International Journal of Quantum Technologies

PRIME OPEN ACCESS

**Volume 1, Issue 2**

**Research Article**

**Date of Submission:** 16 June, 2025
**Date of Acceptance:** 22 July, 2025
**Date of Publication:** 11 August, 2025

# Mesh Denoising

**Constantin Vaillant-Tenzer[1,2,3]***

[1]Ecole Normale Supérieure - PSL: Department of cognitive studies

[2]ENS Paris Saclay - DER of mathematics

[3]MBZAI-IFM, Paris Lab

***Corresponding Author:**
Constantin Vaillant-Tenzer, Ecole Normale Supérieure - PSL: Department of cognitive studies: ENS Paris Saclay - DER of mathematics: MBZAI-IFM, Paris Lab, 45 rue d'Ulm, 75005 Paris.

**Citation:** Tenzer, C. V. (2025). Mesh Denoising. *Int J Quantum Technol, 1*(2), 01-09.

## Abstract

In this paper, we study four mesh denoising methods: linear filtering, a heat diffusion method, Sobolev regularization, and, to a lesser extent, the Sinkhorn algorithm. We show that, for an image, the use of the Gibbs kernel is counterproductive. We demonstrate that while Sobolev regularization is the fastest method, it produces the least faithful denoised mesh compared to the original. We empirically show that for large meshes, the heat diffusion method is slower and less effective than filtering, which is not the case for small meshes. Finally, we observe that the first three methods perform significantly better with high-dimensional meshes.

## Introduction

### The Problem

To represent three-dimensional objects within a computer, meshes are highly practical. A mesh $\mathcal{M}$ consists of an index set $\mathcal{V} = [[1, n]]$, which indexes $n$ vertices, a set of edges $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$, and a set of faces $\mathcal{F} \subset \mathcal{V} \times \mathcal{V} \times \mathcal{V}$. Although these objects are not widely used today, they could become more prevalent, for instance, when transmitting 3D objects via holograms through a digital channel. Since information can be corrupted or poorly transmitted, denoising is necessary. For our study, we start with the noisy meshes shown in Figure 1.1.

However, this task is not straightforward. First, these objects have a high dimension n, especially when they aim to faithfully represent a three-dimensional real-world object. Thus, an algorithm must be sufficiently fast. It may also be necessary for a machine to determine whether a received mesh needs denoising and to measure the noise without knowing the original. These are machine learning problems that we will not address here. What are the possible methods that can be used for supervised learning?

The most promising method, though particularly difficult to implement, would involve efficient optimal transport on geometric domains using the Sinkhorn algorithm [1]. However, other faster methods, such as Sobolev regularization, or simpler ones, like a high-pass filter, can yield good results. But what are their limitations? We will see that they perform much less effectively on small meshes. Nevertheless, they can still make incomprehensible figures intelligible.

### Contributions

After reviewing the theoretical foundations of the Sinkhorn algorithm, namely optimal transport theory and Kullback-Leibler divergence, and presenting the heat equation and Cholesky factorization, we attempt to apply this algorithm to denoise an image. We then focus on three mesh denoising methods: a low-pass filter, a heat diffusion method, and Sobolev regularization. We compare their reliability and execution speed on two meshes of very different sizes ($n$ = 24,955 and $n$ = 299 vertices). Finally, we outline the progress still needed for mesh denoising.

