Non-Euclidean Sliced Optimal Transport Sampling

Baptiste Genest, Nicolas Courty, David Coeurjolly

Volume 0 (1981), Number 0

Non-Euclidean Sliced Optimal Transport Sampling

Baptiste Genest[1], Nicolas Courty[2] and David Coeurjolly[1]

[1]Univ Lyon, CNRS, Lyon1, INSA, LIRIS, France
[2]Université Bretagne Sud, IRISA, CNRS, France

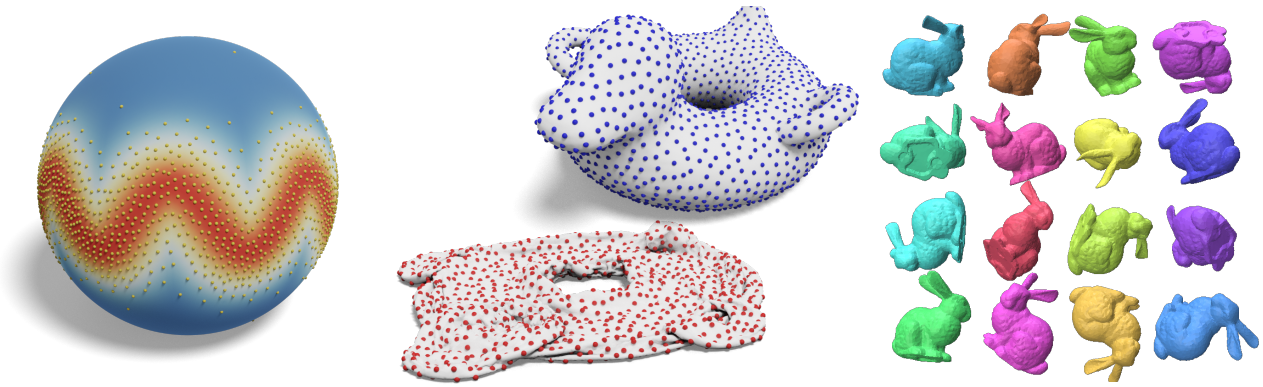


Figure 1: *We propose a new technique to generate well-dispersed samples on non-Euclidean domains (spherical, hyperbolic and projective spaces) using an extension of the sliced optimal transport sampling. As an example, this allows us to sample probability measures on the high-dimensional sphere* (left). *Using the uniformization theorem to conformally embed discrete manifolds to spherical or hyperbolic spaces, we can also generate blue noise samples in a purely intrinsic manner (red samples on the flatten geometry that exhibits blue noise properties when mapped back to a better embedding in* $\mathbb{R}^3$ *in blue). Finally, we also demonstrate that such an approach can be used to* blue noise *sample unit quaternions (hence rotations) on the projective space of dimension 3 (right).*

Abstract

In machine learning and computer graphics, a fundamental task is the approximation of a probability density function through a well-dispersed collection of samples. Providing a formal metric for measuring the distance between probability measures on general spaces, Optimal Transport (OT) emerges as a pivotal theoretical framework within this context. However, the associated computational burden is prohibitive in most real-world scenarios. Leveraging the simple structure of OT in 1*D, Sliced Optimal Transport (SOT) has appeared as an efficient alternative to generate samples in Euclidean spaces. This paper pushes the boundaries of SOT utilization in computational geometry problems by extending its application to sample densities residing on more diverse mathematical domains, including the spherical space* $\mathbb{S}^d$*, the hyperbolic plane* $\mathbb{H}^d$*, and the real projective plane* $\mathbb{P}^d$*. Moreover, it ensures the quality of these samples by achieving a blue noise characteristic, regardless of the dimensionality involved. The robustness of our approach is highlighted through its application to various geometry processing tasks, such as the intrinsic blue noise sampling of meshes, as well as the sampling of directions and rotations. These applications collectively underscore the efficacy of our methodology.*

CCS Concepts

• ***Computing methodologies*** → ***Computer graphics;***

1. Introduction

In recent years, Optimal Transport has become a key mathematical framework for manipulating generalized probability density functions (*e.g.* [V*09]). The most general way to describe the interest of OT is that it allows quantifying meaningfully how costly it is to move masses from a generalized probability density function to another one. This defines a natural notion of distance between

probability measures, the Wasserstein distance, allowing the design of displacement interpolations between measures or when dealing with more than two measures, the notion of Wasserstein barycenter.

The high versatility of the framework and the numerous developments of efficient numerical solvers make the OT become standard in many machine learning [HGK*16, CFTR16, ACB17],computer vision, or computer graphics applications [DGBOD12, SRGB14, SdGP*15,BRPP15,QCHC17,NG18,BC19,PBC*20,SGSS22] (see [BD23] for a recent survey).

Among computer graphics applications, OT has become a widely spread tool for point pattern design and Monte Carlo integration [QCHC17, PBC*20, SGSS22]. The main argument is that OT offers a mathematical framework to characterize well-distributed, or blue noise, samples in a domain leading to an efficient Monte Carlo integration or signal reconstruction [SÖA*19]. This can be achieved by optimizing the samples positions such that the Wasserstein distance to the uniform measure in the domain is minimized. More recently, OT on non-Euclidean spaces has been developed in the machine learning context, as it allows efficiently processing of data for which a spherical or hyperbolic geometry is a natural representation space [BBC*22, BCDC22]. In geometry processing, a spherical or hyperbolic embedding of geometrical objects can be at the core of many surface parametrization, texture mapping or shape matching problems [HAT*00, GY03, GGS03, KSS06, CPS13, BCK18, SCBK20, GSC21]. The challenge addressed in this paper is the design of an OT driven sampling techniques on Riemannian manifolds with applications to computer graphics.

Contributions. Relying on sliced optimal transport formulation for the sphere and the hyperbolic space formulated by Bonet et al. [BBC*22, BCDC22], we propose a blue noise sampling strategy of probability measures on these non-Euclidean spaces. This is achieved by providing explicit formulas for the samples advection steps and direction pooling in a Riemannian gradient descent approach. We then demonstrate the strength of the approach to efficiently sample meshes through the uniformization theorem allowing transforming the intrinsic blue noise sampling problem on the mesh, to a blue noise sampling problem in $\mathbb{S}^2$ or $\mathbb{H}^2$ depending on the mesh topology. We also highlight the interest of the approach through projective plane sampling that can be used to sample 3D rotations (by sampling quaternions in 4d), as well as various geometric objects befined by projective equations (*e.g.* lines, directions...).

2. Background

Optimal transport. Given two measures μ and ν, over some domain Ω, and a function $c(x,y)$ that dictates the cost of moving a particle from x to y in Ω, one can define the Optimal Transport problem from μ to ν as

$$\min_{\pi\in\Pi(\mu,\nu)} \int_\Omega c(x,y)d\pi(x,y)\,. \tag{1}$$

where $\Pi(\mu,\nu)$ is the set of couplings:

$$\{\pi\in\mathcal{P}(\Omega\times\Omega), \forall A\subset\Omega, \pi(A\times\Omega)=\mu(A), \pi(\Omega\times A)=\nu(A)\}\,.$$

In most contexts, $c(x,y)=d^p(x,y)$ where d is a distance on Ω (*e.g.* [PC*19]). In such cases we call the minimum cost the $p-$Wasserstein distance between μ and ν, $W_p^p(\mu,\nu)$. The interest of using measures is that its general enough to handle both discrete and continuous objects at the same time. Depending on the nature of the measures, discrete-to-discrete, semi-discrete, or continuous-to-continuous, a huge literature exists on numerical methods to efficiently solve OT problems [PC*19, FCG*21].

Sliced Optimal Transport. Among alternative numerical methods, we are interested in fast approximation techniques that scale up with the size of the discrete problem and the dimension. First, we observe that the one-dimensional OT problem admits the following closed form solution:

$$W_p^p(\mu,\nu)=\int_0^1 |F_\mu^{-1}(u)-F_\nu^{-1}(u)|^p du\,, \tag{2}$$

where F_μ is the cumulative function of the 1D density μ, and F_μ^{-1} its generalized inverse, or quantile function. For $p=1$, one can derive the equivalent formula:

$$W_1(\mu,\nu)=\int_0^1 |F_\mu(u)-F_\nu(u)|du\,. \tag{3}$$

The transport plan is then simply given by associating the ith point of μ to the ith point of ν (see for example [PC*19]) in the case when μ and ν are both discrete with the same number of atoms. The obtained result is the mapping that minimizes the cost to transport μ to ν. Hence, a very natural idea is to break a d dimensional OT problem into an infinity of 1 dimensional one. Such an approach is referred to as *Sliced Optimal Transport* since it amounts to projecting the measures onto 1D *slices* [PKD05, RPDB11, BRPP15]. Given a direction $\theta\in\mathbb{S}^{d-1}$ and the projection $P^\theta(\mathbf{x}):=\langle\mathbf{x},\theta\rangle$ of any , for all $\mathbf{x}\in\mathbb{R}^d$, the sliced Wasserstein distance is defined as

$$SW_p^p(\mu,\nu):=\int_{\mathbb{S}^{d-1}} W_p^p(P^\theta_\#\mu, P^\theta_\#\nu)\,d\lambda(\theta)\,, \tag{4}$$

where $P^\theta_\#\mu$ is the image measure of μ by the projection operator. The sliced approach receives a lot of attention in the literature as it is topologically equivalent to OT [NDC*20] with bounded approximation of W_p [Bon13]. On the algorithmic side, the integral over $\mathbb{S}^{d-1}$ is obtained used a Monte Carlo approach: we draw random directions uniformly on $\mathbb{S}^{d-1}$ and accumulate 1d Wasserstein distances. The computational advantage is that each 1d slice W_p^p only requires to sort the points, leading to an overall computation cost in $\mathcal{O}(K\cdot n(d+\log(n)))$ time complexity if K denotes the number of slices used in the Monte Carlo estimation.

Sliced Optimal Transport Sampling (SOTS). In the context of Monte Carlo sampling, Paulin et al. [PBC*20] leveraged the Euclidean sliced optimal transport formulation to optimize a point set such that it better approximates a given target distribution, in the sense of the SW_2 metric. In this Monte Carlo rendering setting, given a target measure ν in $[0,1)^d$ (uniform measure for blue noise sampling), the objective is to construct n samples $\{\mathbf{x}_i\}\in[0,1)^d$ defining the discrete distribution $\mu=\sum_{i=1}^n \delta_{\mathbf{x}_i}$, such that $SW_2(\mu,\nu)$ is minimized. One iteration of the sliced optimal transport sampling, SOTS for short, algorithm is the following, if $\mu=\sum_{i=1}^n \delta_{\mathbf{x}_i}$

and if ν is a continuous measure with closed form projection formula on a line (mainly the uniform measure over a ball or a square), we iterate:

$$\mathbf{x}_i^{(K+1)} = \mathbf{x}_i^{(K)} + \frac{\gamma}{L}\sum_{l=1}^{L}\left(T_l\left(P^{\theta_l}(\mathbf{x}_i^{(K)})\right) - P^{\theta_l}\left(\mathbf{x}_i^{(K)}\right)\right), \quad (5)$$

where T_l is the transport plan associated with the solution of the continuous-to-discrete problem between $P^{\theta_l}_{\#}\nu$ and $P^{\theta_l}_{\#}\mu$ and $\gamma > 0$ is a step size (see Fig.2-left). For the sake of simplicity, the $P^{\theta}(\mathbf{x})$ notation refers to the projection of the sample $\mathbf{x}$ onto the slice θ (*i.e.* $P^{\theta}_{\#}\mu = \sum_i \delta_{P^{\theta}(\mathbf{x}_i)}$). Intuitively, we move each point in the direction of the slice proportionally to the distance to its projected 1d optimal mapping. In [PBC*20], the authors have demonstrated the interest of such blue noise sampling in $[0,1)^d$ for Monte Carlo integration and Monte Carlo rendering. This paper extends this approach to non-Euclidean metric spaces.

