Displacement Interpolation Using Lagrangian Mass Transport

Nicolas Bonneel[1,3]   Michiel van de Panne[1]   Sylvain Paris[2]   Wolfgang Heidrich[1]
[1] University of British Columbia   [2] Adobe Systems, Inc.   [3] ALICE/INRIA Nancy

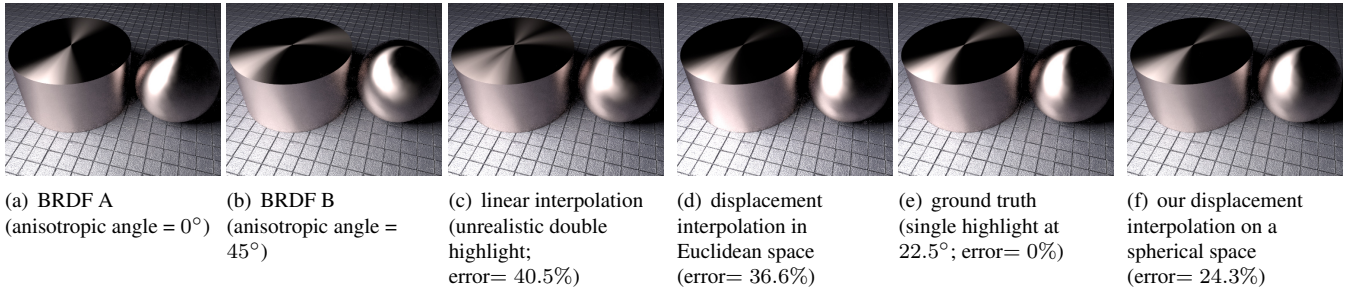| (a) BRDF A (anisotropic angle = $0°$) | (b) BRDF B (anisotropic angle = $45°$) | (c) linear interpolation (unrealistic double highlight; error= 40.5%) | (d) displacement interpolation in Euclidean space (error= 36.6%) | (e) ground truth (single highlight at $22.5°$; error= 0%) | (f) our displacement interpolation on a spherical space (error= 24.3%) |

Figure 1: *Examples of interpolation between two anisotropic sampled BRDFs (a,b). Naive linear interpolation generates an unrealistic double highlight (c). Working with a flattened BRDF representation in a Euclidean space produces large conspicuous errors (d). Since we used a parametric BRDF model, we can compute the ground-truth in-between BRDF by interpolating the parameters (e). Our approach has no knowledge of the parametric BRDF representation but is nonetheless able to produce a similar output (f).*

Abstract

Interpolation between pairs of values, typically vectors, is a fundamental operation in many computer graphics applications. In some cases simple linear interpolation yields meaningful results without requiring domain knowledge. However, interpolation between pairs of distributions or pairs of functions often demands more care because features may exhibit translational motion between exemplars. This property is not captured by linear interpolation. This paper develops the use of *displacement interpolation* for this class of problem, which provides a generic method for interpolating between distributions or functions based on advection instead of blending. The functions can be non-uniformly sampled, high-dimensional, and defined on non-Euclidean manifolds, e.g., spheres and tori. Our method decomposes distributions or functions into sums of radial basis functions (RBFs). We solve a mass transport problem to pair the RBFs and apply partial transport to obtain the interpolated function. We describe practical methods for computing the RBF decomposition and solving the transport problem. We demonstrate the interpolation approach on synthetic examples, BRDFs, color distributions, environment maps, stipple patterns, and value functions.

CR Categories: I.3.7 [Computing Methodologies]: Computer Graphics—Three-Dimensional Graphics and Realism, G.1.1 [Mathematics of Computing]: Numerical Analysis—Interpolation

Keywords: displacement interpolation, mass transport

Links: ⬢DL ⬛PDF ⬛WEB ⬤VIDEO

1 Introduction

Interpolation is a fundamental operation in many mathematical models and algorithms. Linear interpolation between a pair of values is particularly simple. However, linear interpolation does not always achieve meaningful results. In particular, this is often the case for probability distributions. Consider two Gaussian distributions, $\mathcal{N}_A(\mu_A, \sigma_A)$ and $\mathcal{N}_B(\mu_B, \sigma_B)$, that model height in two different populations of humans, group A with mostly short people and group B with mostly tall people. There are two plausible answers as to what an intermediate interpolated distribution might look like, as shown in Figure 2. A first answer is to assume an equal mix of people from both groups, which therefore results in the bimodal height distribution shown in Fig.2 (b). This *mixture model* interpretation is the answer that arises from linear interpolation of the two initial distributions at every point in their domain. An alternative answer can be obtained by interpolating the underlying parameters of the distributions, yielding the distribution $\mathcal{N}_C(\mu_C, \sigma_C)$, where $\mu_C = 0.5\mu_A + 0.5\mu_B$ and $\sigma_C^2 = 0.5\sigma_A^2 + 0.5\sigma_B^2$. Intuitively, this interpretation says that interpolating between the height distribution of a group of tall people and that of a group of short people should result in a distribution of medium-height people.

The need for something other than the mixture model interpretation is in fact rather common. For example, the mean of two bidirectional reflectance distribution functions (BRDFs), which each have a single large reflectance lobe but each pointing in a different direction, is, subjectively speaking, better defined as having a single large reflectance lobe in a halfway direction, rather than two half-sized lobes in the original directions. The difference between these two interpretations is shown in Figure 1 for a BRDF example where we also know the underlying parameterization.

The *displacement interpolation* methods developed in this paper provide an efficient way to achieve the type of *advection-based* or *motion-based* interpolation that characterizes the above examples and a wide range of applications. Importantly, it does this in a *generic way* without requiring access to a domain-specific underlying parameterization. Displacement interpolation casts interpolation as a mass transport problem, wherein each unit of 'mass' of one distribution needs to move to the second distribution. The goal is to find a minimal total cost solution, where the cost for each unit

of mass is a function of the distance traveled, known as the *ground distance*. The paths taken by each unit of mass can be parameterized over a time interval $t \in [0, 1]$ and an interpolated distribution is then created by stopping the mass transport at a desired intermediate value of t.

Challenges: A number of difficulties need to be overcome in order to develop practical displacement interpolation methods. First, the most common mass transport formulations work with binned histograms, which allows the problem to be cast in terms of bipartite graph matching or min cost flow on a graph [Hillier and Lieberman 1990]. Our interest is in developing methods that can be applied to continuous domains: we generate continuous interpolated distributions from a pair of continuous input distributions. While there exists limited work on Eulerian approaches for displacement interpolation in continuous domains solving PDEs, these methods require dense grids that rapidly become impractical in higher dimensions, and are limited to quadratic ground distances in a Euclidean space [Rehman et al. 2009; Rehman et al. 2007]. These methods do not work on non uniformly spaced scattered data typically obtained from measurements. Developing a Lagrangian method requires to start again from the first mass transport principles to build a PDE-free smoothed particles method and linking it to analytic 1D CDF-based interpolation. Second, commonly used discrete mass transport solvers are impractically slow. We study various implementation choices making the method much more efficient. Third, the advection-style interpolation obtained by displacement interpolation easily breaks by the addition of a small constant offset to each function. We solve this issue by proposing a multiscale approach.

Method overview: Our approach proceeds in several steps: we first decompose the functions into basis elements, in practice a multiscale representation in terms of non-negative Gaussian RBFs. We then solve the mass transport problem to find a pairing between these elements. The interpolation step consists of applying a partial transport by advecting the elements only part of the way from their source locations to their target locations. Finally we reconstruct the interpolated function by summing the advected RBFs. The bulk of the computation lies in the function decomposition, which is posed as a non-negative least squares problem, and in solving the transport problem, which is posed using the Monge-Kantorovich formulation. With these steps precomputed, the different interpolated distributions can be generated efficiently.

This paper makes the following contributions:

- We develop a novel Lagrangian method for displacement interpolation of continuous functions and distributions. The method uses non-negative radial-basis function approximations and kernel advection.