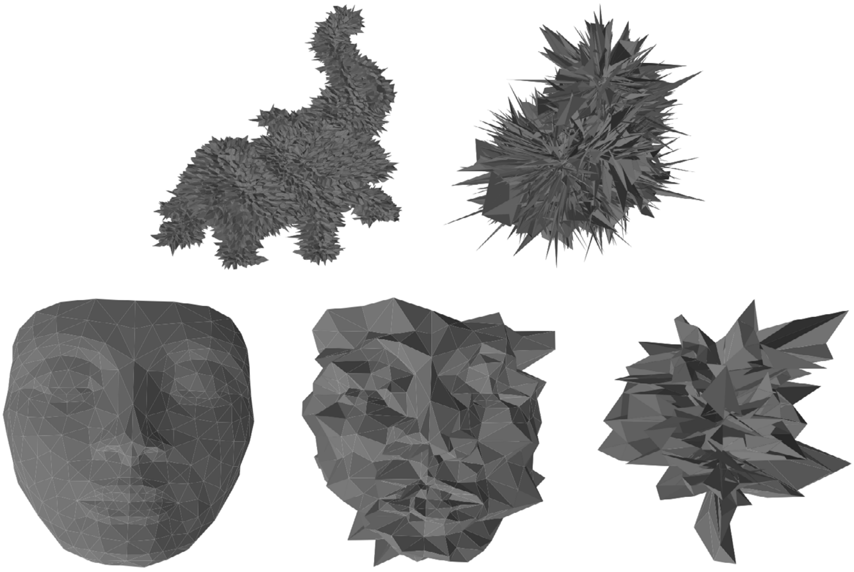

**Figure 1: The Various Noisy Meshes Studied in Table 4.1. From Left to Right: $\rho$ = 0.015, $\rho$ = 0.2, $\rho$ = 1 (Bottom Row Only).**

## Theory Surrounding the Use of the Sinkhorn Algorithm

## Optimal Transport

### The Monge Problem and Its Relaxation by Kantorovich

The Monge problem is an optimization problem.

*Definition* 1. Let $X$ and $Y$ be two sets of $N \geq 1$ elements. We define a cost matrix $c := (c_{x,y})_{(x,y)\in X\times Y} \in \mathbb{R}^{X\times Y}$. We denote Bij($X$, $Y$) as the set of bijections from $X$ to $Y$.

$$\min_{\varphi\in \mathrm{Bij}(X,Y)} \sum_{x\in X} c_{x,\varphi(x)} \tag{2.1.1}$$

*Remark* 1. The sets $X$ and $Y$ are often sets of points in the same vector space. Initially, there is an object at each point in $X$. For $(x, y) \in X \times Y$, $c_{x,y}$ models the cost of transporting the object from $x$ to $y$. The goal is to transport the objects from $X$ to $Y$, placing one object at each destination point, while minimizing the cost. Given the interpretation of $c$, it is logical that $c_{x,y}$ be positive, though this is not mathematically crucial for the proofs that follow.

*Remark* 2. Expressed in this way, the problem is purely discrete, and Bij($X$, $Y$) is finite. This becomes a combinatorial problem with rapidly increasing complexity. The problem can be relaxed to facilitate its resolution.

The Monge-Kantorovich problem is, a priori, more general. Let $X$ and $Y$ be two non-empty finite sets with cardinalities $N$ and $M$, respectively. Let $\mu$ and $\nu$ be probability measures on $X$ and $Y$, respectively. The goal is to transport all the mass from $X$ to $Y$ while respecting the initial quantities and final capacities.

*Definition* 2. A transport plan in this context is defined as follows: $\gamma := (\gamma_{x,y})_{(x,y)\in X\times Y}$, a family of positive real numbers, is a transport plan if $\gamma \in \Pi_{\mu,\nu}$, where:

$$\Pi_{\mu,\nu} := \left\{ \gamma \in \mathbb{R}_+^{X\times Y} \mid \forall x \in X, \sum_{y\in Y} \gamma_{x,y} = \mu_x, \forall y \in Y, \sum_{x\in X} \gamma_{x,y} = \nu_y \right\} \tag{2.1.2}$$

*Definition* 3. The cost function for the Monge-Kantorovich problem, using the same cost family $c$ as for Monge, is defined as a linear form:

$$C(\gamma) := \sum_{(x,y)\in X\times Y} c_{x,y}\gamma_{x,y} \tag{2.1.3}$$

*Definition* 4. The Monge-Kantorovich (MK) problem is expressed as:

$$\min_{\gamma \in \Pi_{\mu,\nu}} C(\gamma) \tag{2.1.4}$$

**Proposition 1.** *The (MK) problem admits a solution* [*San15*].

*Proof*. $\Pi_{\mu,\nu}$ is non-empty: the product measure $\mu \times \nu$ is a transport plan by definition. $\Pi_{\mu,\nu}$ is closed. It is bounded, since:

$$\forall \gamma \in \Pi_{\mu,\nu}, \forall (x,y), 0 \leq \gamma_{x,y} \leq 1.$$

Thus, $\Pi_{\mu,\nu}$ is a non-empty compact set.
*C* is a linear map in finite dimensions, hence continuous.
The existence of a minimum follows.