Non-Euclidean Sliced Wasserstein Distance. Bonet et al. extend the SW distance to Spherical [BBC*22] and Hyperbolic metric spaces [BCDC22], by replacing the Euclidean notions of lines and projections with the Riemannian equivalent of projection over geodesics. Namely, the spherical geodesics are great-circles of the sphere and geodesics passing through the origin of any hyperbolic model are valid replacements. With these constructions at hand, authors perform various machine learning tasks where the SW distance is generally used as a data fitting loss or a meaningful metric to compare objects defined over such spaces.

Blue Noise Mesh Sampling. Blue noise sampling of surfaces in $\mathbb{R}^3$ is one of our targeted applications. On Euclidean domains, a classical approach to construct well-spread samples in a domain consists in making sure that each pair of samples are separated by at least a given minimum distance. Dart throwing and its variations [Bri07] have naturally been extended to manifolds to achieve such Poisson disk sampling [CJW*09, BWWM10, Yuk15, GYJZ15]. Alternatively, Voronoi diagrams driven approaches [LWL*09a, BSD09] and their restriction of discrete manifolds (triangular meshes in most cases), have been used to construct blue noise samples [LWL*09b, XHGL12, AGY*17, XLC*16]. While focusing on remeshing applications, Peyré and Cohen [PC06] have proposed an instrinsic sampling strategy that inserts samples one by one at the location maximizing the (geodesic) distance from the previous samples (approach denoted farthest-point, FP, below). While being efficient from an FMM approximation of the geodesic distance, this algorithm has a greedy approach and is not fit to sample generic non uniform densities. Starting from an initial sampling and pairwise (geodesic) distances between samples, Qin et al. [QCHC17] optimized samples position so that the regularized optimal transport distance between the samples and the uniform measure on the manifold is minimized. Particle based systems can be designed by optimizing the sample distribution on a mesh to unformize the distances between neighboring samples in ambient space, while staying close to the surface thanks to a projection operator [TMN*00, ZGW*13, JZW*15]. Samples could also be optimized such that they capture the spectral content of the targeted surfaces [ÖAG10]. In most cases, for efficiency purposes, the sampling is performed in ambient space and later projected onto the manifold. While those techniques can be very efficient in terms of blue noise quality when the mesh embedding to $\mathbb{R}^3$ is *ambient-compatible* (no too-close sheets of meshes or large enough local shape diameter function [SSCO08], Euclidean unit balls is a good approximation of the geodesic ones...), we propose an efficient purely intrinsic blue noise sampling that can deal with shapes with incorrect embedding (see Fig. 1).

3. Sliced optimal transport sampling on constant curvature manifolds

We first extend the SOTS approach defined on Euclidean domains, to the spherical and hyperbolic cases in arbitrary dimensions, respectively denote $\mathbb{S}^d$ and $\mathbb{H}^d$ (see Fig. 2).

To define the SOTS in such non-Euclidean spaces, we first need to refine the notion of *projection onto a straight line* as the projection of a set of samples onto geodesic slices for the targeted model (Sec. 3.1). Then we need to solve the matching 1d problem on the geodesic slice (Sec. 3.2). These key ingredients are mostly borrowed from Bonet et al. [BBC*22, BCDC22] dedicated to the computation of *SW* on $\mathbb{S}^d$ and $\mathbb{H}^d$. We extend these works with explicit formulas to perform the advection of the samples using group action principle (Sec. 3.3) and Exp and Log maps (Sec. 3.4). Finally, Section 3.5 completes the algorithm describing the extension of the gradient descent of the SW_2 energy. In Section 3.7, we describe a technical improvement of the advection step on batches using a geometric median instead of an average as usually used in SOTS. We summarize the generic algorithm in Alg. 1. Note that we consider a discrete target measure $\nu = \sum_{i=1}^{m}\delta_{\mathbf{y}_i}$ with a number of Diracs m that may be greater than n. This will be discussed in Section 3.6 to allow the sampling of non-uniform densities. Starting from line 5, we thus solve a balanced optimal transport problem as $\tilde{\nu}$ is a random sampling of ν with exactly n Diracs.

3.1. Geodesic slices and projections

The first step is to find an equivalent to straight lines in the Euclidean space. The most natural choice is a geodesic passing through the origin of the model. In both $\mathbb{S}^d$ and $\mathbb{H}^d$ cases, such an object can be obtained by the intersection of a plane with the canonical embedding of each space in $\mathbb{R}^{d+1}$.

Spherical geometry. As proposed by Bonet et al. [BBC*22], random slices are defined by the intersection of $\mathbb{S}^d$ by uniformly sampled Euclidean 2D planes in $\mathbb{R}^{d+1}$ passing through the origin. This is done by generating two $(d+1)-$dimensional vectors with components in $\mathcal{N}(0,1)$, that we orthonormalize (by Gram-Schmidt or Givens rotations). We denote by $\theta = \{\mathbf{e}_1, \mathbf{e}_2\}$ the two vectors in $\mathbb{R}^{d+1}$ generated by this process. Such basis of the plane allows defining the projection in $\mathbb{R}^{d+1}$ onto the associated subspace $\mathrm{span}(\mathbf{e}_1, \mathbf{e}_2)$:

$$\Pi^{\theta}(\mathbf{x}) = \langle \mathbf{x}, \mathbf{e}_1\rangle \mathbf{e}_1 + \langle \mathbf{x}, \mathbf{e}_2\rangle \mathbf{e}_2 . \quad (6)$$

The projection onto the great circle $= \mathrm{span}(\mathbf{e}_1, \mathbf{e}_2) \cap \mathbb{S}^d$ becomes

$$P^{\theta}(\mathbf{x}) := \frac{\Pi^{\theta}(\mathbf{x})}{\|\Pi^{\theta}(\mathbf{x})\|} . \quad (7)$$

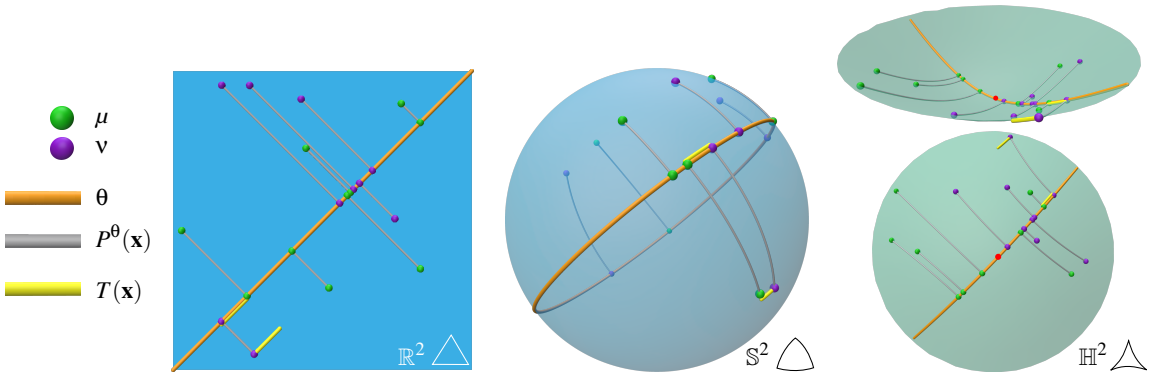


Figure 2: ***Sliced optimal transport sampling and notations****: from left to right, on the Euclidean domain (zero curvature metric space), on the spherical one (positive constant curvature metric space), and on the hyperbolic model (Lorentz's model with only a part of the hyperboloid, negative curvature metric space). We only illustrate the assignment and the associated advection for a single sample (yellow bars).*

Algorithm 1: Non Euclidean Sliced Optimal Transport Sampling – NESOTS

Data: The discrete target distribution $\nu = \sum_{i=1}^m \delta_{\mathbf{y}_i}$, the number of iterations K, the batch size L, the gradient descent step γ
Result: The discrete distribution $\mu^{(K)}$ after K iterations.

$\mu^{(0)} = \text{SubSample}(\tilde{\nu}, n)$; // Init.
for $j \in [[1,K]]$ **do**
  parallel for $l \in [[1,L]]$ **do** // Batch
    $\tilde{\nu} = \text{SubSample}(\tilde{\nu}, n)$; // Sec. 3.6
    $\theta = \text{RandomSlice}()$; // Sec. 3.1
    $\tilde{\nu}_\theta = P^\theta\left(\tilde{\nu}^l\right)$; // Sec. 3.1
    $\mu_\theta = P^\theta\left(\mu^{(j)}\right)$; // Sec. 3.1
    $T = \text{Solve1DOT}(\mu_\theta, \tilde{\nu}_\theta)$; // Sec. 3.2
    for $i \in [[1,n]]$ **do**
      $\mathbf{g} = \Gamma_\theta\left(P^\theta\left(\mathbf{x}_i^{(j)}\right), T\left(P^\theta\left(\mathbf{x}_i^{(j)}\right)\right)\right)$; // Sec. 3.3
      $\mathbf{d}_i^l = \text{Log}_{\mathbf{x}_i^{(j)}}\left(\mathbf{g}\left(\mathbf{x}_i^{(j)}\right)\right)$; // Sec. 3.4
    end
  end
  parallel for $i \in [[1,n]]$ **do**
    $\mathbf{d}_i = \text{GeoMed}\left(\{\mathbf{d}_i^l\}_L\right)$; // Sec. 3.7
    $\mathbf{x}_i^{(j+1)} = \text{Exp}_{\mathbf{x}_i^{(j)}}(\gamma \mathbf{d}_i)$; // Sec. 3.5
  end
end
return $\mu^{(K)} = \sum_{i=1}^m \delta_{\mathbf{x}_i^{(K)}}$

Hyperbolic geometry. The $d-$dimensional hyperbolic plane $\mathbb{H}^d$ admits many isometric models (*e.g.* the Poincaré disk or the Lorentz's hyperboloid models) [Lee06]. For the sake of simplicity of the associated formulas and numerical reasons, we will be using the hyperboloid model, i.e., the upper sheet of the hyperbola

$$\mathbb{H}^d := \{\mathbf{x} \in \mathbb{R}^{d+1}, \langle \mathbf{x}, \mathbf{x} \rangle_{\mathbb{L}} = -1\},$$

where $\langle \mathbf{x}, \mathbf{y} \rangle_{\mathbb{L}} := \sum_{i=1}^d x_i y_i - x_{d+1} y_{d+1}$ is the Lorentzian dot product. We denote by $\mathbf{x}_O$ the origin of the hyperbola (red dot in Fig. 2), i.e., $\mathbf{x}_O = (0, \dots, 0, 1)^t$. We follow Bonet et al. [BCDC22] by defining the projection on the geodesic obtained as the intersection between a 2D plane containing $\mathbf{x}_O$ and the hyperboloid. The sampling of uniform slices is achieved by sampling uniformly the space orthogonal to $\mathbf{x}_O$, i.e. $\mathbf{d} \sim \mathcal{U}(\mathbb{S}^d \times \{0\})$. We then have the projector

$$P^\theta(\mathbf{x}) := \frac{\Pi^\theta(\mathbf{x})}{\sqrt{-\langle \Pi^\theta(\mathbf{x}), \Pi^\theta(\mathbf{x}) \rangle_{\mathbb{L}}}}, \tag{8}$$

where we denote by $\theta := \{\mathbf{d}, \mathbf{x}_O\}$ the generator of the 2D slice in $\mathbb{H}^d$.

3.2. Solving the discrete 1D Wasserstein problem

As we will need to evaluate the transport cost on projected samples onto the sliced θ, we need to clarify the distances between two points in $\mathbb{S}^d$ or $\mathbb{H}^d$, and the coordinate on their projection onto θ, denoted $t_\theta(\mathbf{x})$, the signed geodesic distance to a given origin in θ .