- We extend the displacement interpolation framework so that it can be applied to more general functions using a multi-scale decomposition. We interpolate different frequency bands independently so that features of different frequencies are properly advected.
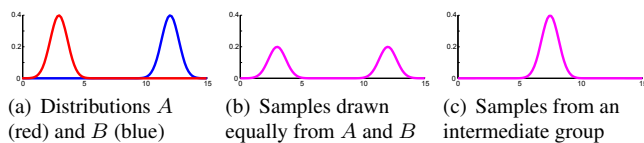

(a) Distributions A (red) and B (blue)

(b) Samples drawn equally from A and B

(c) Samples from an intermediate group

Figure 2: *Interpolating distributions A and B: (a) original distributions; (b) linear interpolation; (c) displacement interpolation.*

- We evaluate several discrete mass transport solvers for the discrete problem that underlies our continuous transport method. We show that a fixed-point network flow implementation is significantly faster than currently available libraries dedicated to solving earth-mover's distance problems.

- We demonstrate the application of displacement interpolation to a range of problems including synthetic data, BRDFs, color distributions, environment maps, stipple patterns, and value functions.

- We show that CDF-based methods for interpolating 1D distributions are a special case of displacement interpolation.

2 Related Work

We first briefly review a number of existing interpolation techniques used in the context of computer graphics. We then provide background on the problem of mass transport and the Monge-Kantorovich formulation that we use to solve it.

2.1 Function interpolation in computer graphics

Interpolation of 1D probability distribution functions (PDFs) is often achieved by means of the inverse cumulative distribution functions (CDFs) [Read 1999; Matusik et al. 2005]. This approach exhibits the behavior that we seek because it tends to advect features instead of cross-fading them. In this paper (§3.5), we show that this is in fact a special case of displacement interpolation. Unfortunately, the CDF approach has no direct equivalence in higher dimensions, and therefore cannot be applied to the more general settings we are interested in. Pitié et al. [2005] describe how to perform histogram transfer between multi-dimensional PDFs but it is unclear if there would be a way to adapt their method to achieve interpolated distributions. Bursal [1996] proposes a PDF interpolation method for arbitrary dimension, but it is fundamentally restricted by its use of a single affine transform to align two distributions. In the animation domain, da Silva et al. [2009] exploit the structure of a specific class of problem, to derive a non-linear transform for the interpolation of value functions.

Mass transport methods have been proposed to establish distance metrics between images [Rubner et al. 2000] and geometries [Lipman and Daubechies]. For example, Rubner et al. [2000] use the earth movers distance (EMD) between color histograms as a metric for image retrieval [Rubner et al. 2000]. For the task of 2D shapes interpolation, clustered mass transport has also been proposed [Makihara and Yagi 2011], reconstructing binary silhouettes from the advected fuzzy clustering. In computer vision, mass transport has been used for visual tracking. In particular, Zhao et al. [2010] specifically tailored a fast EMD computation for visual tracking, but results have only been demonstrated on 16 clusters while we handle thousands.

In computer graphics, filtering of values requires some form of interpolation. For the task of normal map filtering, it has been proposed to fit Gaussians to normal distributions using Expectation-Maximization (E-M), and then interpolate the parameters of the Gaussians [Tan et al. 2005; Han et al. 2007]. The E-M algorithm allows for recovery of the underlying mixture-of-Gaussians probability model giving rise to the distribution of normals. This density estimation problem is very different from our fitting of Gaussians which finds the sum of Gaussians which best fit the data. In addition, their use of the E-M algorithm to match Gaussians together only provides a one-to-one mapping while we allow for one-to-many.
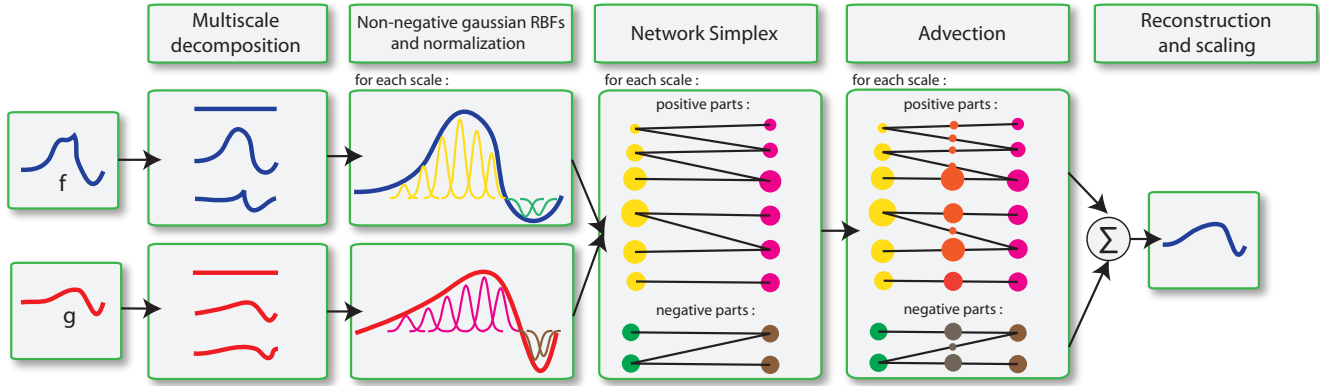
Figure 3: *Overview of our pipeline.*

We propose a method that applies to a variety of interpolation problems with only minor adjustments and that comes with a number of guaranteed properties. We believe that a generic method for the interpolation of multi-dimensional functions is a useful addition to the set of mathematical tools available in the computer graphics toolbox.

2.2 Background on displacement interpolation

Displacement interpolation has first been introduced by Mc-Cann [1997] in the context of gases in Euclidean spaces. It consists of the partial advection of a source measure (or a continuous function) to a target measure in a mass preserving manner. This interpolation is based on a mass transport problem known as *Monge-Kantorovich* mass transport. In the remainder of this section, we present the principal aspects of the mass transport problem, which are relevant to our work. We refer the reader to related books for a more extensive review, e.g. [Villani 2003; Villani 2008].

Mass transport The Monge-Kantorovich problem reshapes a heap of matter f to match a target function g, while minimizing the *cost function*, or *ground distance* $c(x,y)$, which describes the expense of moving a particle from x to y. Kantorovich proposed a linear formulation of the problem. More details on a measure theory view of displacement interpolation are provided in a short document included as supplemental material. Intuitively, Kantorovich treats the mass moved between all pairs of points x and y as an unknown, and seeks to minimize the transport cost over all possible movement patterns. In our work, we assume that f and g come from continuous density functions which have been discretely sampled. In this context and under reasonable assumptions the mass transport problem is known to have a unique solution [Ahmad et al. 2010]. This result holds for Euclidean and Riemannian manifolds, e.g. $\mathbb{R}^n$ and spheres, using a geodesic distance with a power $p > 1$ as a cost function. This covers all the applications demonstrated in this paper.

In the special case where the cost function is quadratic and the space is Euclidean, the Kantorovich formulation can be solved with linear partial differential equations [Benamou and Brenier 2000; Haker et al. 2004; Rehman et al. 2009; Zhu et al. 2007]. However, this Eulerian approach has been demonstrated only on Euclidean spaces and requires storing a dense grid, which rapidly becomes impractical for multi-dimensional functions. This approach has particularly been exploited for image morphing [Haker et al. 2004; Zhu et al. 2007] where cartesian grids naturally arise. In the case of sparse scattered data, a grid needs to cover the whole space, including the

empty regions between samples, thereby quickly becoming impractical for large spaces.

In this paper, we take a *Lagrangian approach* as advocated by Villani [2003, § 5.4]. We follow a discrete set of moving particles and use the Hitchcock-Koopman formulation [Flood 1953] that allows for more general ground distances, dimension, topology and non-uniform sampling.

Given a set of supplies f_i, a set of demands g_j, and a shipping cost $c_{i,j}$ to move one unit from f_i to g_j, we minimize the following energy:

$$\min_x \sum_i \sum_j c_{i,j}\, x_{i \to j} \tag{1a}$$

$$\text{such that:} \quad x_{i \to j} \geq 0 \tag{1b}$$

$$\sum_i x_{i \to j} = g_j \tag{1c}$$

$$\sum_j x_{i \to j} = f_i \tag{1d}$$

over all $x_{i \to j}$, which represent the (unknown) amount of goods to be moved from f_i to g_j.

