUTSCHER, A. 2014. Wasserstein propagation for semi-supervised learning. In *Proc. ICML*, 306–314.

VARADHAN, S. R. S. 1967. On the behavior of the fundamental solution of the heat equation with variable coefficients. *Comm. on Pure and Applied Math. 20*, 2, 431–455.

VILLANI, C. 2003. *Topics in Optimal Transportation*. Graduate Studies in Mathematics. AMS.

ZHAO, X., SU, Z., GU, X. D., KAUFMAN, A., SUN, J., GAO, J., AND LUO, F. 2013. Area-preservation mapping using optimal mass transport. *IEEE Trans. Vis. and Comp. Graphics 19*, 12.

ZHU, J.-Y., LEE, Y. J., AND EFROS, A. A. 2014. AverageExplorer: Interactive exploration and alignment of visual data collections. *ACM Trans. Graph. 33*, 4 (July), 160:1–160:11.