Spherical geometry On the $d-$dimensional unit sphere, geodesics are great circles (intersection of a 2-plane passing through the origin, and $\mathbb{S}^d$). The geodesic distance between two points $\mathbf{x}, \mathbf{y} \in \mathbb{S}^d$ is simply the angle between the two vectors from the origin to the points

$$d_{\mathbb{S}}(\mathbf{x}, \mathbf{y}) := \arccos(\langle \mathbf{x}, \mathbf{y} \rangle). \tag{9}$$

As projections lie on a circle, any origin on θ can be considered to define t_θ. If $\theta = \{\mathbf{e}_1, \mathbf{e}_2\}$, we use

$$t_\theta(\mathbf{x}) := \frac{\pi + \arctan2(\langle \mathbf{e}_2, \mathbf{x} \rangle, \langle \mathbf{e}_1, \mathbf{x} \rangle)}{2\pi}. \tag{10}$$

On $\mathbb{S}^d$, the optimal transport problem needs to take into account the periodicity of the space, and its associated coordinate systems. Fortunately, it can be shown [DRG09] that the problem still boils down to a simple sorting of the samples coordinates t_θ provided that the circle is identified to the Real line through an optimal cut. Finding the optimal cut can be formulated as a weighted median problem, as detailed in Cabrelli et al. [CM98], and admits a $\mathcal{O}(n \log(n))$ solution. For some μ, ν in $\mathbb{S}^d$ and $\mathbf{x} \in \mu$, the map $T(P^\theta(\mathbf{x}))$ denotes the optimal assignment on the slice θ of $\mathbf{x}$ to some $\mathbf{y} \in \nu$.

Hyperbolic geometry On $\mathbb{H}^d$, the geodesic distance between two points is

$$d_{\mathbb{H}}(\mathbf{x},\mathbf{y}) := \operatorname{arccosh}(-\langle \mathbf{x},\mathbf{y}\rangle_{\mathbb{L}}). \tag{11}$$

Since the slice is directed by $\mathbf{d}$, we define the geodesic distance coordinate induced by $\mathbf{d}$

$$t_\theta(\mathbf{x}) := \operatorname{sign}(\langle \mathbf{x},\mathbf{d}\rangle) d_{\mathbb{H}}(\mathbf{x}_O,\mathbf{x}). \tag{12}$$

On $\mathbb{H}^d$, the optimal assignment is simply obtained by sorting the projected samples on θ and mapping the first projected sample in $P^\theta_\#\mu$ to the first one in $P^\theta_\#\nu$ (with respect to t_θ), similarly to the Euclidean case.

3.3. Transitivity and group action

In the Euclidean space, samples are advected by a simple translation in the straight line direction by the distance $t_\theta(\mathbf{x}) - t_\theta(T(P^\theta(\mathbf{x})))$. In spherical (Eq. (13)) and hyperbolic (Eq. (14)) domains, we rely on group actions. More precisely, we are interested in group actions that preserve the geodesics.

Spherical Geometry The right group to act on the sphere is $SO(d)$, i.e., the group of all $d-$dimensional rotations. One can build the rotation that maps a point $\mathbf{x}$ to a point $\mathbf{y}$ in $\mathbb{S}^d$ simply by building the 2D rotation in their common span, $\operatorname{span}(\{\mathbf{x},\mathbf{y}\})$, i.e

$$\begin{pmatrix} \cos(\varphi) & -\sin(\varphi) \\ \sin(\varphi) & \cos(\varphi) \end{pmatrix},$$

for some $\phi \in \mathbb{R}$. To make sure that the part of the vector orthogonal to $\operatorname{span}(\mathbf{x},\mathbf{y})$ is left unchanged and to avoid building the $d \times d$ matrix, we decompose any vector $\mathbf{w}$ in the orthonormal basis given as the result of the Gram-Schmidt algorithm applied to $\mathbf{x}$ and $\mathbf{y}$. Leading to

$$\begin{aligned}\Gamma_\theta(\mathbf{x},\mathbf{y}) : \mathbf{w} \to \mathbf{w}^\perp &+ \mathbf{x}(\cos(\varphi)w_x - \sin(\varphi)w_y) \\ &+ \tilde{\mathbf{y}}(\sin(\varphi)w_x + \cos(\varphi)w_y),\end{aligned} \tag{13}$$

where $\tilde{\mathbf{y}} = \mathbf{y} - \langle \mathbf{x},\mathbf{y}\rangle \mathbf{x}$, $w_x = \langle \mathbf{w},\mathbf{x}\rangle$, $w_y = \langle \mathbf{w},\tilde{\mathbf{y}}\rangle$, $\mathbf{w}^\perp$ is the component of $\mathbf{w}$ orthogonal to $\operatorname{span}(\{e_1,e_2\})$ and $\varphi = d_{\mathbb{S}}(\mathbf{x},\mathbf{y})$. One can verify that we have $\Gamma_\theta(\mathbf{x},\mathbf{y})(\mathbf{x}) = \mathbf{y}$. It is also possible to check that a rotation of φ degree along the slice θ applied to $\mathbf{x}$ will offset $t_\theta(\mathbf{x})$ by φ (modulo 1). Hence, it is indeed a translation along the slice, which is the behavior we wanted to translate from the Euclidean setting.

Hyperbolic Geometry As a direct analogy, translations along hyperbolic slices are hyperbolic rotations, i.e., the elements of the Lorentz group $SO_0(d-1,1)$ (standard rotations preserve the Euclidean scalar product whereas hyperbolic ones preserve $\langle\cdot,\cdot\rangle_{\mathbb{L}}$, hence the hyperboloid). Computationally, it is very similar to the spherical case, we want to apply the following 2D rotation in the $\operatorname{span}(\mathbf{x},\mathbf{y})$:

$$\begin{pmatrix} \cosh(\varphi) & \sinh(\varphi) \\ \sinh(\varphi) & \cosh(\varphi) \end{pmatrix},$$

leading to the analogous decomposition along the right subspaces:

$$\begin{aligned}\Gamma_\theta(\mathbf{x},\mathbf{y}) : \mathbf{w} \to \mathbf{w}^\perp &+ \mathbf{d}(\cosh(\varphi)w_d + \sinh(\varphi)w_0) \\ &+ \mathbf{x}_O(\sinh(\varphi)w_d + \cosh(\varphi)w_0),\end{aligned} \tag{14}$$

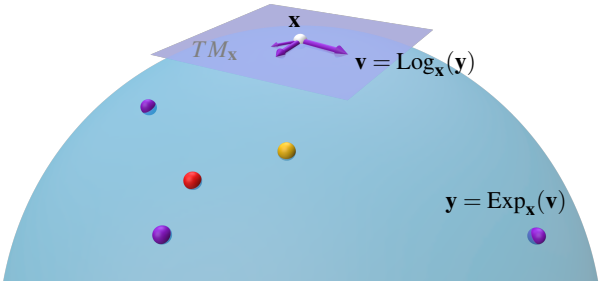


Figure 3: ***Exp and Log maps:*** *on $\mathbb{S}^2$, the orange point is the point obtained by iteratively going in the average of the Logs $x_{n+1} = Exp_{x_n}(\frac{\gamma}{n}\sum_i Log_{x_n}(y_i))$, which is equivalent to Fréchet means, whereas the red one is obtained by going in the geometric median of the directions $x_{n+1} = Exp_{x_n}(\gamma\, GeoMed(\{Log_{x_n}(y_i)\}_i))$.*

where $\mathbf{d} = \frac{\Pi_{x_0^\perp}(\mathbf{y}-\mathbf{x})}{\|\Pi_{x_0^\perp}(\mathbf{y}-\mathbf{x})\|}$, $w_d = \langle \mathbf{w},\mathbf{d}\rangle$, $w_0 = \langle \mathbf{w},\mathbf{x}_O\rangle$, $\mathbf{w}^\perp$ is the component of $\mathbf{w}$ orthogonal to $\operatorname{span}(\mathbf{x}_O,\mathbf{d})$ and $\varphi = d_{\mathbb{H}}(\mathbf{x},\mathbf{y})$. The only difference being that we decompose along $\mathbf{x}_O$ and $\mathbf{y}-\mathbf{x}$ instead of directly $\mathbf{x}$ and $\mathbf{y}$ (which gives the same span) to make sure that the points remain on the hyperboloid. We also have $\Gamma_\theta(\mathbf{x},\mathbf{y})(\mathbf{x}) = \mathbf{y}$.

3.4. Exp and Log Maps

Beside group actions, Exp and Log maps are key ingredients in Riemannian geometry [Lee06] (see illustration Fig. 3). The $\operatorname{Exp}_{\mathbf{x}}(\mathbf{v})$ map allows one to follow the geodesic γ, satisfying $\gamma(0) = \mathbf{x}$ and $\dot\gamma(0) = \mathbf{v} \in T\mathcal{M}_{\mathbf{x}}$, i.e., following the most natural path going from $\mathbf{x}$ with initial direction and velocity $\mathbf{v}$ from $t = 0$ to $t = 1$. Conversely, the $\operatorname{Log}_{\mathbf{x}}(\mathbf{y}) \in T\mathcal{M}_{\mathbf{x}}$ map, the inverse of $\operatorname{Exp}_{\mathbf{x}}$, gives the direction (and velocity) to go from $\mathbf{x}$ to $\mathbf{y}$, i.e. $\operatorname{Exp}_{\mathbf{x}}(\operatorname{Log}_{\mathbf{x}}(\mathbf{y})) = \mathbf{y}$. In $\mathbb{S}^d$ and $\mathbb{H}^d$, Exp and Log maps admit closed form expressions.

Spherical geometry. If $\Pi_{T\mathcal{M}_{\mathbf{x}}}$ denotes the projections from $\mathbb{R}^d$ onto the tangent space of $\mathbb{S}^d$ at $\mathbf{v}$, we have

$$\operatorname{Exp}_{\mathbf{x}}(\mathbf{v}) = \cos(\|\mathbf{v}\|)\mathbf{x} + \sin(\|\mathbf{v}\|)\frac{\mathbf{v}}{\|\mathbf{v}\|}, \tag{15}$$

$$\operatorname{Log}_{\mathbf{x}}(\mathbf{y}) = \frac{\Pi_{T\mathcal{M}_{\mathbf{x}}}(\mathbf{y}-\mathbf{x})}{\|\Pi_{T\mathcal{M}_{\mathbf{x}}}(\mathbf{y}-\mathbf{x})\|} d(\mathbf{x},\mathbf{y}), \tag{16}$$

(see Alimisis et al.'s supplemental [ADVA21]).

Hyperbolic geometry. In the Lorentz hyperbolic model, we have similar expressions (see *e.g.* Dai et al. [DWGJ21]):

$$\operatorname{Exp}_{\mathbf{x}}(\mathbf{v}) = \cosh(||\mathbf{v}||_{\mathbb{L}})\mathbf{x} + \sinh(||\mathbf{v}||_{\mathbb{L}})\frac{\mathbf{v}}{||\mathbf{v}||_{\mathbb{L}}}, \tag{17}$$

$$\operatorname{Log}_{\mathbf{x}}(\mathbf{y}) = \frac{\operatorname{arccosh}(-\langle \mathbf{x},\mathbf{y}\rangle_{\mathbb{L}})}{\sqrt{\langle \mathbf{x},\mathbf{y}\rangle_{\mathbb{L}}^2 - 1}}(\mathbf{y} + \langle \mathbf{x},\mathbf{y}\rangle_{\mathbb{L}}\mathbf{x}). \tag{18}$$

3.5. Stochastic Riemannian gradient descent

In Euclidean SOTS, when optimizing point sets for blue noise sampling, one can compute a descent direction of the SW energy for

each point by averaging each advection computed for a given number of slices (batch size L in Alg. 1), hence recovering a mini-batch stochastic gradient descent. On non-Euclidean domains, the advected positions cannot be simply averaged. We propose to use a stochastic Riemannian gradient descent (SRGD) approach combining the gradients obtained in each batch in the tangent plane of each sample [Bou23]. In standard SRGD this would be done by taking the average of the gradients

$$\mathbf{d}_i := \frac{1}{L}\sum_{l=1}^{L} \mathbf{d}_i^l, \tag{19}$$

but we instead use the geometric median, see 3.7. In our case, $\mathbf{d}_i^l := \mathrm{Log}_{\mathbf{x}_i^{(j)}}\left(\mathbf{g}(\mathbf{x}_i^{(j)})\right)$, where, following the notations of Alg. 1, $\mathbf{g}$ is the map that advects the point $\mathbf{x}_i^{(j)}$ in the θ *direction* following the 1D assignment obtained from the projection onto θ. Once the descent direction is computed for each sample, we advect the points using the Exp map by an, exponentially decaying, step size γ:

$$\mathbf{x}_i^{(j+1)} = \mathrm{Exp}_{\mathbf{x}_i^{(j)}}\left(\gamma\, \mathbf{d}_i\right). \tag{20}$$

Note that in the Euclidean setting, this boils down to the original SOTS algorithms [BRPP15] for blue noise sampling in $[0,1)^d$. As a first experiment, Figure 4 compares the blue noise characteristics of the uniform sampling of using NESOTS and classical point patterns on $\mathbb{S}^2$ [PSC*15].