The above formulation is generally used to compute the *Earth Mover's distance* (EMD). This formulation can be represented by a weighted bipartite graph where each source node i provides a mass f_i, and each target node j requests a mass g_j. The flow of mass is modeled using a set of arcs from each source i to one or more targets j. This is illustrated in Figure 4. Minimizing Equation 1 is a special case of the *min cost flow* problem. The number of arcs describing the optimal flow is known to be at most $n_f + n_g - 1$ where n_f and n_g are the numbers of source and target nodes [Flood 1953]. The number of these arcs matters because our interpolated result maintains a particle for each arc. We will discuss possible algorithms to minimize Equation 1 in Section 4. While fast method have been proposed for particular cases, such as thresholded ground distances [Pele and Werman 2009], they do not apply in our context. In particular, thresholded ground distances would yield an arbitrary assignment for distant particles.

Displacement interpolation along geodesic curves. Our approach works on Riemannian manifolds such as $\mathbb{R}^n$, a sphere, or a torus. The cost c controls how we match the source function to the target function (Eq. 1a). In this paper, we follow Villani [2008] and define c in terms of a particle moving from x to y along a path $\gamma(t)$ such that $\gamma(0) = x$ and $\gamma(1) = y$. We characterize γ with a Lagrangian function, i.e., a function whose integral is minimized along the path. Using the speed of the particle raised to a power
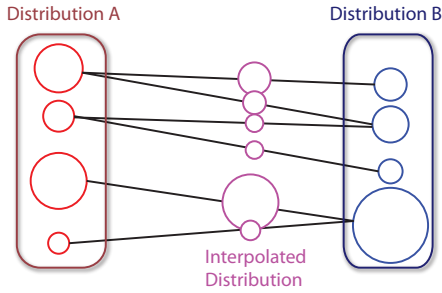
Figure 4: *The EMD sees the distribution as bins that need to be matched as in a graph matching problem, independently of the underlying dimension. We find an interpolant by advecting the mass of each bin, possibly split into many. Using a geodesic ground distance at the power p, the resulting distribution interpolates the EMD at the power $1/p$.*

$p > 1$, i.e., $|\partial\gamma/\partial t|^p$, as the Lagrangian leads to $c(x, y) = d(x, y)^p$ where $d(x, y)$ is the geodesic distance between x and y on the manifold. Solving Equation 1 pairs the source and target particles and for each such pair it gives us a geodesic path γ. Moving the particles along these paths is the core of displacement interpolation. Further, it is known that for $p = 1$, paths do not cross. With $p > 1$, at any time t, particles are never collocated. These properties are useful in practice – for example, they ensure that stipples do not collide when interpolating between two stipple patterns.

3 Displacement Interpolation of Continuous Distributions

Our interpolation follows a three-step process. First, the source and target distributions are decomposed into a sum of Gaussians. Each Gaussian in the source distribution is then paired to one or more Gaussians of the target distributions by solving a mass transport problem. In the last step, an interpolated distribution is constructed by summing the Gaussians after a partial advection to the target locations. When appropriate, this whole procedure is repeated independently at different scales to achieve a multiscale decomposition and reconstruction. Figure 3 summarizes the complete pipeline.

3.1 RBF decomposition

We assume that the source and target functions f and g are given as a set of samples at locations x_i and y_j, that is, we know a set of values $f(x_i)$ and $g(y_j)$. We do not require that the x_i and y_j points be the same or that they be regularly spaced. These samples could be used directly as particles for mass transport, but this yields highly discontinuous interpolated results that are not desirable for most applications. We address this issue by associating a smooth kernel with each sample point. We choose Gaussian radial basis functions (RBFs) as our kernel. They will also serve as the particles for the formulation of the mass transport problem. Each particle is represented by $w\,G_\sigma$, where w is the mass associated with the particle, and G_σ is a normalized Gaussian kernel with variance σ^2 and centered at the sample point. This representation is different from Gaussian Mixture Models used for density estimation with Expectation-Maximization techniques [Tan et al. 2005]: while Expectation-Maximization gives higher values where the *density* of samples is higher, our RBF decomposition *interpolates* values stored at arbitrarily scattered sample points.

In order to represent a non-negative function f as an appropriate sum-of-Gaussians, we seek to approximate $f(x)$ by

$\sum_i w_i\,G_{\sigma_i}(d(x_i, x))$, where $d(x_i, x)$ is the geodesic distance between point x and the center of a particle x_i, and w_i and σ_i are the unknowns to be determined. The kernel width σ_i controls the smoothness of particle representations. Kernels that are too narrow will yield poor interpolation of f in between sample points, while kernels that are too wide will fail to capture fine-scale details. In practice, we use $\sigma_i = d_i^N$, where d_i^N is the distance between x_i and its Nth nearest neighbor. This produces good results for $1 \le N \le 10$ depending on the application (§5). Given σ_i, we can then estimate the weights w_i using a non-negative least-squares formulation:

$$\min_{w_i} \sum_k \left[f(x_k) - \sum_i w_i\,G_{\sigma_i}(d(x_i, x_k)) \right]^2 \quad \text{with } w_i \ge 0 \quad (2)$$

The non-negativity constraint is important when dealing with distributions or physically constrained properties such as BRDFs. In our experiments, the residual of Equation 2 is usually on the order of 1% of the L_2 norm of f.

The above method assumes a non-negative function. For a function with positive and negative values, we decompose it into its positive and negative components, $f^+ = \max(0, f)$ and $f^- = \max(0, -f)$ (the same applies to g), interpolate each component separately, and recombine the result $h = h^+ - h^-$.

Matrices used in RBF fitting are often ill-conditioned. We therefore use a dense non-negative least-squares solver based on QR factorization [Lawson and Hanson 1995], which is both more robust and faster than solvers based on sparse matrices and the normal equations [Cantarella and Piatek 2004].

3.2 Mass transport

The core of the algorithm resides in the mass transport scheme [Flood 1953]. The goal is to pair each particle of the source distribution f to one or more Gaussian particles of the target distribution g. First, we ensure that the source and target functions have the same unit mass by normalizing their weights w_i and w_j by $s_f = \sum_i w_i$ and $s_g = \sum_j w_j$ respectively, yielding normalized weights, $\bar{w}$. Since the Gaussian kernels G are normalized, the mass of a particle $\bar{w}\,G$ is simply $\bar{w}$. To compute the transportation cost, we multiply the mass $\bar{w}$ of a particle by the cost function, which is based on the distance traveled by its center, allowing us to apply Equation 1 without modification. This assumes that the mass of a particle is concentrated at its center, which further motivates our use of small kernels (§ 3.1).

The transport problem, i.e., the minimization posed by Equation 1, is solved using a network simplex method. This choice is motivated in detail in Section 4. The output of this algorithm is a pairing between the source and target particles. A particle may be associated with several particles in the other distribution. In this case, it is split into as many particles as pairs in which it is involved, each with an associated weight that meets the mass transported requirement along the links and summing up to the mass of the original particle. As a result, we have new source and target particles, denoted by $\hat{\imath}$ and $\hat{\jmath}$. The notation $\hat{\jmath}(\hat{\imath})$ is used for the index of the target particle paired with the source particle $\hat{\imath}$. The new set of particles is such that there are equal numbers of source and target particles, and that the paired particles have the same weight, i.e. $\bar{w}_{\hat{\imath}} = \bar{w}_{\hat{\jmath}(\hat{\imath})}$.