*Remark* 3. With *X* and *Y* of the same size and uniform probabilities, by viewing permutations as matrices: $\left(\varphi \in \mathrm{Bij}(X,Y) \mapsto \left(\nVdash_{\varphi(x)=y}\right)_{x,y}\right)$:

$$\arg \min_{\varphi \in \mathrm{Bij}(X,Y)} \sum_{x \in X} c_{x,\varphi(x)} = \arg \min_{\gamma \in \Pi_{\mu,\nu}} C(\gamma) \cap \mathrm{Perm}(N) \tag{2.1.5}$$

where Perm(*N*) is the set of *N* × *N* permutation matrices. The inclusion to the right is clear. The inclusion to the left requires some intermediate results.

Thus, the (MK) problem, which relaxes (M) (larger set), allows us to recover the solutions of (M).

### Wasserstein Metric

*Definition* 5. $P(\mathbb{R}^n)$ denotes the probability measures on $\mathbb{R}^n$.

$$A(\mathbb{R}^n) := \left\{ \mu \in P(\mathbb{R}^n) \mid \exists N \in \mathbb{N}^*, (x_i) \in (\mathbb{R}^n)^N, (\mu_i) \in [0,1]^N, \mu = \sum_{i=1}^{N} \mu_i \delta_{x_i} \right\} \tag{2.1.6}$$

*Definition* 6. Let *p* ≥ 1. For *X, Y* ⊂ $\mathbb{R}^n$, and *μ, ν* given in (MK), $\mu = \sum_{x \in X} \mu_x \delta_x$ and $\nu = \sum_{y \in Y} \nu_y \delta_y.$

We set $c := (\|y - x\|^p)_{(x,y) \in X \times Y}.$

We define $W_p : A(\mathbb{R}^n) \times A(\mathbb{R}^n) \to \mathbb{R}_+$:

$$W_p(\mu,\nu) := \min_{\gamma \in \Pi_{\mu,\nu}} C(\gamma)^{\frac{1}{p}} \tag{2.1.7}$$

In other words:

$$W_p(\mu,\nu) := \min_{\gamma \in \Pi_{\mu,\nu}} \left( \sum_{(x,y) \in X \times Y} \gamma_{x,y} \|y - x\|^p \right)^{\frac{1}{p}} \tag{2.1.8}$$

*Remark* 4. $W_p$ is a distance on $A(\mathbb{R}^n)$. Wp can be defined analogously on a broader set of probability measures than atomic measures. This metric thus captures, with an optimal transport plan between *μ* and *ν*, an interesting notion of the total displacement required to transition from one mass distribution to another.

### Generalized Entropy

The notion of entropy is fundamental in statistics [3]. It quantifies the dispersion of a distribution and the information required to better understand a phenomenon. However, this notion is only well-defined with a discrete uniform prior. To apply optimal transport and the Sinkhorn algorithm, we need a more general definition that can account for any continuous prior. This is why we introduce the Kullback-Leibler divergence [4].