3.6. Non-uniform densities

When dealing with continuous non-uniform measures ϕ using a sliced approach (*e.g.* importance sampling Monte Carlo rendering, image stippling), we would first need to have a closed-form formulation of the Radon transform of the target measure of ϕ along the slice θ, as discussed Paulin et al. [PBC*20] for the uniform measure in $[0,1)^d$. To overcome such issue, Salaün et al. [SGSS22] have used a binning strategy of the target points across n adaptive bins that follow the target distribution. We further simplify this approach on $\mathbb{S}^d$ and $\mathbb{H}^d$ using an empirical approximation of ϕ from a discrete measure ν with a large number of samples m (see Fig. 5). The key idea of Alg. 1 is to start from ν with $m \gg n$, and to uniformly pick n samples from ν at each slice (line 5). As long as $\nu \sim \phi$, this does not affect the minimization of the SW energy, while allowing a lot of flexibility with respect to the applications (see below) and keeping a balanced n-to-n 1d optimal transport problem to solve.

3.7. Geometric median

In our experiments, we observe that when targeting non-uniform measures, artifacts may appear during the gradient descent (*e.g.* alignment of samples as illustrated in Fig. 5-*c*). Some approaches handle this fact with a more robust advection computation, such as Salaün et al. [SGSS22] but they all require a non-negligible computational overhead, proportional to the input size (for example taking $m = kn$). To overcome this problem without adding limited extra computations, instead of taking the mean of the descent directions, we compute their geometric median. The idea arose from the analogy between the arbitrary bad batches that occurs with poor quality subsamples $\tilde{\nu}^l$ and malignant voters in voting systems, see [EMFGH23]. The geometric median can be computed very efficiently, in practice using the Weiszfeld algorithm [Wei37], see Appendix 8.

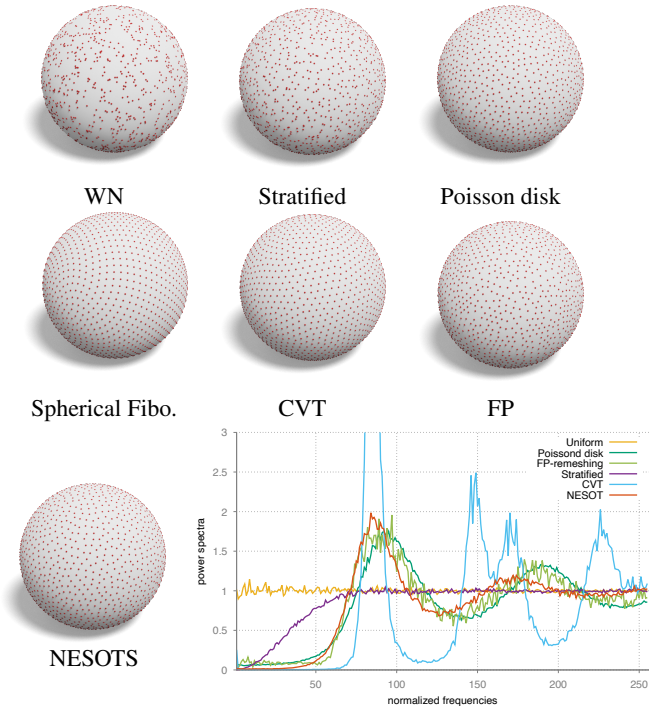


Figure 4: ***Blue noise on the sphere.*** *On $\mathbb{S}^2$, we evaluate the blue noise property of our sampling (2048 samples). Our result as to be compared to a uniform sampling, a stratified sampling using a healpix spherical structure [PSC*15], a Poisson disk sampling, a spherical Fibonacci sequence [KISS15], and a Lloyd's relaxation approach (Centroidal Voronoi Tesselation, CVT) [LWL*09b], and a geodesic farthest point greedy strategy [PC06] (FP) . The graph corresponds to the angular power spectra of the spherical harmonic transform of the point sets (except for spherical Fibonacci whose regular patterns make the spectral analysis less relevant) . As discussed in Pilleboue et al. [PSC*15], our sampler exhibits correct blue noise property with low energy for low frequencies, a peak at the average distance between samples and a plateau with few oscillations for higher frequencies.*

3.8. Real projective plane sampling

A slight modification of the NESOTS algorithm on the sphere allows sampling any density defined on the real projective plane $\mathbb{P}^d$ in the same blue noise way. Such sampling might have great use in graphics applications since many geometric objects are defined up to signs (such as directing vectors of lines or plane normals). Applications are detailed in section 6.

4. Intrinsic discrete manifold sampling

As a first application, we demonstrate the interest of the non-Euclidean sliced optimal transport approach for intrinsic sampling

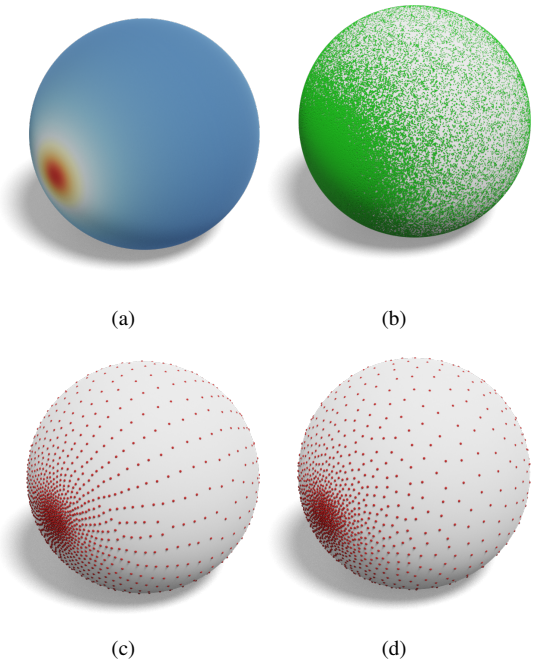


Figure 5: ***Non-uniform measure sampling:*** *given a non-uniform probability measure ϕ in $\mathbb{S}^2$ (a), we first construct a discrete measure $\nu \sim \phi$ with a large number of samples, 2048 samples here (b). Figures (c) and (d) are the output of the NESOTS algorithm for 2048 samples ($L = 32$, $K = 300$), when averaging the directions during the advection (c), or using the geometric median (d). While both distributions approximate the density, the latter provides a more stable result without sample alignment artifacts.*

of meshes in $\mathbb{R}^3$. Given a (closed) mesh $\mathcal{M}$, the core idea is to construct an injective map ψ from $\mathcal{M}$ to $\mathbb{S}^2$ or $\mathbb{H}^2$, to apply NESOTS on these domains to sample the image of the uniform measure $\mathcal{U}(\mathcal{M})$ on the mesh by ψ and to pull back the samples onto $\mathcal{M}$ with ψ^{-1}. Fig. 6 gives an illustration of this general pipeline.

For surfaces in $\mathbb{R}^d$, ψ can be built as a conformal map through the uniformization theorem [Abi81]. For short, any Riemannian surface of genus g admits a constant Gaussian curvature metric: spherical metric if $g = 0$ ($\mathbb{S}^{d-1}$, positive constant curvature space), a flat metric f $g = 1$ ($\mathbb{R}^{d-1}$, zero-curvature space) and an hyperbolic metric for $g \geq 2$ ($\mathbb{H}^{d-1}$, negative curvature space). In the discrete setting, $\mathcal{M}$ and $\mathcal{M}'$ are discrete conformal equivalent if the edge lengths l_{ij} and l'_{ij} are such that $l'_{ij} = \exp^{(u_i+u_j)/2} l_{ij}$, for some conformal factors $\{u_i\} \in \mathbb{R}$ on vertices [SSP08, BPS15, GLSW18, SCBK20]. In the following, we specifically target the $g = 0$ and $g \geq 2$ cases.

Note that in our pipeline, we do not explicitly require the map to be conformal. Any injective map between the mesh and the target space could be considered. We focus here on conformal maps as theoretical guarantees of existence and efficient algorithms to compute them exist. In Fig. 7, we illustrate that comparable blue noise sampling can be obtained non-conformal maps.

In the next section, we describe the sampling algorithm on the sphere, also illustrated in Fig. 6. Section 4.2 focuses on high genus surfaces using an iterated local hyperbolic embedding. Our samples minimize the sliced transport energy to the target measure with respect to the ground metric of the embedded space ($\mathbb{S}^d$ or $\mathbb{H}^d$), not the intrinsic metric of $\mathcal{M}$. Yet, from the regularity of the conformal maps we observe that blue noise characteristics are preserved when pulled back from the embedded space to $\mathcal{M}$ (see Sec. 4.3).

4.1. Global spherical embedding

The construction of the mapping ψ through the uniformization theorem depends on the genus g of $\mathcal{M}$. For the sake of simplicity, we start with the spherical case i.e., $g = 0$. By the uniformization theorem, a conformal map exists from $\mathcal{M}$ to $\mathbb{S}^2$. Here, we take advantage of the robust tools provided by Gillespie et al. [GSC21] to construct a bijective conformal map $\psi : \mathcal{M} \to \mathbb{S}^2$, allowing a global optimization.

Algorithm 2: Intrinsic Spherical blue noise surface sampling

Data: $\mathcal{M}$, ν, m, n, K, L and γ (see Alg. 1)
$\mathcal{M}_G = \text{BuildMapping}(\mathcal{M}, \mathbb{S}^2)$;
$\nu_G = \text{sampleMeshFaces}(\mathcal{M}_G, \nu, m)$;
$\tilde{\mu_G} = \text{SubSample}(\nu_G, n)$;
$\mu_G = \text{NESOTS}(\tilde{\mu_G}, \nu_G, K, L, \gamma)$; // Alg. 1
$\mu = \text{MapToMesh}(\mu_G, \mathcal{M}, \mathcal{M}_G)$; // Alg. 5
return μ

The global spherical sampling algorithm (Alg. 2) can thus be sketched as follows. For a mesh $\mathcal{M}$ homeomorphic to the sphere, we first construct ψ and the global mesh layout $\mathcal{M}_G$ on $\mathbb{S}^2$. We then construct the target density ν_G by uniformly sampling $\mathcal{M}$ with a large number of samples m (importance sampling of the triangles from the face areas), and projecting the samples onto $\mathcal{M}_G$. Note that ν_G is not uniform on the sphere since it captures the distortion induced by ψ. Finally, we use the NESOTS algorithm to compute the sliced optimal transport sampling μ_G and pullback this measure onto the input mesh as described in Sec. 4.3.