3.3 Particle advection

To build the interpolated function h corresponding to the parameter $t \in [0, 1]$, we advect the particles to the position $\gamma(t)$ along the geodesic path γ that links their source location to their target destination. We use z_ℓ to denote the new position of each particle. We

linearly interpolate the size of each particle: $\sigma_\ell^2 = (1-t)\sigma_i^2 + t\,\sigma_{j(i)}^2$ and the total mass of the function $s_h = (1-t)s_f + t\,s_g$. Since the weight is constant in a pair, it remains the same, i.e., $\bar{w}_\ell = \bar{w}_i = \bar{w}_{j(i)}$. With these values, we construct the interpolated function h:

$$h(x) = s_h \sum_\ell \bar{w}_\ell \, G_{\sigma_\ell}(d(z_\ell, x)) \tag{3}$$

3.4 Multiresolution interpolation

The Lagrangian mass transport method described thus far results in intuitive advection-like behavior in many settings. For example, interpolating between two bumps generates a translating bump. However, the intuitive behavior is lost when a constant offset or low frequency is added. In the example shown in Figure 5, the displacement interpolation solution (top two rows) does not establish correspondences between the peaks of the two distributions, nor for the valleys. Instead, the solution resembles a linear blending solution because of particles representing the peaks and valleys finding correspondences with particles that represent the constant offset associated with the function.

To remedy this problem, we propose a multi-resolution scheme that interpolates different band-passed versions of the functions separately. The lowest frequency bands are constant functions that are blended linearly. The other frequency bands are interpolated using the displacement interpolation method as describes previously. We typically use 3 frequency bands. The bottom two rows of Figure 5 illustrate the resulting more intuitive interpolation. We further discuss this point when describing the individual application (§ 5).
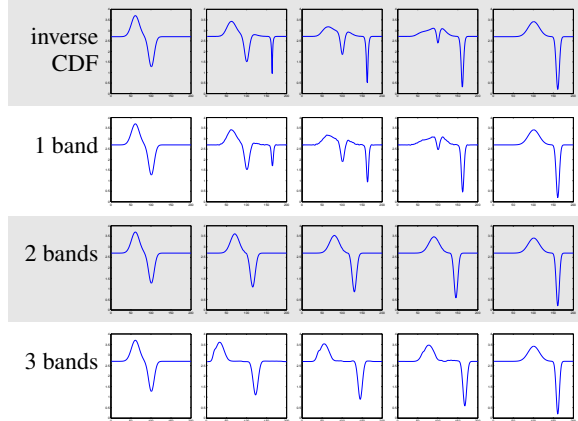

Figure 5: *Left to right: 1D interpolation of a constant plus a difference of two gaussians. Top to bottom: inverse CDF formula (failure case, due to the constant); displacement interpolation with 1 band (same as inverse CDF); 2 bands; 3 bands. More bands can yield undesirable artifacts in this case.*

3.5 Relationship to CDF-based interpolation

As noted in prior art, e.g., [Read 1999; Matusik et al. 2005], interpolation between 1D PDFs can be achieved by linearly interpolating the inverse of the cumulative density function (CDF) of the two 1D distributions:

$$F_\alpha^{-1}(x) = (1-\alpha)F_0^{-1}(x) + \alpha F_1^{-1}(x),$$

where $F(x)$ is the CDF of the distribution $f(x)$. To relate this to mass transport, it can be shown that for continuous 1D functions, the transformation

$$T_1 = F_1^{-1} \circ F_0 \tag{4}$$

gives an optimal change of variable from f_0 to f_1, for any cost function of the form $c(x, y) = k(x - y)$, where $k(x)$ is a symmetric convex function [Villani 2003]. In addition, for quadratic costs, one can express the displacement interpolation as a linear interpolation of this change of variable and the identity: $T_\alpha = (1-\alpha)Id + \alpha T_1$, with $\alpha \in [0, 1]$ [Villani 2003]. By composing with F_0^{-1}, we can see that the displacement interpolation on the real line is obtained as $T_\alpha \circ F_0^{-1} = (1-\alpha)F_0^{-1} + \alpha F_1^{-1}$. This operation is equivalent to the inverse CDF interpolation method since it transforms the function f_0 to f_α using Eq. 4. In addition, the displacement interpolation interpolates in the sense of the quadratic Wasserstein distance, defined as the square root of the EMD with a quadratic cost [Villani 2003, § 2.2]. For more general convex costs, one can easily compute the EMD on the real line by using the relationship $\mathrm{EMD}(f_0, f_1) = \int_0^1 c(F_0^{-1}(x), F_1^{-1}(x))dx$, [Villani 2003, § 2.2].

3.6 Linearly interpolated integral

Another property of interest is that the interpolant preserves the integral of the original functions. Since the s_h coefficients are linearly interpolated from the s_f and s_g coefficients (§ 3.2), we have:

$$\int h = (1-t)\int f + t\int g \tag{5}$$

That is, the integral of our interpolated result is a linear combination of the source and target integrals. This property is useful for instance when dealing with BRDFs because it guarantees that the interpolated BRDF reflects an amount of energy that varies linearly between that of the source and target BRDFs.

4 Mass Transport Solver

Several algorithms exist for solving the mass transport problem on graphs. We analyze two main categories. First, we investigate methods based on the *transportation simplex* that are dedicated to the computation of Earth Mover's Distance [Hillier and Lieberman 1990]. We test Rubner's code [1998] and a faster, more recent implementation [MacDonald 2006]. These are commonly used to solve computer vision problems with EMD, e.g. [Rubner et al. 2000]. We keep only the latter for our analysis because the original code by Rubner is limited to 350 samples, which is sufficient for tasks based on image histograms but is insufficient for applications that require a larger number of samples, i.e., upwards of several thousand. MacDonald's code [2006] uses floating-point precision. We also experiment with the generic graph library LEMON [2010] that provides several algorithms to solve the general min cost flow problems. In our context, preliminary testing showed that the *network simplex* algorithm with a *block search pivoting* strategy [Kelly and O'Neill 1991] performs best with a clear margin among the options offered in LEMON. We thus use this in the subsequent analysis. This code supports floating-point and fixed precisions and we test both. We also modify the code to exploit the fact that the graphs for solving the mass transport problem are fully connected, which allows us to avoid storing the graph connectivity explicitly. This enables us to process up to 32,000 particles compared to 15,000 for the same memory usage with the original implementation.

The transportation simplex has a worst-case exponential complexity but it has been observed to have a polynomial average case complexity under various distributions. The network simplex has a known complexity in $\mathcal{O}(n^3)$ [Ahuja et al. 1993]. To determine how these algorithms behave in practice in our context, we generate 20 random mass transport problems to evaluate the two classes of algorithm. Figure 6 plots the resulting performance. The tests reproduce the $\mathcal{O}(n^3)$ complexity of the transportation simplex but

also reveal that the network simplex behaves in $\mathcal{O}(n^2)$ in our context, which is a major gain at the scale at which we typical work, i.e. thousands of particles. This finding is also useful for applications that use EMD, where using the network simplex instead of the transport simplex can bring a significant performance increase. Our experiments also show that fixed-point precision further speeds up the computation. We observed that the value of the final transport cost is less accurate because of the limited precision, but that the particle pairing that produces the actual interpolation scheme remains unchanged. We used the fixed point method to generate the results presented in this paper. The results of the performance study are also of broader interest, as current EMD image retrieval or color transfer techniques rely on slower solvers [Rubner et al. 2000; Kanters et al. 2003; Morovic and Sun 2003].
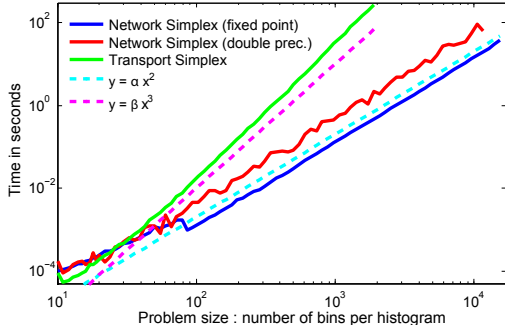
Figure 6: *Log-log plot of the running times of different solvers. The network simplex behaves as a $\mathcal{O}(n^2)$ algorithm in practice whereas the transport simplex runs in $\mathcal{O}(n^3)$.*

5 Results

In this section, we discuss specific applications and their associated details such as the choice of ground distance. We first present applications that handle continuous data, BRDFs and value functions of animation controllers. We then apply our method to discrete problems such as stipple rendering. Further results are also shown in the video that accompanies this paper.