*Definition* 7: Let *M* be a compact connected set. Let *P* and *Q* be two absolutely continuous measures, i.e., admitting a density, on *M*. The Kullback-Leibler divergence is defined as:

$$KL(P|Q) = \iint_{M\times M} P(x,y)\log\left(\frac{P(x,y)}{Q(x,y)} - 1\right) dx\,dy \tag{2.2.1}$$

### Sinkhorn Algorithm

As seen in subsection 2.1, minimizing the Wasserstein distance is not an easy problem, particularly because it is not convex. Therefore, a regularized Wasserstein distance is used:

$$\mathcal{W}_2^2(\mu_0, \mu_1) = \varepsilon\,(1 + KL(\pi, K_\varepsilon)) \tag{2.3.1}$$

where $K$ is the kernel, typically the Gibbs kernel:

$$K = e^{-\frac{\|x-y\|^2}{\varepsilon}} \tag{2.3.2}$$

The algorithm, equivalent to computing the regularized Wasserstein distance, takes the form [5]:

Perform a number of iterations, with $u$ and $v$ sought such that: $\pi = u \odot (Kv) = a \text{ and } \pi^\top = v \odot (K^\top u) = b$:

$$u = \frac{a}{\langle K, v\rangle} \tag{2.3.3}$$

Compute the error:

$$E_q = E_q + \|v\langle^T K, u\rangle - b\|_1 \tag{2.3.4}$$

Then compute:

$$v = \frac{b}{\langle^T K, u\rangle} \tag{2.3.5}$$

$$E_p = E_p + \|u\langle K, v\rangle - a\|_1 \tag{2.3.6}$$

This algorithm has a complexity of $O(n^3 \ln(n))$, empirically $O(n^2)$, where $n$ is the size of the matrix $\pi$ [6].

### The Heat Equation

The heat equation is a fundamental partial differential equation in mathematics. In three dimensions, it is written as:

$$\frac{\partial f_t}{\partial t} = \frac{\partial^2 f_t}{\partial x_1^2} + \frac{\partial^2 f_t}{\partial x_2^2} + \frac{\partial^2 f_t}{\partial x_3^2} = \Delta f_t \tag{2.4.1}$$

with the initial condition $f_0$ known.

This equation can also be written with the Laplacian $\tilde{L}$, with the same initial condition, as:

$$\frac{\partial f_t}{\partial t} = -f_t{}^t\tilde{L} \tag{2.4.2}$$

### Cholesky Factorization

The Cholesky factorization method used here is a linear algebra technique that simplifies the computation of matrix inverses. It relies on the following proposition:

**Proposition** 2. *Any strictly positive matrix $A$ can be written as $A = L\ {}^tL$, where $L$ is a lower triangular matrix.*

Computing the inverse thus reduces to computing the inverses of triangular matrices, which is much simpler.

### Denoising Meshes and Images Using the Sinkhorn Algorithm

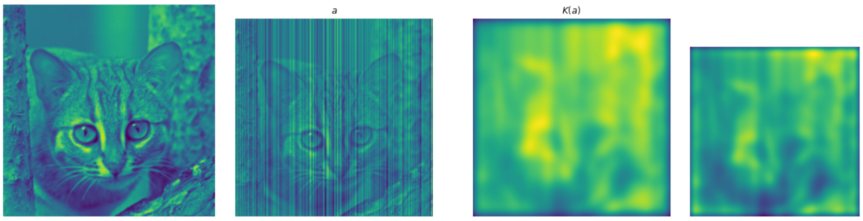

**Figure 2: From left to right: a photo of a cat; the same photo corrupted with white noise of amplitude $\rho$ = 0.2; the kernel of the noisy image; the barycenter b sought after one iteration of the algorithm.**

It is possible to compute the barycenter of a noisy image or mesh to remove parasitic noise, which amounts to solving the problem:

$$\min_b \sum_{k=1}^{R} W_\gamma(a_k, b) \tag{2.6.1}$$

This problem can be solved using the Sinkhorn algorithm [7].
We attempted to denoise an image using the Gibbs kernel, but the results were not very conclusive, highlighting the importance of choosing an appropriate kernel.

### Three Mesh Denoising Methods

We briefly present the three mesh denoising methods we will test. For this, we need to introduce the adjacency matrix of a mesh.