4.2. Local hyperbolic embedding

If $\mathcal{M}$ has higher genus, a conformal map exists from $\mathcal{M}$ to $\mathbb{H}^2$. Conformal coefficients can be obtained using the hyperbolic Discrete Yamabe Flow formulation [Luo04, BPS15]. Please refer to Section 4.3 for numerical details. The Yamabe flow allows us to compute the per vertex conformal factors $\{u_i\}$, and then the associated (hyperbilic) edge length l'_{ij} of the embedded mesh $\mathcal{M}_G$ onto $\mathbb{H}^2$. From the updated metric, one can embed $\mathcal{M}_G$ onto the hyperboloid of the Lorentz model (see Fig 6) using a greedy approach: starting from a initial vertex V_0 set to the origin $\mathbf{x}_O$, triangles are layed down onto $\mathbb{H}^2$ in a greedy breadth first strategy process following Schmidt et al.'s approach [SCBK20]. If we continue the BFS visiting the triangles several times, this process reveals that the mapping from $\mathcal{M}$ to $\mathbb{H}^2$ is periodic and the conformal map pave the entire hyperbolic plane. This prevents us from duplicating the global approach as described in Section 4.1 since the image of the

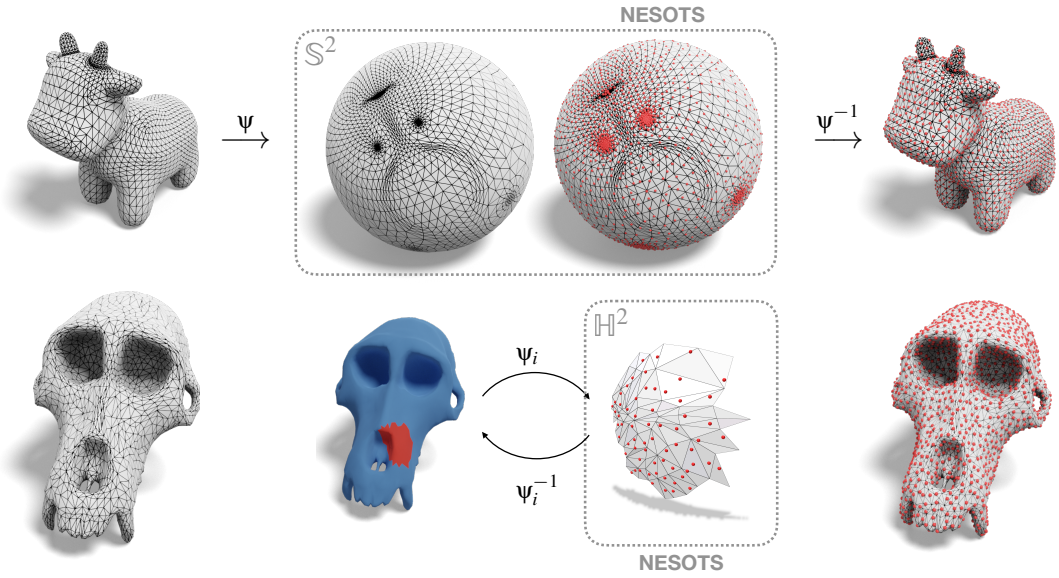


Figure 6: ***Overall pipeline of our intrinsic discrete manifold sampling****: starting from an input shape, we conformally embed the discrete structure onto either $\mathbb{S}^2$ for 0-genus surfaces, or local patches to $\mathbb{H}^2$ for higher genus one. Then, the NESOTS (Alg. 1) is used (globally or locally) to blue noise sample the embedded structure targeting a measure taking into account the metric distortion.*

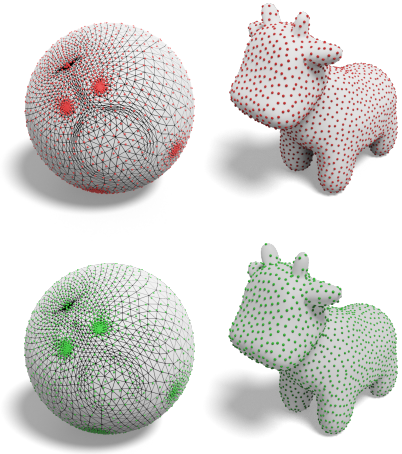


Figure 7: ***Sampling using a non-conformal spherical mapping:*** *first, we recall the NESOTS samplings using CEPS conformal maps (first row). In green, we have updated the mapping using some Laplacian smoothing steps on the sphere, resulting comparable sampling (second row).*

uniform measure $\mathcal{U}(\mathcal{M})$ by a periodic function is not integrable anymore, and hence the Optimal Transport framework cannot be used since it is only defined for probability measures.

To overcome this problem we restrict the embedding to patches of the mesh (see Fig. 6 and Alg. 3): starting from a global Yamabe Flow that is solved only once, we iterate over a local layout construction with an associated low distortion map ψ_i, and use NESOTS on this compact subset of $\mathbb{H}_2$. In this process, the choice of the first vertex of the layout matters since the distortion will be very low in a neighborhood of V_0 (mapped to $\mathbf{x}_O$), and will grow exponentially with the distance to it. Hence, using the embedding for $\mathbb{H}^2$ in $\mathbb{R}^3$, the main idea of the local algorithm is to construct a local layout until the (Euclidean) distance to the origin $\mathbf{x}_O$,in the z direction, exceeds a certain threshold ε. As we will ignore triangles far from the origin, we only build low distortion mappings. Note that the size of the patch for which the distortion is low depends on the quality of the mesh (triangle aspect), and on the curvature around V_0. The choice of ε allows controlling the scale of the optimization, giving a tradeoff between the sliced energy quality and speed (smaller patches leads to faster iterations). The effect of ε is evaluated in Fig. 9.

When a sample is displaced outside of the patch layout on $\mathbb{H}^2$, we just ignore the displacement (similarly to [PBC*20] when sampling $[0,1)^d$ or the d-Ball). To make sure that all the points are optimized as equally as possible, we just keep track of the number of times a given vertex $\mathcal{M}$ has been used as the origin v_0 of a patch and iterate on the local patch construction starting by the least embedded vertex (the priority queue in Alg. 3). Note that the local layout construction is extremely fast (linear complexity in the number of triangles of the patch).

In Fig. 8, we demonstrate the interest of the intrinsic sampling on high genus meshes. When the embedding is ambient-compatible (first row), we observe a slightly better sample distribution using our approach than FP and Poisson Disk sampling. In contrast, the CVT based approach produces a very high quality point pattern. Although, when the embedding is defective, our purely intrinsic approach led to an almost identical point pattern (in red) when mapped back to a better embedding (in blue) (b), whereas both

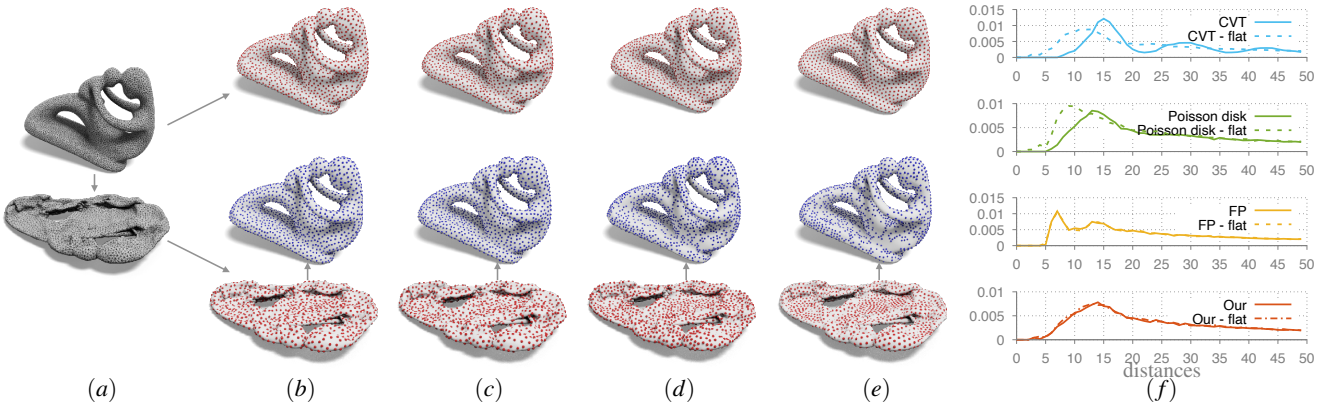


Figure 8: ***Intrinsic blue noise sampling of manifolds:*** *Given the* `fertility` *shape with two different Euclidean embeddings* (a)*. The flattened one is obtained through a physical simulation such that the two embeddings are intrinsically isometric. We illustrate the sampling of the meshes with red dots using our approach* (b)*, the intrinsic farthest point approach (FP) [PC06]* (c)*, the Poisson disk sampling in ambient space* (d)*, and the CVT sampling [LWL*09b]* (e)*. The blue dots correspond to the sampling on the* flat *embedding that are mapped to the* unflattened *one. First, we observe that our purely intrinsic approach leads to similar point sets in blue and red in* (c)*. Best point patterns are obtained using CVT when the embedding is correct in* $\mathbb{R}^3$*,* i.e. *no thin layers (*$(d)-$*top). However, for both Poisson disk and CVT, the sampling of the* flat *embedding leads to defective point patterns (holes in blue samples in* (d) *and* (e)*). In* (f)*, we present pair correlation functions for each sampler (both on the flat and top row meshes).*

Algorithm 3: Intrinsic local hyperbolic blue noise surface sampling

Data: $\mathcal{M}$, ν on $\mathcal{M}$, n, N , K, L, γ, $G = \mathbb{H}^2$ (see Alg. 1)

$\{u_i\} = \text{YamabeFlow}(\mathcal{M})$;
$\nu_G = \text{sampleMeshFaces}(\mathcal{M}_G, \nu, m)$;
for $i \in [[1, N]]$ **do**
  $vert = \text{PopVertexVisitPriorityQueue}()$;
  $(V_i, F_i) = \text{ComputeLocalHyperbolicLayout}(\{u_i\}, vert, \varepsilon)$;
  $\text{UpdateVertexVisitPriorityQueue}(V_i)$;
  $\mu_i = \text{ComputeRestrictionToLayout}(\mu, F_i)$;
  $\nu_i = \text{ComputeRestrictionToLayout}(\nu, F_i)$;
  $\mu_G = \text{NESOTS}(\mu_i, \nu_i, K, L, \gamma)$; `// Alg. 1`
  $\mu = \text{MapToMesh}(\mu_G, \mathcal{M}, \mathcal{M}_G)$; `// Alg. 5`
end
return μ

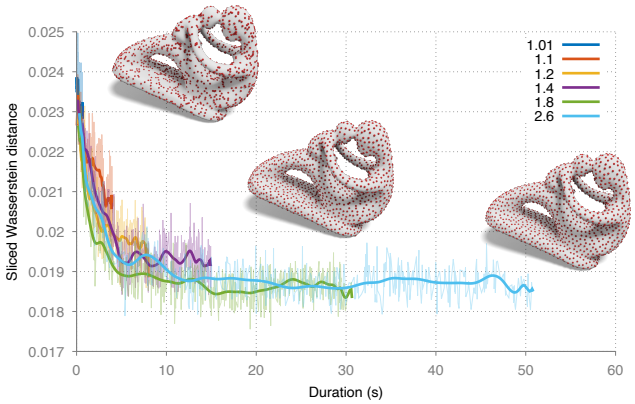
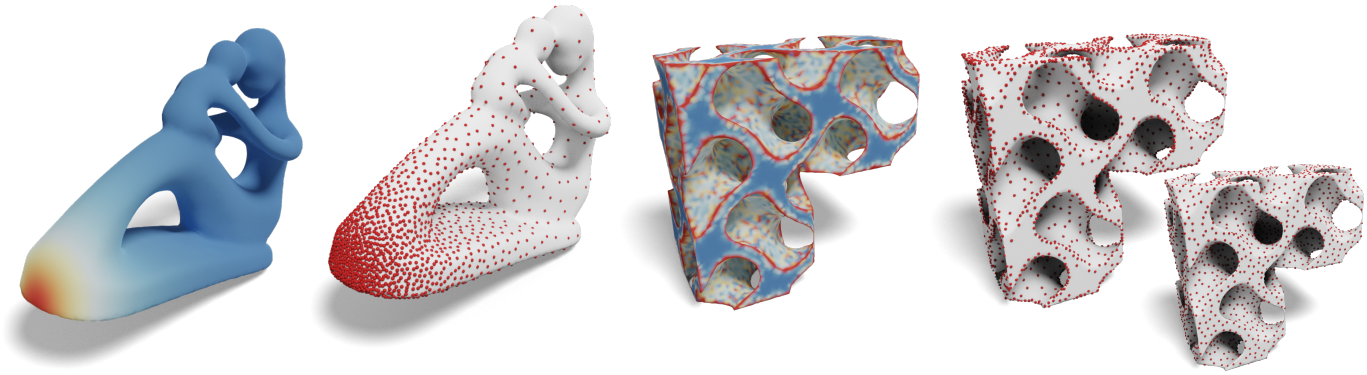