5.1 Synthetic Data

Synthetic 1D examples are shown in Figure 5 as well as in the video that accompanies the paper. The synthetic 2D datasets shown in Figure 7 illustrate the general intuitive nature of the results obtained via Lagrangian-based displacement interpolation. In particular, they demonstrate interpolation between anisotropic distributions, isotropic distributions, distributions that require a split, and sharp-edged distributions that change shape. These examples are constructed using a grid of 140×140 samples, using a kernel width set according to the 10th nearest neighbor, except for the shape example which uses the first nearest neighbor. We use the 1-band interpolation solution.

5.2 BRDF interpolation

We demonstrate our method for interpolating BRDFs. Since the BRDF model does not include fluorescence, we can treat wavelengths independently, as there is no energy transfer across wavelengths. We use cosine-weighted BRDFs to ensure proper energy conservation, and work in the log domain. Logarithmic values give more importance to low intensities, which yields perceptually more meaningful results [Rusinkiewicz 1998]. In practice, we apply $\log(1 + x)$ to remap the values so that the function remains
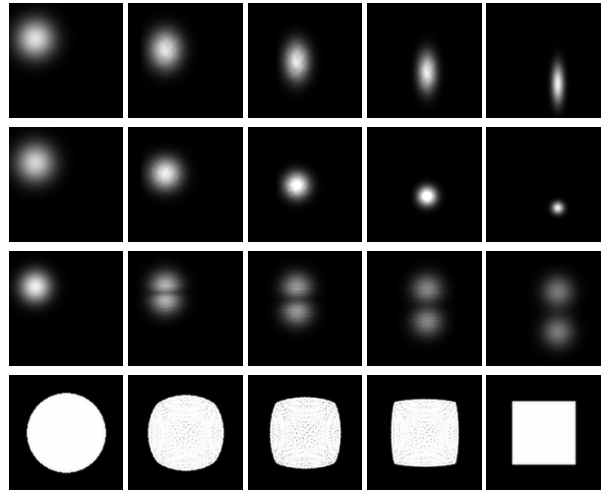

Figure 7: *Synthetic 2D examples on a Euclidean domain. The left and right columns show the input distributions, while the center columns show interpolations for $\alpha = 1/4$, $\alpha = 1/2$, and $\alpha = 3/4$.*

positive. A negative side effect of this choice is that interpolating between BRDFs of equal energy conserves their log energy (§ 3.6) instead of their energy. Because we apply a concave remapping, the interpolated value is guaranteed to be always lower, which ensures that our result does not break the energy preservation rule. That is, our interpolated BRDFs never reflect more light than they receive as long as the source and target BRDFs have the same property. Further, in our experiments, we measured only limited energy losses between 0.1% and 2%. Also, since energy preservation applies to the 2D slices representing the outgoing directions associated to a given incoming direction, we perform interpolation slice by slice. Reciprocity is not guaranteed in this process, but could be enforced in a postprocessing step. We use the squared geodesic distance on the sphere as the ground distance, which corresponds to using spherical linear interpolation on the paired particles. We render the results with PBRT [Pharr and Humphreys 2010].

Discussion Previous work on BRDF interpolation relies either on linear blending [Lensch et al. 2001] or on manifold learning [Matusik et al. 2003; Dong et al. 2010]. While simple, linear blending can exhibit significant visual artifacts (Fig. 1 and 8, and [Matusik et al. 2003]). Manifold-based interpolation addresses this shortcoming with a nonlinear space within which interpolation is performed. Building this space requires a large number of example BRDFs that may not be always available. Our approach provides an alternative that works with only two BRDFs. The "speed" of interpolation from the source to the target BRDF is uniform according to the geodesic metric on the sphere. However the perceived change is known to be related to properties of the material such as the frequency content of the BRDF [Pellacini et al. 2000; Wills et al. 2009]. This could be incorporated in our method by reparameterizing the interpolation parameter t according to a perceptual metric akin to the work of Ngan et al. [2006]. For very specular BRDFs, we observed RBF reconstruction errors of up to 15% thus slightly degrading their visual appearance. Adaptively adjusting the variance of each Gaussian according to the local frequency content could improve the quality in this specific case.

Validation and Experiments We test our method with a parametric BRDF model so that we can render reference images by interpolating the model parameters. We use the Ashikhmin-Shirley
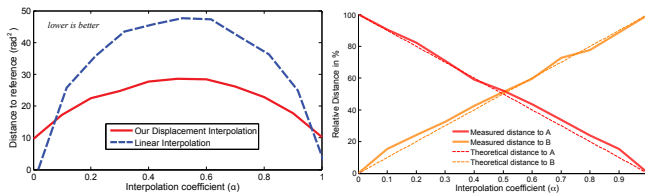
Figure 9: *Numerical validation on the anisotropic BRDF. Left: Error to the reference in terms of Wasserstein distance. Displacement interpolation performs up to 1.8 times as well as linear interpolation. Right: Demonstration that the Wasserstein distance to the source and target BRDFs is linearly interpolated.*

model of anisotropic material [2000] and generate two instances of brushed metal by varying only the anisotropic angle. The results are shown in Fig. 1 and Fig. 8. The expected result in this case is a BRDF with an anisotropic angle in between the source and target angles, which is what our approach achieves although it has access only to a set of samples and does not know about the parametric model. In comparison, linear blending produces implausible double highlights. The benefit of working on a non-Euclidean space such as a sphere is illustrated in Fig. 1(d) and Fig.8: an Euclidean space does not allow particles to follow the sphere geodesics, producing inconsistencies in the highlights. In Fig.9 (left), we measure the reconstruction error with respect to the reference in terms of Wasserstein distance. We validate the Wasserstein distance interpolation property in Fig.9 (right). We also test our method with measured BRDFs [Matusik et al. 2003]. We select a specular and a matte material and interpolated between them. Figure 10 shows a result close to the matte BRDF ($t = 0.8$). With linear interpolation, the material keeps a constant shininess and the amplitude of the specular lobe is reduced. In comparison, with our approach, the specular amplitude is reduced and at the same time, the lobe becomes wider and the material appears rougher. We believe that our method offers a complementary behavior that artists will find useful. We also experimented with the number of frequency bands in our representation (§ 3.4) and found that there is little gain in using more than one band (Fig.10) because BRDFs are relatively low frequency except for highly specular materials. In these examples, the kernel width is set according to the 5th nearest neighbor.

5.3 Value function interpolation

In character animation and reinforcement learning, value functions are a convenient way to compute and store optimal control policies. A value function stores the optimal discounted cost-to-go across a state space that is defined by the state of the character with respect to a goal. Meaningful interpolation of value functions would allow for existing control policies to be reused for new situations. In the example we shall demonstrate, this takes the form of an obstacle whose position is not known until run-time. We wish to interpolate between two reference solutions for the value function that are computed for two particular locations of the obstacle.

We perform interpolation experiments where the value function is computed on the 4D state space of an oriented particle. The 4D state space represents the particle position, angular velocity and sagital acceleration. Particle trajectories can be seen in Fig.11. Initial curls at the start of the trajectories are due to the particle initial velocities and their initial angular velocities. As shown in Figure 11 and in the accompanying video, the value function computed using displacement interpolation does significantly better at preserving the desired intent than the linearly interpolated value function. The particle can then be used to drive a 3D physically-based character (see
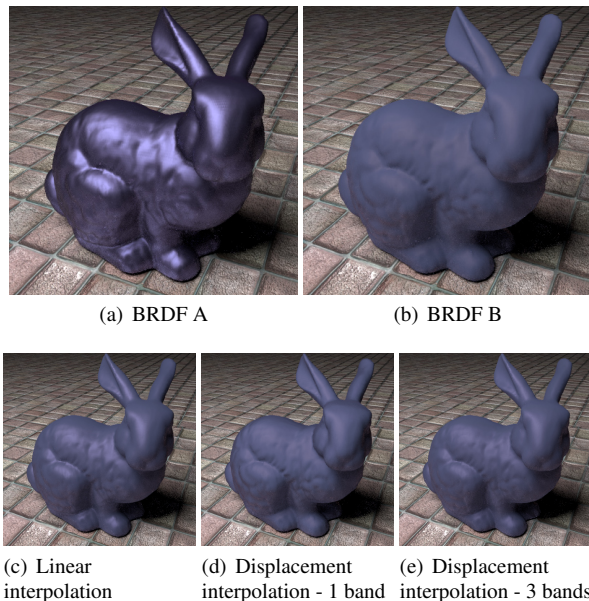