*Definition* 8: (Adjacency Matrix). Let $E$ be the matrix containing the edges and $\varepsilon$ the set of edges of $\mathcal{M}$. We define the matrix $W$ such that:

$$W_{i,j} = \begin{cases} 1 & \text{if } (i,j) \in \mathcal{E}, \\ 0 & \text{otherwise.} \end{cases} \tag{3.0.1}$$

And the vector $d \in \mathbb{N}^n$ such that:

$$d_i = \sum_j W_{i,j} \tag{3.0.2}$$

Setting $D = \text{diag}_i\,(d_i)$, we define the adjacency matrix as:

$$\tilde{W} = D^{-1}W \tag{3.0.3}$$

Note that $\tilde{W}\mathrm{Id}_n$ is a low-pass filter, and the Laplacian $L := D - W$ is a high-pass filter.

### Filtering

The filtering operation consists of multiplying the matrix $X$ of noisy vertices by ${}^t\tilde{W}$ on the right, as many times as necessary.

The algorithm's complexity is that of matrix multiplication. Each iteration is thus $O(n^2)$, where $n$ is the size of the matrix $X$.

### Linear Heat Diffusion

We set $\tilde{L} := D^{-1}L = \mathrm{Id}_n - \tilde{W}$ as the normalized Laplacian.

The idea is to approximate the denoised mesh by a solution to the heat equation 2.4.2. To compute it, we approximate the solution using the Euler method, according to the following recurrence formula:

$$X^{(\ell+1)} = X^{(\ell)} - \tau X^{(\ell)}\,{}^t\tilde{L} = (1-\tau)X^{(\ell)} + \tau X^{(\ell)}\,{}^t\tilde{W} \tag{3.2.1}$$

with $0 < \tau < 1$ a precision parameter. Note that if $\tau = 1$, this reduces to the filtering method described previously.

The algorithm's complexity is that of matrix multiplication. Each iteration is thus $O(n^2)$, where $n$ is the size of the matrix $X$.

### Sobolev Regularization

The idea is that the solution to the problem of minimizing a penalized distance with parameter $\mu > 0$ is given by the linear system:

$$^{t}X_{\mu} = (\mathrm{Id}_n + \mu L)^{-1}\,{}^{t}X \tag{3.3.1}$$

This relies on a Cholesky factorization of the Laplacian using the gradient operator.

The higher μ is, the more significant the mesh smoothing. The complexity of this method is that of matrix multiplication, thus $O(n^2)$, regardless of $\mu$. There are no additional iterations, making this a very fast method.

### Empirical Comparison of These Methods on Two Different Meshes

To empirically test which of the three methods performed best, we applied them to two different meshes representing an elephant (24,955 vertices, 49,918 faces, and 74,877 edges) and the bust of Nefertiti (299 vertices, 562 faces, and 860 edges) [8].

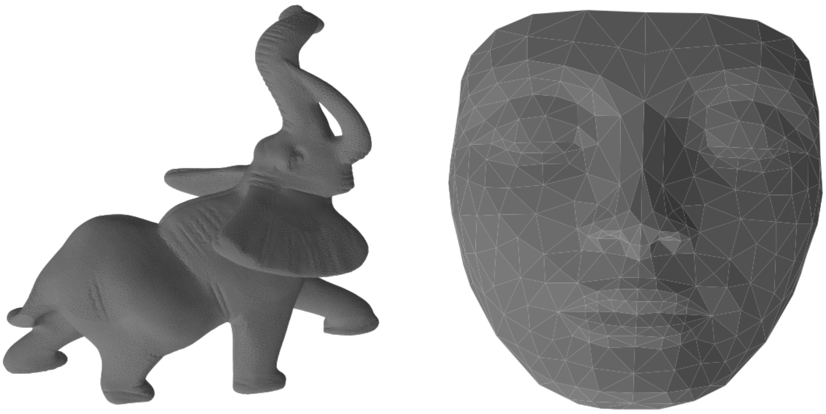

**Figure 3: The original meshes, an elephant and the bust of Nefertiti [8,9].**

We first added white noise to these meshes by displacing the vertices along the normal direction:

$$x_i = x_{0,i} + \rho \epsilon_i N_i \in \mathbb{R}^3 \tag{4.0.1}$$

with $\epsilon_i \sim \mathcal{N}(0,1)$, $N_i \in \mathbb{R}^3$ the normal of each vertex indexed by $i$, and $\rho > 0$ a parameter quantifying the noise level. We compare the results in terms of effectiveness and empirical execution time. To measure effectiveness, we calculate the error in decibels (dB) using the formula:

$$SNR(X,Y) := -20\log_{10}\left(\frac{\|X-Y\|}{\|Y\|}\right) \tag{4.0.2}$$

where $X$ and $Y$ are the vertices of the original and denoised meshes, respectively. A higher $SNR$ indicates a denoised mesh closer to the original ($SNR = +\infty$).

### Comparative Elements and Interpretations

The three denoising methods are relatively effective, as they transform meshes that are difficult for a human to interpret in 3D into identifiable meshes when displayed on a screen.

However, we observe significant differences between the two meshes, due to their size, as both meshes have similar connectivity characteristics proportionally. The three methods perform much better on the larger mesh. Additionally, the higher the noise, the harder it is to recover the original mesh. Filtering and linear heat diffusion yield similar results. Sobolev regularization is slightly less efficient, but only by 1 to 2 dB in each case.

| Mesh | Noisy Image | Filtering | Heat | Sobolev |
|---|---|---|---|---|
| Elephant, $\rho = 0.015$ | 26.77 | 39.82 | 39.81 | 38.13 |
| Elephant, $\rho = 0.2$ | 4.34 | 26.81 | 26.81 | 25.27 |
| Nefertiti, $\rho = 0.015$ | 41.76 | 41.76 | 41.83 | 21.77 |
| Nefertiti, $\rho = 0.2$ | 19.33 | 23.17 | 24.17 | 22.95 |
| Nefertiti, $\rho = 1$ | 5.15 | 14.56 | 14.62 | 13.86 |

**Table 1: Best SNR (in dB) Obtained by the Different Denoising Methods**

| Mesh | Filtering | Heat | Sobolev |
|---|---|---|---|
| Elephant, $\rho = 0.015$ | $14.2 \pm 0.449$ | $9.66 \pm 0.098$ | $0.0355 \pm 0.002$ |
| Elephant, $\rho = 0.2$ | $50.8 \pm 1.38$ | $70.7 \pm 0.6$ | $0.0349 \pm 0.0046$ |
| Nefertiti, $\rho = 0.015$ | 0 | $0.106 \pm 0.0134$ | $0.0831 \pm 0.0178$ |
| Nefertiti, $\rho = 0.2$ | $0.0447 \pm 0.0022$ | $0.507 \pm 0.051$ | $0.101 \pm 0.0275$ |
| Nefertiti, $\rho = 1$ | $0.303 \pm 0.0351$ | $0.465 \pm 0.048$ | $0.0787 \pm 0.016$ |

**Table 2: Machine Usage Time (in milliseconds, ms) for the Best Result Obtained by the Different Denoising Methods, calculated using the Time its Method**

Sobolev regularization is clearly much faster. Moreover, by examining the curves showing execution speed versus the number of iterations and analyzing these, we realize that for equivalent results, Sobolev regularization is always faster (see Figure 4.1). Additionally, the τ chosen to achieve the optimal result using the heat equation is often 1 or close to 1 for the elephant mesh. This method is thus more costly than beneficial for large meshes. However, for the Nefertiti bust, it achieves significantly better denoising quality, with an ideal $\tau$ of around 0.5 (ρ = 0.2) or 0.7 ($\rho$ = 1) (see Figure 4.1).