Figure 9: ***NESOTS convergence results:*** *we illustrate the convergence of Alg. 3 using* $N = 500$ *iterations (*$K = 500$*and* $L = 32$*) for 2048 samples, as a function of the* ε *parameter. If* ε *is too small, local patches are small which implies short timing but low quality blue noise point pattern (as quantified by the SW distance to the uniform measure). As* ε *increases, the blue noise quality is improved, but each iteration is longer. For* $\varepsilon \in \{1.01, 1.1, 1.2, 1.4, 1.8, 2.6\}$*, the average number of* μ_i *samples in each patch is respectively* $\{3.31, 14.76, 29.97, 61.86, 124.92, 242.82\}$*. Sampled meshes correspond to the final step of* $\varepsilon \in \{1.1, 1.4, 2.6\}$ *respectively.*

Poisson disk and CVT have critical voids and clusters due to bad assignments. To quantify this finding, we have computed the pair correlation function (pcf) [IPSS08] the exact geodesic distance on the manifold between each pair of samples [MMP87]. In Euclidean domains, pcf and radial mean power spectra capture similar point pattern characteristics [SÖA*19]. In Fig. 8-(f), we observe similar blue noise distribution (a peek at some characteristic distance and no too-close samples). We also observe that on the flat and non-flat meshes, our approach leads to similar pcfs. The pcfs CVT and Poisson disk are highly degraded on the flat geometries. In Fig. 10 we present sampling examples of non-uniform target measures on meshes. Additional sampling results are given in Fig. 13.



(a) (b) (c) (d)

Figure 10: ***Non-uniform target density examples on meshes****: given an input probability density function, a smooth one* (a) *on the* `fertility` *shape (genus*-4 *manifold, AIM@shape) and mean curvature driven one in* (c) *(gryroid surface, genus*-32 *manifold), our approach is able to generate blue noise samples* μ *approximating the density (2048 samples for* (b) *and 4096 samples for* (d)*). In* (d) *we also illustrate the sampling of the gyroid targeting the uniform density for comparison.*

4.3. Implementation details and complexity

First of all, for the hyperbolic case, discrete conformal coefficients $\{u_i\}$ are obtained by minimizing a convex energy, whose gradient and Hessian are given in [BPS15]. We thus apply a Newton descent approach with backtracking to ensure convergence. On the models presented in this paper, timings are detailed in Table 1. In the spherical case, we rely on the CEPS code provided by Gillespie et al. [GSC21] to explicitly construct the spherical embedding. Once obtained, Alg. 2 is a direct application of Alg. 1 with the same computational cost.

For the analysis of the local hyperbolic optimization (Alg. 3), we experimentally observe that the number of samples μ_i and ν_i on the layout grows linearly with ε. If C_ε denotes the average computation cost per slice and per patch, using a batch size L, K steps per patch and N global iterations, we obtain a $\mathcal{O}(N \cdot K \cdot L \cdot C_\varepsilon)$ complexity. Note that unless specified otherwise, we have used $N = 500, K = 10, \varepsilon = 1.5$ and $L = 32$ for all experiments. Although performances were not our primary concern, typical timings are given in Table 1. Please refer to Appendix 8 for a discussion on the computational cost overhead when using the geometric median instead of simply averaging directions in Alg. 1.

Once samples are optimized in, either globally for $\mathbb{S}^2$, or locally for $\mathbb{H}^2$, we need an efficient way to retrieve the face of the mesh a given sample falls in (and the barycentric coordinates of that sample in the face). For that purpose, we exploit the convexity of the domains: we first construct a BVH of the spherical or hyperbolic layout triangles and get the face id by shooting a ray through the origin $(0,0,0)$ and the sample (see Alg. 5 in Appendix A). Finally, in the hyperbolic case, to avoid having to map all the m points of $\tilde{\nu}$ on each layout, for each slice, we only map the n points that are subsampled from $\tilde{\nu}$. Source code is available at https://github.com/baptiste-genest/NESOTS.

Shape	Credits	$\lvert V\rvert$	$\lvert F\rvert$	g	Yamabe Flow	NESOTS
`spot`	[CPS13]	2930	5856	0	n.a.	17.48
`duck`	deriv. of K. Crane	29999	60006	3	10.67	27.73
`fertility`	AIM@Shape	8192	16396	4	3.02	15.13
`macaca`	[WAA*05]	3494	7000	4	1.36	11,12
`gyroid`	Thingi10k #111246	22115	44354	32	30.37	1.94

Table 1: ***Timings.*** *Mesh statistics and typical timings (in seconds) for the* $g \geq 2$ *shapes using the parameters presented in Sec. 4.3 (AMD Ryzen 5000-H, 16 cores).*

5. Real projective plane $\mathbb{P}^d$ sampling

Many objects generated by vectors are defined regardless of their length or sign. For instance, the orthographic projection of a 3d shape in the direction $\mathbf{d}$ is the same for all $\lambda\mathbf{d}, \forall \lambda \neq 0$. The space where collinear vectors are identified is called the Projective Plane $\mathbb{P}^d$. One idea might be to project the points on the sphere, which will successfully identify the vectors equivalent up to a positive scale $\lambda > 0$ but not up to a sign. Hence, trying to generate a "uniform" set of lines with any blue noise sampler on the sphere does not output satisfactory results as the points are not optimized to take into account this equivalence relationship. A simple modification of Alg. 1 described in Alg. 4, allows us to successfully extend the blue noise generation of points, in any dimension on $\mathbb{P}^d$ following any density on the sphere satisfying $f(\mathbf{x}) = f(-\mathbf{x})$ for $\mathbf{x} \in \mathbb{S}^d$. To the best of our knowledge, this is new.

Lines and Hyperplanes sampling. As already stated, lines, characterized by their unit vector, can be generated uniformly on $\mathbb{P}^d$ using Alg. 4 (see Fig. 5 for a 3d blue noise line sampling in $\mathbb{P}^2$). By taking the orthogonal complement of such lines, we can similarly obtain a blue noise sampling of $(d-1)-$hyperplanes.

Affine line and hyperplane sampling. Note than even *affine* spaces can be sampled by Alg. 4. For instance, an affine line can

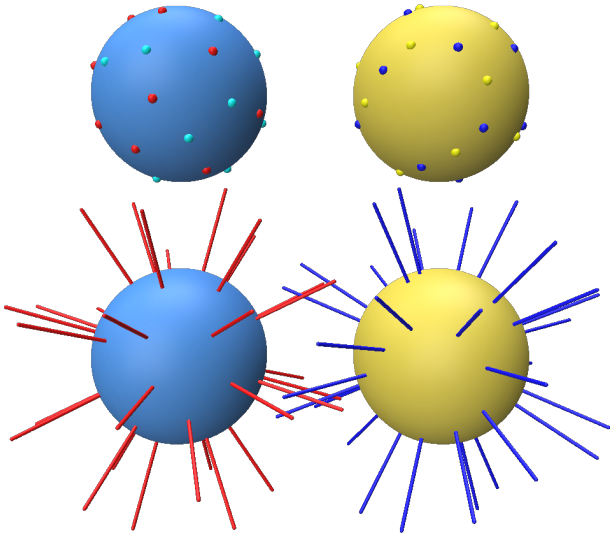


Figure 11: ***Projective plane*** $\mathbb{P}^2$ ***sampling:*** *red points are sampled with Alg. 4, light blue points are the opposites of the red ones. Similarly, blue and yellow points are given by a spherical Fibonacci [KISS15]. Points obtained by Alg. 4 have better blue noise characteristics when considered with their opposites. To illustrate its use, we display at the bottom row the lines generated by the points.*

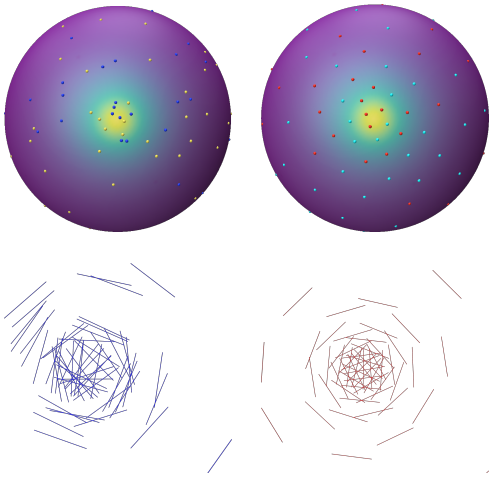


Figure 12: ***Affine lines sampling:*** *from the mapping of lines coefficients to* $\mathbb{P}^2$*, we generate 64 blue noise affine lines following a non-uniform density (top row) using either a white noise sampling (left column) or Alg. 4. When mapped back to* $\mathbb{R}^2$*, our sampling exhibits blue noise characteristics in* $\mathbb{R}^2$ *with respect to the metric induced by the Cartesian mapping (second row). Note that here only segments are displayed for the sake of clarity but they are actual lines of* $\mathbb{R}^2$*.*

described by its Cartesian equation, i.e. in dimension 2

$$a\mathbf{x} + b\mathbf{y} + c = 0, \tag{21}$$

but notice that, $\forall k \neq 0$, if $\mathbf{x}$ and $\mathbf{y}$ are solutions of (21), then $ka\mathbf{x} + kb\mathbf{y} + kc = 0$. Hence each affine space of dimension d can be represented in the projective plane $\mathbb{P}^d$ by its Cartesian coefficients (here $(a,b,c)^t$). See Fig. 12 for a 2d affine line sampling experiment.

Rotation Sampling by Unit Quaternion sampling. A unit quaternion q can act on a vector as a rotation

$$\mathbf{x} \mapsto \mathbf{q}^{-1}\tilde{\mathbf{x}}\mathbf{q},$$

where $\tilde{\mathbf{x}}$ is the imaginary quaternion with $\mathbf{x}$ as vector part. Since $\mathbf{q}$ appears twice in the product, $\mathbf{q}$ and $-\mathbf{q}$ gives the same rotation. Hence one can use Alg. 4 on $\mathbb{P}^3$ to uniformize a set of unit quaternions (represented as unit 4 dimensional unit vectors). Previous approaches such as Alexa's technique [Ale22], provides good sampling on the 3-Sphere but does not directly tackle the sign equivalence problem, which leads to imperfect rotation sampling . The results of the rotation sampling process is displayed in Fig. 1-*(right)* where each shape is rotated by a rotation generated by Alg. 4.