(a) BRDF A   (b) BRDF B

(c) Linear interpolation   (d) Displacement interpolation - 1 band   (e) Displacement interpolation - 3 bands

Figure 10: 80% *interpolation between two measured BRDFs. Even at* 80%*, the linear interpolation remains specular, while the 1 band displacement interpolation increased the roughness of the surface: linear blending cannot control the roughness of the underlying microgeometry, as could be desired. The target BRDF does not contain high frequencies to be matched: the multi-scale approach does not perform better than linear blending in this case.*

also the accompanying video). Although our method does not result in optimal trajectories, most of the particles achieved the expected interpolated goal location while avoiding the interpolated obstacle. Linear Bellman combination [da Silva et al. 2009] results in an optimal behavior, but yields two different targets and obstacles. We performed numerical comparison with the simulated reference (Fig. 12). For each particle, we integrate the contribution of the reference value function along its path. As the reference particles maximize this criterion, this provides a numerical reference score to which we can compare. As can be seen in Fig.12, linear interpolation fails at obtaining a good score at intermediate times since particles trajectory significantly diverge from the expected paths.

5.4 Color distribution interpolation

We apply our method to transfer and interpolate color histograms. Given a source and target image, we use clustering and solve the mass transport on the resulting discrete problem. The pairing between source and target particles define a remapping of the source colors onto the target colors, or any distribution interpolated using this pairing.

Discussion and Experiments In this particular context, our color transfer algorithm is similar to the method of Morovic and Sun [Morovic and Sun 2003]. However, their implementation relies on a transport simplex solver which cannot handle more than approximately 3000 particles, which results in a limited representation of the color space. With our network simplex solver, we can use 16 000 particles and remove the artifacts stemming from the coarse sampling imposed by the transport simplex (Fig. 13). We also compared our approach to the method by Pitié et al. [2005]. Among all the possible transfers, this method does not guarantee to minimize a given transport cost and thus cannot control the cost,
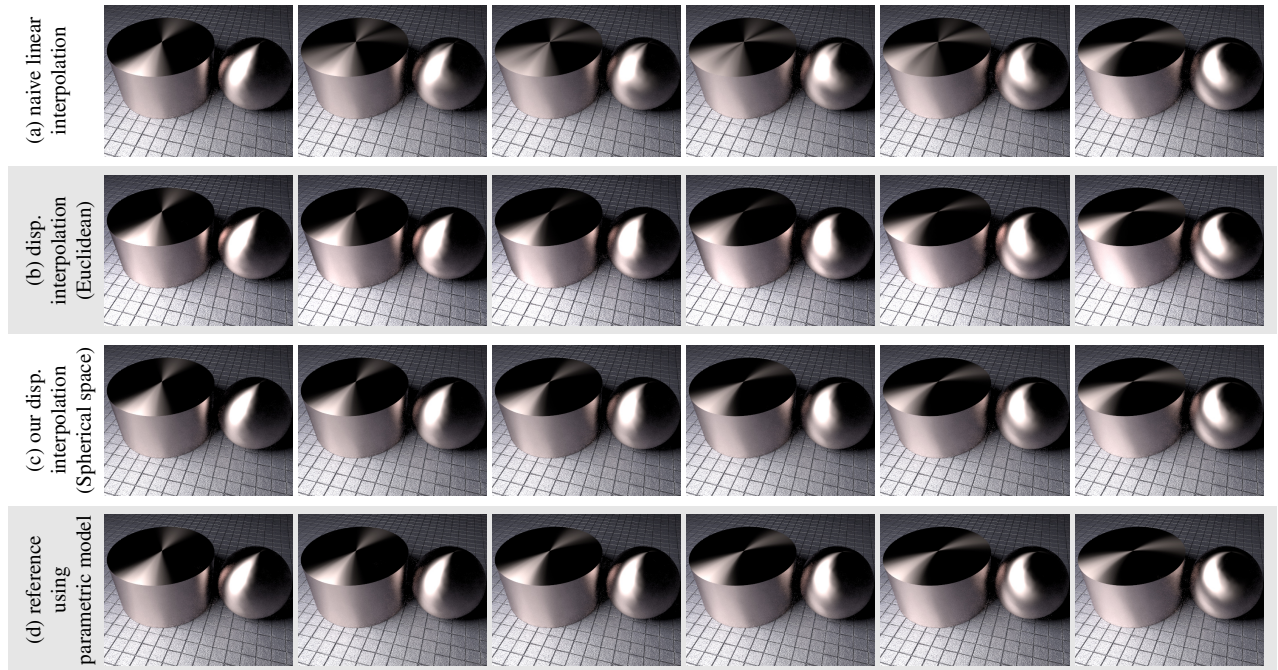
Figure 8: *Validation on an anisotropic Ashikhmin-Shirley synthetic BRDF. Naive interpolation (a) and our method (c) do not know about the underlying parametric model and are given only a sampled representation of the BRDFs. Naive linear interpolation (a) cross-fades the two BRDFs, which produces unrealistic double highlights. In comparison, our approach (c) rotates the highlights in a realistic way similar to interpolating the parameters of the model (d). In (b), we orthogonally projected the spherical BRDF samples on the 2D xy-plane and performed a displacement interpolation, exhibiting inconsistencies in the highlights.*
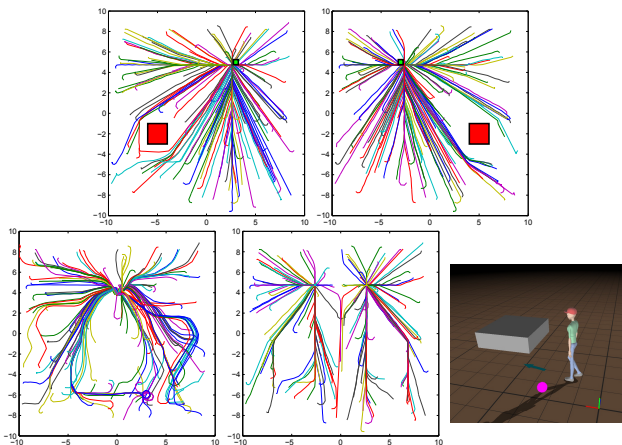

Figure 11: *Interpolation between two 4D value functions. The red square is an obstacle, the green square a target. From left to right, top to bottom: particles trajectories for reward A; for reward B; using our interpolated value function ($\alpha = 0.5$); using a linear reward interpolation; character driven by a particle.*

Figure 12: *Numerical validation on the value function. We measure the integral of the value function over each particle path and determine the deviation of particles with respect to their objective. High values indicate more optimal paths (100% is the reference).*

unlike ours. However we found that the results produced by both approaches look similar in practice when using a quadratic transport cost.