We also note a difference in optimal parameters between the two meshes for the same noise level. Generally, for the large mesh, they are much higher: $\mu$ = 2 for Sobolev, 3 filter passes, and a time

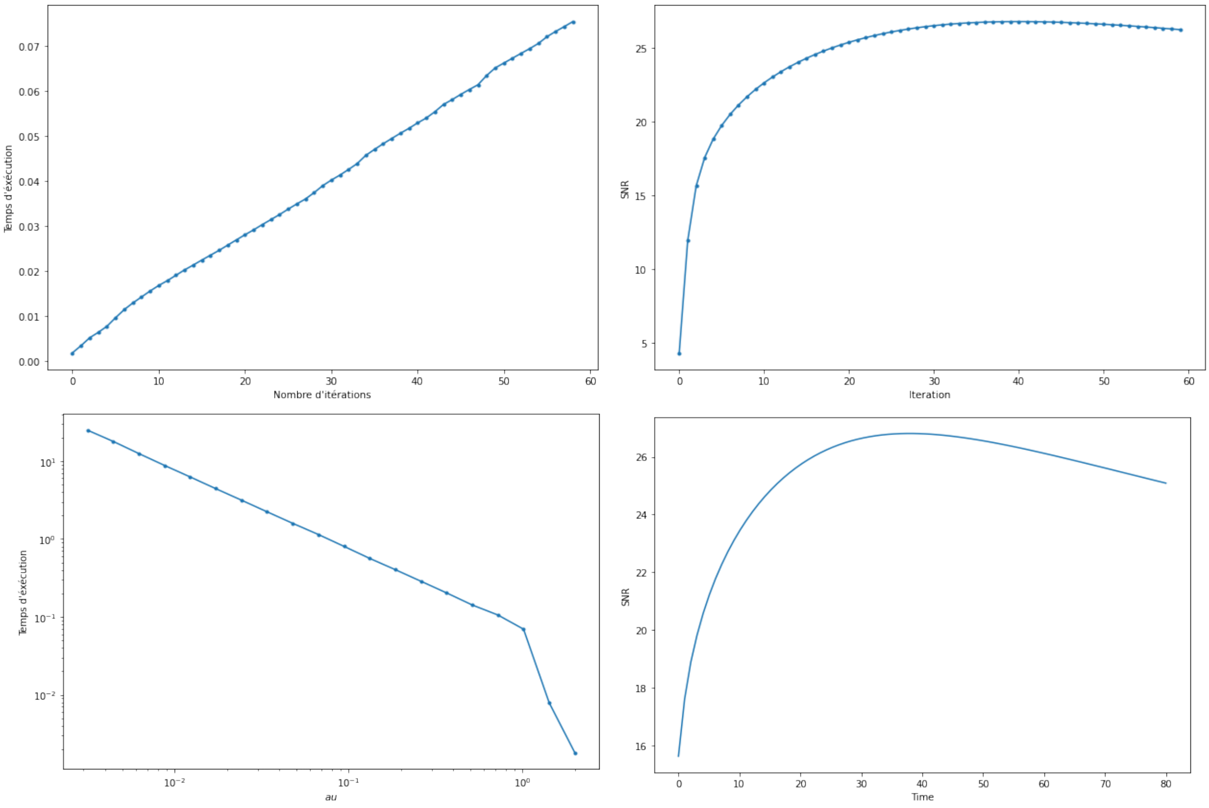

**Figure 4: Top: Processing time versus the number of filtering iterations (left, in seconds) and denoising level versus the number of iterations (right), for the elephant mesh with noise $\rho$ = 0.2. Bottom: Processing time versus the choice of parameter $\tau$ to achieve the best possible result (left, in seconds) and denoising level versus the number of iterations for the most effective and fastest algorithm, with $\tau$ = 1.01, maximum achieved at $T$ = 37.4, with an execution time of 70 ms (right), for the elephant mesh with noise $\rho$ = 0.2.**

Note that to achieve a denoising level similar to Sobolev regularization (25 dB), it takes 20 ms for filtering, as well as for the heat equation method. Since the execution time of the Sobolev algorithm does not depend on the parameter $\mu$, we have all the necessary information for comparison.