6. Limitations and future Work

Our approach extends the blue noise sampling of any probability measure through the sliced optimal transport energy, originally designed for Euclidean domains, to Riemannian manifolds: the spherical space $\mathbb{S}^d$, the hyperbolic space $\mathbb{H}^d$, and the projective one $\mathbb{P}^d$. In a nutshell, from explicit advection and direction averaging steps

Algorithm 4: Real Projective Plane Sampling $\mathbb{P}^d$

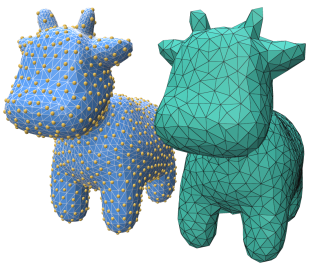
Data: $\nu = \sum_{i=1}^m \delta_{\mathbf{y}_i}$, K, L, and γ (see Alg. 1).
Result: $\mu^{(K)}$

$\mu^{(0)} = \text{subSample}(\nu, 2n)$
for $j \in [[1,K]]$ **do**
  parallel for $l \in [[1,L]]$ **do** // Batch
    $\tilde{\nu} = \text{subSample}(\tilde{\nu}, 2n)$; // Sec. 3.6
    $\theta = \text{RandomSlice}()$; // Sec. 3.1
    $\tilde{\nu}_\theta = P^\theta\left(\tilde{\nu}^l\right)$; // Sec. 3.1
    $\mu_\theta = P^\theta\left(\mu^{(j)}\right) \cup -P^\theta\left(\mu^{(j)}\right)$; // Sec. 3.1
    $T = \text{Solve1DOT}(\mu_\theta, \tilde{\nu}_\theta)$; // Sec. 3.2
    for $i \in [[1,2n]]$ **do**
      $\mathbf{g} = \Gamma_\theta\left(P^\theta\left(\mathbf{x}_i^{(j)}\right), T\left(P^\theta\left(\mathbf{x}_i^{(j)}\right)\right)\right)$; // Sec. 3.3
      $\mathbf{d}_i^l = \text{Log}_{\mathbf{x}_i^{(j)}}\left(\mathbf{g}\left(\mathbf{x}_i^{(j)}\right)\right)$; // Sec. 3.4
    end
  end
  parallel for $i \in [[1,n]]$ **do**
    $\mathbf{d}_i = \text{GeoMed}\left(\{\mathbf{d}_i^l\}_L \cup \{-\mathbf{d}_{i+n}^l\}_L\right)$; // Sec. 3.7
    $\mathbf{x}_i^{(j+1)} = \text{Exp}_{\mathbf{x}_i^{(j)}}(\gamma \mathbf{d}_i)$; // Sec. 3.5
  end
end
return $\mu^{(K)} = \sum_{i=1}^m \delta_{\mathbf{x}_i^{(K)}}$

on these spaces, we present a gradient descent strategy that optimizes a point set minimizing the sliced Wasserstein energy.

First of all, concerning the generic NESOTS approach, there are many opportunities for performance improvements. We are convinced that many variance reduction techniques could be borrowed from Monte Carlo rendering approach to speed up the sliced strategy (*e.g.* importance sampling of the θ directions, control variates using a proxy for the SW energy).

Thanks to the uniformization theorem, we demonstrated the interest of the approach for intrinsic blue noise sampling of discrete surfaces. Although we may not compete with existing extremely fast restricted Voronoi based techniques when the mesh has a good embedding, we advocate that the purely intrinsic nature of our construction is of interest. An important limitation is the robustness of the global conformal map in the spherical case that may impact the sampling when high distortion occurs. In the hyperbolic case, our local construction mitigates this by controlling potential distortion issues (the ε parameter) but we are convinced that improvements exist, *e.g.* using implicit intrinsic remeshing as in Gillespie et al. [GSC21]. On the geometric side, we only focused $g = 0$ and $g \geq 2$ surfaces leaving the flat metric space case aside. For $g = 1$, cut-and-open strategies must be designed that we avoid in spherical and hyperbolic domains. In this paper, we also focus on the sample generation, leaving the use cases of the point set as future work (*e.g.* decal placement, function reconstruction, remeshing). For remeshing, the convexity of the $\mathbb{S}^2$ and $\mathbb{H}^2$ could be further exploited to reconstruct a mesh: on the $\mathbb{S}^2$ the convex hull of the optimized samples leads to a trivial (manifold) surface reconstruction (see inset). The hyperbolic case is more complicated as holes could be embedded in a compact subset of $\mathbb{H}^2$ for which the global convex hull topology does not make sense. We believe that a local combinatorial construction from the convex hull using a small ε could be investigated.



Finally, we have only scratched the use of blue noise sampling in the projective space $\mathbb{P}^d$ for computer graphics applications. For instance, Monte Carlo-like line and segment sample estimators may lead to drastic reductions of variance in rendering for some effects such as soft shadows or defocusing blur [SMJ17]. We believe that affine line sampling approaches as illustrated in Fig. 12 would be of great interest in this context.

Acknowledgments

This research was partially funded by the projects StableProxies (ANR-22-CE46-0006) and OTTOPIA (ANR-20-CHIA-0030) of the French National Research Agency (ANR), and gifts from Adobe Inc.

Appendix A: Additional algorithms

The objective of Alg. 5 is to find the face a point is lying on, and to compute the correspondence between its position on the face embedded in $\mathbb{R}^3$ and on the layout in $\mathbb{S}^2$ (resp $\mathbb{H}^2$) through barycentric coordinates. Even if we theoretically should use spherical (resp. hyperbolic) barycentric coordinates, we observe that the Euclidean barycentric coordinates make a good enough quality proxy while avoiding computing transcendental functions at each mapping. For high performances, the face retrieval can be done leveraging the convexity of $\mathbb{S}^2$ and $\mathbb{H}^2$ through a ray shooting approach (rays starting from the domain origin to the sample to locate), with a BVH of the faces. In our implementation, we used the library [PG23]. In

Algorithm 5: Mapping measures between two meshes

Data: μ_G ,$\mathcal{M}$ and $\mathcal{M}_G$
BVH $=$ BuildBVH($\mathcal{M}_G$) ;
for $i \in [[1,n]]$ **do**
  $\tilde{F} =$ BVH.intersect($\mathcal{M}_G, \mathbf{x}_O, \mathbf{x}_i^G$) ;
  $b_i =$ ComputeBarycentricCoordinates($M_G, \mathbf{x}_i^G, \tilde{F}$) ;
  $F =$ FindCorrespondingFace($\tilde{F}, \mathcal{M}$) ;
  $\mathbf{x}_i =$ PositionFromBarycentricCoordinates($\mathcal{M}, b_i, F$) ;
end
return μ

Alg. 6, we detail the Weiszfeld's algorithm we use for the geometric median computation using an iterative least squares approach. Note that, as stated in Section 3.7, Weiszfeld's algorithm is used to combine the gradients (in $\mathbb{R}^n$) during the Riemannian stochastic gradient descent. Theoretically, without the τ term, this algorithm

Algorithm 6: Weiszfeld's geometric median algorithm [Wei37]

Data: The samples $\{\mathbf{x}_i\}_L \in \mathbb{R}^d$, a stability parameter $\tau \in \mathbb{R}$
$\mathbf{y} = 0$;
$j = 2\tau$;
while $j > \tau$ **do**
  $d = 0$;
  $w = 0$;
  $\tilde{\mathbf{y}} = 0$;
  for $i \in [[0,L]]$ **do**
    $d = \tau + \|\mathbf{y} - \mathbf{x}_i\|_2$;
    $w \mathrel{+}= d$;
    $\tilde{\mathbf{y}} \mathrel{+}= \frac{\mathbf{x}_i}{d}$;
  end
  $\tilde{\mathbf{y}} \mathrel{/}= w$;
  $j = \|\mathbf{y} - next\|_2$;
  $\mathbf{y} = \tilde{\mathbf{y}}$;
end
return y

does not converge if $\mathbf{y}_0 = \mathbf{x}_i$ for some i. In practice, with $\tau > 0$, we do not observe convergence issues (interested readers may refer to Cohen et al. [CLM*16] for a review of standard algorithms). While geometric median is an essential element to guarantee quality of the result for highly non-uniform density functions, a slight computational overhead exists when compared to the geometric mean. On the `fertility` mesh with standard parameters (see Sec 4.3), the optimization part of the Alg. 3 takes 12.38s with the mean and 13.33s with the geometric median ($L = 32$ and $\tau = 10^{-7}$ for all experiments).

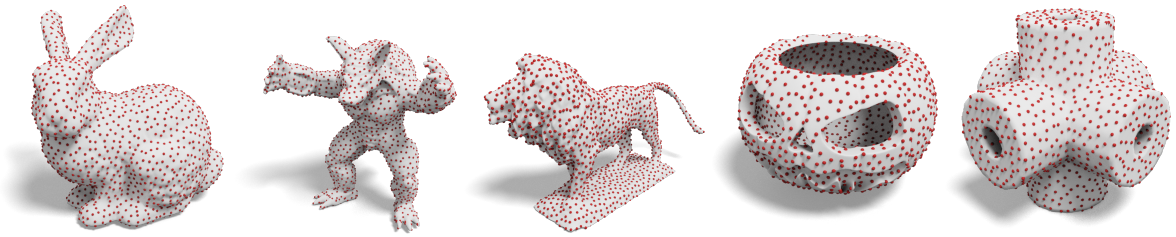


Figure 13: ***Intrinsic discrete manifold sampling:*** *additional sampling results with 2048 samples for $g = 0$ and $g \geq 2$ surfaces.*

References

[Abi81] ABIKOFF W.: The uniformization theorem. *The American Mathematical Monthly 88*, 8 (1981), 574–592. 7

[ACB17] ARJOVSKY M., CHINTALA S., BOTTOU L.: Wasserstein generative adversarial networks. In *International conference on machine learning* (2017), PMLR, pp. 214–223. 2

[ADVA21] ALIMISIS F., DAVIES P., VANDEREYCKEN B., ALISTARH D.: Distributed principal component analysis with limited communication. *Advances in Neural Information Processing Systems 34* (2021), 2823–2834. 5

[AGY*17] AHMED A., GUO J., YAN D.-M., FRANCESCHIA J.-Y., ZHANG X., DEUSSEN O.: A simple push-pull algorithm for blue-noise sampling. *Transactions on Visualization and Computer Graphics 23*, 12 (Dec. 2017), 2496–2508. 3

[Ale22] ALEXA M.: Super-fibonacci spirals: Fast, low-discrepancy sampling of so (3). In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 8291–8300. 11

[BBC*22] BONET C., BERG P., COURTY N., SEPTIER F., DRUMETZ L., PHAM M.-T.: Spherical sliced-wasserstein. *arXiv preprint arXiv:2206.08780* (2022). 2, 3

[BC19] BONNEEL N., COEURJOLLY D.: SPOT: Sliced Partial Optimal Transport. *ACM Trans. Graph. 38*, 4 (July 2019). 2

[BCDC22] BONET C., CHAPEL L., DRUMETZ L., COURTY N.: Hyperbolic sliced-wasserstein via geodesic and horospherical projections. *arXiv preprint arXiv:2211.10066* (2022). 2, 3, 4

[BCK18] BADEN A., CRANE K., KAZHDAN M.: Möbius registration. 211–220. 2

[BD23] BONNEEL N., DIGNE J.: A survey of optimal transport for computer graphics and computer vision. 439–460. 2

[Bon13] BONNOTTE N.: *Unidimensional and evolution methods for optimal transportation*. PhD thesis, Université Paris Sud-Paris XI; Scuola normale superiore (Pise, Italie), 2013. 2

[Bou23] BOUMAL N.: *An introduction to optimization on smooth manifolds*. Cambridge University Press, 2023. 6

[BPS15] BOBENKO A. I., PINKALL U., SPRINGBORN B. A.: Discrete conformal maps and ideal hyperbolic polyhedra. *Geometry & Topology 19*, 4 (2015), 2155–2215. 7, 10

[Bri07] BRIDSON R.: Fast poisson disk sampling in arbitrary dimensions. *SIGGRAPH sketches 10*, 1 (2007), 1. 3

[BRPP15] BONNEEL N., RABIN J., PEYRÉ G., PFISTER H.: Sliced and radon wasserstein barycenters of measures. *Journal of Mathematical Imaging and Vision 51* (2015), 22–45. 2, 6

[BSD09] BALZER M., SCHLÖMER T., DEUSSEN O.: Capacity-constrained point distributions: A variant of Lloyd's method. *ACM Trans. Graph. 28*, 3 (2009), 86:1–8. 3

[BWWM10] BOWERS J., WANG R., WEI L. Y., MALETZ D.: Parallel poisson disk sampling with spectrum analysis on surfaces. *ACM Transactions on Graphics 29* (2010). 3