5.5 Animated Stipples

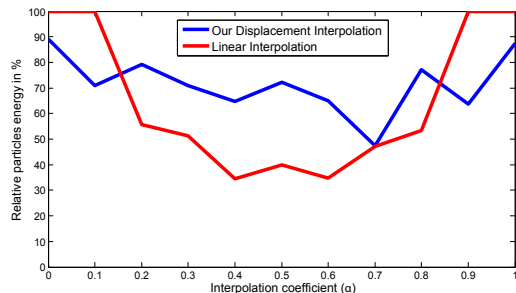Our method can also be applied to discrete functions. We illustrate this with stippling, an NPR method that generates images made of small primitives such as dots or small lines. We use our approach to warp one stippled image into another one by moving its primitives. The advantage of our approach is that it guarantees collision-free transformations (§ 2.2). We use the method of Secord et al. [2002] to generate particles on the source and target images. We set the method such that the number of source and target particles is the same. We solve the transport problem by assigning a unit mass to each particle. In this case, using the same number of particles with unit mass, mass transport reduces to an assignment problem and its solution does not require splitting or merging particles [Hillier and Lieberman 1990]. In our context, it ensures that a source stipple is associated to single target particle and reciprocally. For the ground distance, we use the standard L^2 distance that generates straight lines as geodesics. For small lines, their source and target orientations are computed using the input images, and linearly interpolated

Figure 13: *Our method can be used for histogram transfer as in [Morovic and Sun 2003]. Top row: source and target images. Bottom row: results. The use of an efficient network simplex allows the use of 16000 colors (bottom right) while the transportation simplex used in [Morovic and Sun 2003] only achieves up to 3072 colors (bottom left - artifacts due to color reduction are highlighted).*

between these values at intermediate frames. Our method extends the original work of Secord et al. [2002] that applies to still images and videos by enabling it to achieve warping between unrelated images. The results are shown in Figure 14, with additional results shown in the video. The use of a quadratic ground distance guarantees that the displacement interpolation produces particles with minimal kinetic energy, since it minimizes the sum of the squared traveled distances and since particles have constant velocity. We numerically evaluate this property: over the telephone sequence, the greedy matching produces a kinetic energy of 899.5 units, while the displacement interpolation halves this energy to 447.0 units.
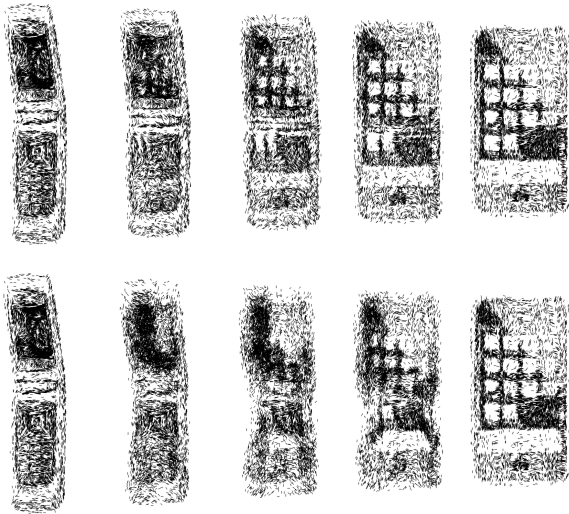

Figure 14: *Stipple interpolation using displacement interpolation (top) and greedy nearest neighbor interpolation (bottom).*

5.6 Environment map interpolation

We interpolate environment maps that are represented by a sum of directional light sources [Ostromoukhov et al. 2004]. We process each color channel separately and use the channel intensity as

weight. We use the squared L^2 norm on the sphere for the ground distance, similarly to the BRDF scenario, which makes the particle moves along sphere geodesics.

Validation and Experiments We used the LightGen plug-in of HDRShop [HDRShop] to decompose the environment maps into 300 light sources. We first test our approach using the parametric sky model of Preetham et al. [1999] to generate two different times of the day. We used linear blending and our method on the two environment maps to render images at intermediate times of the day (Fig. 15). Linear blending does not capture the motion of the sun and renders fixed shadows with only changing intensities. In the displacement interpolation result, shadows rotate similarly to the reference image obtained with the parametric model.

6 Limitations and future work

In this section, we describe limitations of the current approach and provide directions for extending our work.

Limitations The current implementation of the method cannot achieve real time interpolation for histograms larger than a few hundred bins. An interesting direction to alleviate this problem is the use of GPU as has been used for the Earth Mover's Distance computation [Rehman et al. 2009], although currently limited to voxels using a multigrid approach. While simplex computations have been performed on the GPU [Bieling et al. 2010], their performance is still much lower than efficient CPU implementations. Storing a full cost matrix between the two histograms remain a problem for current generation of GPUs. GPUs have also recently been used for non negative least square problems using QR factorization [Luo and Duraiswami 2010], which remain in $O(n^3)$ complexity. We did not test this method since our matrices do not fit in current GPU memory and that their problem size are much smaller than ours. While the preprocessing can be slow, the interpolation itself is fast, and is trivially parallelized as it merely consists in computing a sum of Gaussian for each sample. Detailed timings are provided in table 6.

We proposed a multiscale approach allowing the advection of different frequency bands. While the choice of the number of advected frequency bands can remain a user-defined choice, this can lead to an undesired linear interpolation in certain cases. For example, if BRDF A only contains high frequencies and BRDF B only contains low frequencies, the multiscale approach will advect the high frequencies of BRDF A toward a frequency band with zero energy in the BRDF B, hence fading out the high frequency band. This is illustrated in Fig.10. This behavior becomes more likely as we increase the number of frequency bands. The use of different frequency decompositions such as Empirical Mode Decomposition or based on local extrema [Subr et al. 2009] remains a challenge for scattered data, but could provide a better multiscale representation.

Finally, advecting too narrow gaussians can lead to aliasing artifacts (Fig.16). While such aliasing barely influence computations on PDFs (such as BRDF importance sampling) since the advected PDF remains close to the desired smooth PDF in a mass transport

Figure 16: *Disk to square example generated using overly narrow Gaussians, exhibiting aliasing artifacts.*
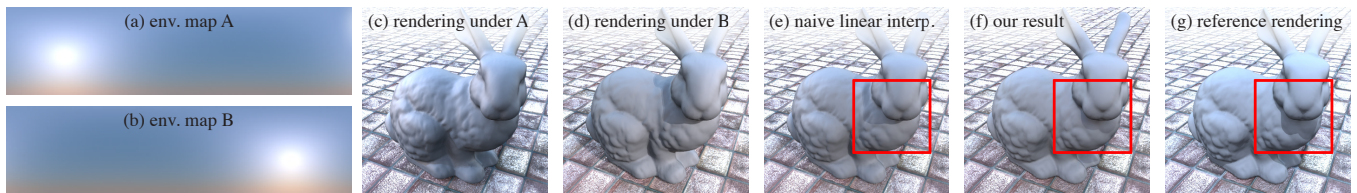
Figure 15: *Environment maps interpolation between two synthetic skies (a,b) using the parametric model of Preetham et al. [1999]. Linearly blending the environment maps produce fixed shadows that do not capture the rotation of the sun in the sky (e). Our approach properly captures this rotation although the height of the sun in the sky differs, yielding slightly shorter shadows (f). The non-monotonic variation in intensity and color of the sky across the day are not predicted from an interpolation scheme. We selected largely different skies for our test to make differences more visible. In a practical scenario, one would record environment maps more often during the day to capture these variations, and use our approach to smoothly interpolate them. These results are best seen in the accompanying video.*

Application	Data	# Bands	RBFs (Preprocessing)	EMD (Preprocessing)	Interpolation
Synthetic BRDF	500 slices x 1200 samples, 1 color	1	16 s	4 min 40s	25 s
Measured BRDF	500 slices x 1200 samples, 3 colors	1	8 min	2h	6 min
Value function	10830 samples in 4D	3	43 min	15 s	13 s
Stippling	8000 stipples	-	-	49 s	<1 s
Color Histograms	16000 bins	-	-	4 min 14s	<1 s

Table 1: *Timings for different applications*

sense, this is not the case for more general functions such as images or value functions.

Future work While our method currently only provides interpolation between any two distributions, it would be interesting to study the interpolation between N different distributions. This idea has recently been investigated for texture mixing [Rabin et al. 2010]. It should also be possible smoothly interpolate sequences of distributions using a method analogous to subdivision methods for modeling curves.

In the context of Value Function interpolation, we explored the displacement interpolation as a way to *extrapolate*. We believe this is a promising direction for further investigations.

7 Conclusion

Displacement interpolation offers a generic method of interpolating functions and distributions that in many situations provides more meaningful results than linear interpolation, while not requiring any domain knowledge. We have presented a practical Lagrangian method for displacement interpolation of continuous distributions. We have also shown how meaningful feature advection can often be achieved in the case of continuous functions using a multiscale approach. We show that a network simplex algorithm is in many cases preferable to the transportation simplex algorithm that is currently the most popular choice in EMD-applications. We have also shown that the 1D CDF interpolation arises as a special case of displacement interpolation. Lastly, we have demonstrated the utility of displacement interpolation in multiple applications.

Acknowledgements

References

AHMAD, N., HWA KIL, K., AND MCCANN, R., 2010. Optimal transportation, topology and uniqueness. arXiv:1008.4419v1.

AHUJA, R. K., MAGNANTI, T. L., AND ORLIN, J. B. 1993. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall.

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

BENAMOU, J. D., AND BRENIER, Y. 2000. A computational fluid mechanics solution to the Monge–Kantorovich mass transfer problem. *Numer. Math. 84*, 3, 375–393.

BIELING, J., PESCHLOW, P., AND MARTINI, P. 2010. An efficient GPU implementation of the revised simplex method. 1–8.

BURSAL, F. 1996. On interpolating between probability distributions. *Applied Mathematics and Computation 77*, 213–244.

CANTARELLA, J., AND PIATEK, M. 2004. Tsnnls: A solver for large sparse least squares problems with non-negative variables. *CoRR cs.MS/0408029*.

DA SILVA, M., DURAND, F., AND POPOVIĆ, J. 2009. Linear Bellman combination for control of character animation. *ACM Trans Graph (SIGGRAPH'09) 28*, 3, 1–10.

DONG, Y., WANG, J., TONG, X., SNYDER, J., LAN, Y., BEN-EZRA, M., AND GUO, B. 2010. Manifold bootstrapping for SVBRDF capture. *ACM Trans Graph (SIGGRAPH'10) 29*, 4, 1–10.

FLOOD, M. M. 1953. On the Hitchcock distribution problem. *Pacific J. Math. 3*, 2, 369–386.

HAKER, S., ZHU, L., TANNENBAUM, A., AND ANGENENT, S. 2004. Optimal mass transport for registration and warping. *International Journal on Computer Vision 60*, 3, 225–240.

HAN, C., SUN, B., RAMAMOORTHI, R., AND GRINSPUN, E. 2007. Frequency domain normal map filtering. *ACM Trans on Graphics (SIGGRAPH'07) 26*, 3, 28:1–28:12.

HDRSHOP,. LightGen plug-in. `http://gl.ict.usc.edu/HDRShop/lightgen/`.

HILLIER, F. S., AND LIEBERMAN, G. J. 1990. *Introduction to Operations Research, 5th ed.* McGraw-Hill.

KANTERS, F., PLATEL, B., FLORACK, L., AND ROMENY, B. M. T. H. 2003. Content based image retrieval using multiscale top points a feasibility study. In *Proc. of the 4th Int. Conf. on Scale space methods in Computer Vision*, 33–43.

KELLY, D. J., AND O'NEILL, G. M., 1991. The minimum cost flow problem and the network simplex method, September.

LAWSON, C. L., AND HANSON, R. J. 1995. *Solving least squares problems*, 3 ed.

LEMON, 2010. LEMON: Library for efficient modeling and optimization in networks. http://lemon.cs.elte.hu/trac/lemon.

LENSCH, H. P. A., KAUTZ, J., GOESELE, M., HEIDRICH, W., AND SEIDEL, H.-P. 2001. Image-based reconstruction of spatially varying materials. *Rendering Techniques*. Proc. of Eurographics Rendering Workshop.

LIPMAN, Y., AND DAUBECHIES, I. Conformal Wasserstein distances: comparing surfaces in polynomial time. *Advances in Mathematics (In Press)*.

LUO, Y., AND DURAISWAMI, R. 2010. Efficient parallel nonnegative least squares on the GPU. *submitted to the SIAM Journal on Scientific Computing*.

MACDONALD, D., 2006. A C++ implementation of the transportation simplex algorithm. `http://www.site.uottawa.ca/~dmacd070/emd/`.

MAKIHARA, Y., AND YAGI, Y. 2011. Earth mover's morphing: topology-free shape morphing using cluster-based emd flows. In *Proc. 10th Asian conference on Computer vision*, 202–215.

MATUSIK, W., PFISTER, H., BRAND, M., AND MCMILLAN, L. 2003. A data-driven reflectance model. *ACM Trans Graph (SIGGRAPH'03) 22*, 3, 759–769.

MATUSIK, W., ZWICKER, M., AND DURAND, F. 2005. Texture design using a simplicial complex of morphable textures. *ACM Trans Graph (SIGGRAPH'05) 24*, 3, 787–794.

MCCANN, R. J. 1997. A convexity principle for interacting gases. *Advances in Mathematics 128*, 153–179.

MOROVIC, J., AND SUN, P.-L. 2003. Accurate 3D image colour histogram transformation. *Pattern Recognition Letters 24*, 11, 1725 – 1735.

NGAN, A., DURAND, F., AND MATUSIK, W. 2006. Image-driven navigation of analytical brdf models. In *Proc. of the Eurographics Symposium on Rendering*.

OSTROMOUKHOV, V., DONOHUE, C., AND JODOIN, P.-M. 2004. Fast hierarchical importance sampling with blue noise properties. *ACM Trans Graphics (SIGGRAPH'04) 23*, 3, 488–495.

PELE, O., AND WERMAN, M. 2009. Fast and robust earth mover's distances. In *ICCV*.

PELLACINI, F., FERWERDA, J. A., AND GREENBERG, D. P. 2000. Toward a psychophysically-based light reflection model for image synthesis. In *Proc. ACM SIGGRAPH 2000*, 55–64.

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. 2005. N-dimensional probablility density function transfer and its application to colour transfer. *IEEE ICCV 2*, 1434–1439.

PREETHAM, A. J., SHIRLEY, P., AND SMITS, V. 1999. A practical analytic model for daylight. In *Proc. ACM SIGGRAPH 99*.

RABIN, J., PEYRÉ, G., DELON, J., AND BERNOT, M. 2010. Wasserstein Barycenter and its Application to Texture Mixing. Tech. rep.

READ, A. L. 1999. Linear interpolation of histograms. *Nuclear Instruments and Methods in Physics Research A 425* (Apr.), 357–360.

REHMAN, T. U., PRYOR, G., AND TANNENBAUM, A. 2007. Fast multigrid optimal mass transport for image registration and morphing. In *Proc. of the 18th British Machine Vision Conference*, 102–111.

REHMAN, T. U., HABER, E., PRYOR, G., MELONAKOS, J., AND TANNENBAUM, A. 2009. 3D nonrigid registration via optimal mass transport on the GPU. *Medical Image Analysis 13*, 6, 931 – 940.

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, 99–121.

RUBNER, Y., 1998. Code for the Earth Mover's Distance (EMD). `http://vision.stanford.edu/~rubner/emd/`.

RUSINKIEWICZ, S. 1998. A new change of variables for efficient BRDF representation. In *Rendering Techniques (Proc. Eurographics Workshop on Rendering)*.

SECORD, A., HEIDRICH, W., AND STREIT, L. 2002. Fast primitive distribution for illustration. In *Rendering Techniques (Proc. Eurographics Workshop on Rendering)*, 215–226.

SUBR, K., SOLER, C., AND DURAND, F. 2009. Edge-preserving multiscale image decomposition based on local extrema. *ACM Trans Graph (SIGGRAPH ASIA'09) 28*, 147:1–147:9.

TAN, P., LIN, S., QUAN, L., GUO, B., AND SHUM, H.-Y. 2005. Multiresolution reflectance filtering. In *Rendering Techniques*, 111–116.

VILLANI, C. 2003. *Topics in Optimal Transportation (Graduate Studies in Mathematics, Vol. 58)*. American Mathematical Society, March.

VILLANI, C. 2008. *Optimal Transport: Old and New*, 1 ed. Grundlehren der mathematischen Wissenschaften. Springer, November.

WILLS, J., AGARWAL, S., KRIEGMAN, D., AND BELONGIE, S. 2009. Toward a perceptual space for gloss. *ACM Trans Graph 28*, 4, 1–15.

ZHAO, Q., YANG, Z., AND TAO, H. 2010. Differential earth mover's distance with its applications to visual tracking. *IEEE Trans Patt. Analysis and Machine Intelligence 32*, 2, 274–287.

ZHU, L., YANG, Y., HAKER, S., AND TANNENBAUM, A. 2007. An image morphing technique based on optimal mass preserving mapping. *IEEE Trans Image Process 16*, 6 (06), 1481–1495.