[CFTR16] COURTY N., FLAMARY R., TUIA D., RAKOTOMAMONJY A.: Optimal transport for domain adaptation. *IEEE transactions on pattern analysis and machine intelligence 39*, 9 (2016), 1853–1865. 2

[CJW*09] CLINE D., JESCHKE S., WHITE K., RAZDAN A., WONKA P.: Dart throwing on surfaces. In *Computer Graphics Forum* (2009), vol. 28, Wiley Online Library, pp. 1217–1226. 3

[CLM*16] COHEN M. B., LEE Y. T., MILLER G., PACHOCKI J., SIDFORD A.: Geometric median in nearly linear time. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing* (2016), pp. 9–21. 12

[CM98] CABRELLI C. A., MOLTER U. M.: A linear time algorithm for a matching problem on the circle. *Information processing letters 66*, 3 (1998), 161–164. 4

[CPS13] CRANE K., PINKALL U., SCHRÖDER P.: Robust fairing via conformal curvature flow. *ACM Transactions on Graphics (TOG) 32*, 4 (2013), 1–10. 2, 10

[DGBOD12] DE GOES F., BREEDEN K., OSTROMOUKHOV V., DESBRUN M.: Blue noise through optimal transport. *ACM Trans. Graph. 31*, 6 (2012), 171. 2

[DRG09] DELON J., RABIN J., GOUSSEAU Y.: Transportation distances on the circle and applications. *arXiv preprint arXiv:0906.5499* (2009). 4

[DWGJ21] DAI J., WU Y., GAO Z., JIA Y.: A hyperbolic-to-hyperbolic graph convolutional network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), pp. 154–163. 5

[EMFGH23] EL-MHAMDI E.-M., FARHADKHANI S., GUERRAOUI R., HOANG L.-N.: On the strategyproofness of the geometric median. In *International Conference on Artificial Intelligence and Statistics* (2023), PMLR, pp. 2603–2640. 6

[FCG*21] FLAMARY R., COURTY N., GRAMFORT A., ALAYA M. Z., BOISBUNON A., CHAMBON S., CHAPEL L., CORENFLOS A., FATRAS K., FOURNIER N., GAUTHERON L., GAYRAUD N. T., JANATI H., RAKOTOMAMONJY A., REDKO I., ROLET A., SCHUTZ A., SEGUY V., SUTHERLAND D. J., TAVENARD R., TONG A., VAYER T.: Pot: Python optimal transport. *Journal of Machine Learning Research 22*, 78 (2021), 1–8. 2

[GGS03] GOTSMAN C., GU X., SHEFFER A.: Fundamentals of spherical parameterization for 3d meshes. In *ACM SIGGRAPH 2003 Papers*. 2003, pp. 358–363. 2

[GLSW18] GU X. D., LUO F., SUN J., WU T.: A discrete uniformization theorem for polyhedral surfaces. *Journal of differential geometry 109*, 2 (2018), 223–256. 7

[GSC21] GILLESPIE M., SPRINGBORN B., CRANE K.: Discrete conformal equivalence of polyhedral surfaces. *ACM Trans. Graph. 40*, 4 (2021). 2, 7, 10, 12

[GY03] GU X., YAU S.-T.: Global conformal surface parameterization. In *Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing* (2003), pp. 127–137. 2

[GYJZ15] GUO J., YAN D.-M., JIA X., ZHANG X.: Efficient maximal

poisson-disk sampling and remeshing on surfaces. *Computers & Graphics 46* (2015), 72–79. 3

[HAT*00] HAKER S., ANGENENT S., TANNENBAUM A., KIKINIS R., SAPIRO G., HALLE M.: Conformal surface parameterization for texture mapping. *IEEE Transactions on Visualization and Computer Graphics 6*, 2 (2000), 181–189. 2

[HGK*16] HUANG G., GUO C., KUSNER M. J., SUN Y., SHA F., WEINBERGER K. Q.: Supervised word mover's distance. *Advances in neural information processing systems 29* (2016). 2

[IPSS08] ILLIAN J., PENTTINEN A., STOYAN H., STOYAN D.: *Statistical analysis and modelling of spatial point patterns*. John Wiley & Sons, 2008. 9

[JZW*15] JIANG M., ZHOU Y., WANG R., SOUTHERN R., ZHANG J. J.: Blue noise sampling using an sph-based method. *Transactions on Graphics 34*, 6 (2015), 1–11. 3

[KISS15] KEINERT B., INNMANN M., SÄNGER M., STAMMINGER M.: Spherical fibonacci mapping. *ACM Transactions on Graphics (TOG) 34*, 6 (2015), 1–7. 6, 11

[KSS06] KHAREVYCH L., SPRINGBORN B., SCHRÖDER P.: Discrete conformal mappings via circle patterns. *ACM Transactions on Graphics (TOG) 25*, 2 (2006), 412–438. 2

[Lee06] LEE J. M.: *Riemannian manifolds: an introduction to curvature*, vol. 176. Springer Science & Business Media, 2006. 4, 5

[Luo04] LUO F.: Combinatorial yamabe flow on surfaces. *Communications in Contemporary Mathematics 6*, 05 (2004), 765–780. 7

[LWL*09a] LIU Y., WANG W., LÉVY B., SUN F., YAN D.-M., LU L., YANG C.: On centroidal voronoi tessellation, energy smoothness and fast computation. *Transactions on Graphics 28*, 4 (08 2009), 1–17. 3

[LWL*09b] LIU Y., WANG W., LÉVY B., SUN F., YAN D.-M., LU L., YANG C.: On centroidal voronoi tessellation—energy smoothness and fast computation. *ACM Transactions on Graphics (ToG) 28*, 4 (2009), 1–17. 3, 6, 9

[MMP87] MITCHELL J. S., MOUNT D. M., PAPADIMITRIOU C. H.: The discrete geodesic problem. *SIAM Journal on Computing 16*, 4 (1987), 647–668. 9

[NDC*20] NADJAHI K., DURMUS A., CHIZAT L., KOLOURI S., SHAHRAMPOUR S., SIMSEKLI U.: Statistical and topological properties of sliced probability divergences. *Advances in Neural Information Processing Systems 33* (2020), 20802–20812. 2

[NG18] NADER G., GUENNEBAUD G.: Instant transport maps on 2d grids. *ACM Trans. Graph. 37*, 6 (2018), 13. 2

[ÖAG10] ÖZTIRELI C., ALEXA M., GROSS M.: Spectral sampling of manifolds. *ACM Transactions on Graphics (TOG) 29*, 6 (2010), 1–8. 3

[PBC*20] PAULIN L., BONNEEL N., COEURJOLLY D., IEHL J.-C., WEBANCK A., DESBRUN M., OSTROMOUKHOV V.: Sliced optimal transport sampling. *ACM Trans. Graph. 39*, 4 (2020), 99. 2, 3, 6, 8

[PC06] PEYRÉ G., COHEN L. D.: Geodesic remeshing using front propagation. *International Journal of Computer Vision 69* (2006), 145–156. 3, 6, 9

[PC*19] PEYRÉ G., CUTURI M., ET AL.: Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning 11*, 5-6 (2019), 355–607. 2

[PG23] PÉRARD-GAYOT A.: BVH construction and traversal library, 2023. URL: https://github.com/madmann91/bvh. 12

[PKD05] PITIÉ F., KOKARAM A. C., DAHYOT R.: N-dimensional probablility density function transfer and its application to colour transfer. In *IEEE Int. Conf. on Computer Vision (ICCV)* (2005). 2

[PSC*15] PILLEBOUE A., SINGH G., COEURJOLLY D., KAZHDAN M., OSTROMOUKHOV V.: Variance analysis for monte carlo integration. *ACM Trans. Graph. (Proc. SIGGRAPH) 34*, 4 (2015), 124:1–124:14. 6

[QCHC17] QIN H., CHEN Y., HE J., CHEN B.: Wasserstein blue noise sampling. *ACM Transactions on Graphics (TOG) 36*, 5 (2017), 1–13. 2, 3

[RPDB11] RABIN J., PEYRÉ G., DELON J., BERNOT M.: Wasserstein barycenter and its application to texture mixing. In *International Conference on Scale Space and Variational Methods in Computer Vision* (2011), Springer, pp. 435–446. 2

[SCBK20] SCHMIDT P., CAMPEN M., BORN J., KOBBELT L.: Inter-surface maps via constant-curvature metrics. *ACM Transactions on Graphics (TOG) 39*, 4 (2020), 119–1. 2, 7

[SdGP*15] SOLOMON J., DE GOES F., PEYRÉ G., CUTURI M., BUTSCHER A., NGUYEN A., DU T., GUIBAS L.: Convolutional wasserstein distances: Efficient optimal transportation on geometric domains. *ACM Trans. Graph. 34*, 4 (2015), Art. 66. 2

[SGSS22] SALAÜN C., GEORGIEV I., SEIDEL H.-P., SINGH G.: Scalable multi-class sampling via filtered sliced optimal transport. *arXiv preprint arXiv:2211.04314* (2022). 2, 6

[SMJ17] SINGH G., MILLER B., JAROSZ W.: Variance and convergence analysis of Monte Carlo line and segment sampling. *Computer Graphics Forum (Proceedings of EGSR) 36*, 4 (June 2017). 12

[SÖA*19] SINGH G., ÖZTIRELI C., AHMED A. G., COEURJOLLY D., SUBR K., DEUSSEN O., OSTROMOUKHOV V., RAMAMOORTHI R., JAROSZ W.: Analysis of sample correlations for monte carlo rendering. 473–491. 2, 9

[SRGB14] SOLOMON J., RUSTAMOV R., GUIBAS L., BUTSCHER A.: Earth mover's distances on discrete surfaces. *ACM Trans. Graph. 33*, 4 (2014), Art. 67. 2

[SSCO08] SHAPIRA L., SHAMIR A., COHEN-OR D.: Consistent mesh partitioning and skeletonisation using the shape diameter function. *The Visual Computer 24* (2008), 249–259. 3

[SSP08] SPRINGBORN B., SCHRÖDER P., PINKALL U.: Conformal equivalence of triangle meshes. In *ACM SIGGRAPH 2008 papers*. 2008, pp. 1–11. 7

[TMN*00] TANAKA S., MORISAKI A., NAKATA S., FUKUDA Y., YAMAMOTO H.: Sampling implicit surfaces based on stochastic differential equations with converging constraint. *Computers & Graphics 24*, 3 (2000), 419–431. 3

[V*09] VILLANI C., ET AL.: *Optimal transport: old and new*, vol. 338. Springer, 2009. 1

[WAA*05] WILEY D. F., AMENTA N., ALCANTARA D. A., GHOSH D., KIL Y. J., DELSON E., HARCOURT-SMITH W., ROHLF F. J., ST JOHN K., HAMANN B.: *Evolutionary morphing*. IEEE, 2005. 10

[Wei37] WEISZFELD E.: Sur le point pour lequel la somme des distances de n points donnés est minimum. *Tohoku Mathematical Journal, First Series 43* (1937), 355–386. 6, 12

[XHGL12] XU Y., HU R., GOTSMAN C., LIU L.: Blue noise sampling of surfaces. *Computers & Graphics 36*, 4 (2012), 232–240. 3

[XLC*16] XIN S.-Q., LÉVY B., CHEN Z., CHU L., YU Y., TU C., WANG W.: Centroidal power diagrams with capacity constraints: Computation, applications, and extension. *Transactions on Graphics 35*, 6 (2016), 1–12. 3

[Yuk15] YUKSEL C.: Sample elimination for generating poisson disk sample sets. In *Computer Graphics Forum* (2015), vol. 34, Wiley Online Library, pp. 25–32. 3

[ZGW*13] ZHONG Z., GUO X., WANG W., LÉVY B., SUN F., LIU Y., MAO W., ET AL.: Particle-based anisotropic surface meshing. *Transactions on Graphics 32*, 4 (2013), 99–1. 3