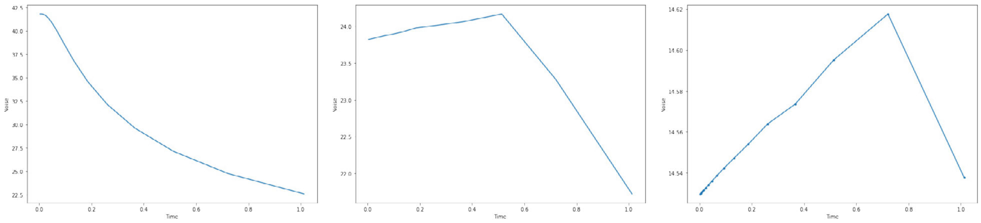

**Figure 5: Best noise reduction after processing by heat diffusion on the Nefertiti busts. From left to right: $\rho$ = 0.015, $\rho$ = 0.2, $\rho$ = 1.**

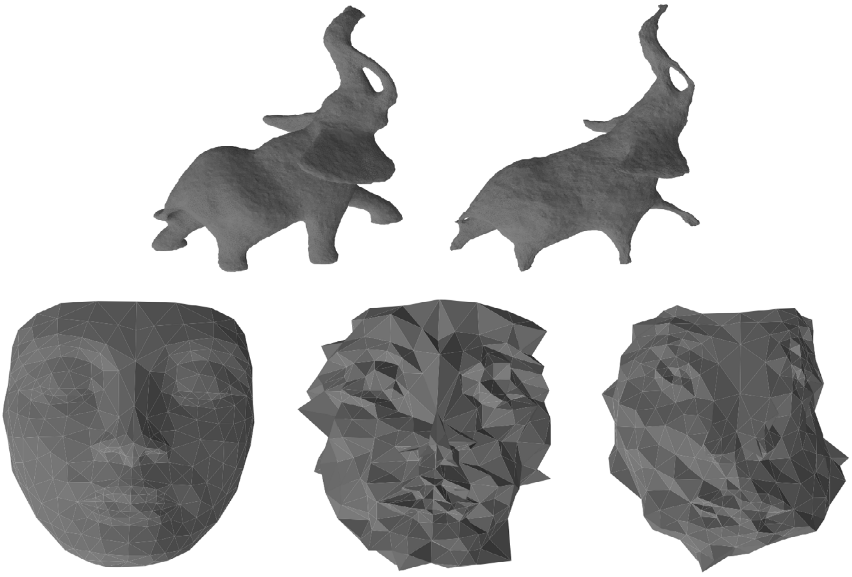

**Figure 6: The various meshes studied, denoised using Sobolev regularization. From left to right: $\rho$ = 0.015, $\rho$ = 0.2, $\rho$ = 1.**

of around 3 suffice to reach the maximum for the Nefertiti mesh with an initial noise of 5.15 dB. However, the higher the noise, the more iterations are needed. For the elephant mesh, with an initial noise of 4.34 dB, to achieve the most effective denoising, one must reach μ = 51, perform 40 filtering iterations, and wait around 40-time units for the heat equation method.

## Conclusion

Our study shows that among the three methods studied in detail, Sobolev regularization is the most efficient for denoising meshes, as it achieves results nearly as accurate as the other two and is significantly faster.

However, three limitations emerge: first, these methods are much less effective on small meshes; second, the maximum achievable SNR is bounded, indicating a theoretical limit to their effectiveness. For example, the elephant's eye is not recovered in any of the denoised meshes, so the reconstruction is far from perfectly satisfactory. Finally, these methods perform poorly when the noise is low.

One avenue we have not explored here to improve mesh denoising would be to use the Sinkhorn algorithm for barycentric interpolation [1]. The challenge lies in choosing the kernel, with the most relevant likely being a solution to the heat equation, which already aids in denoising. Another difficulty is applying this to the specific objects that are meshes.

**Acknowledgments :** Thanks to Gabriel Peyré for the valuable resources used here.

**Competing Interests:** I declare no competing interests.

Supplementary Information The code is available at the following address: https://github.com/cvt8/Meshdenoising.

## References

1. Solomon, J., De Goes, F., Peyré, G., Cuturi, M., Butscher, A., Nguyen, A., ... & Guibas, L. (2015). Convolutional

wasserstein distances: Efficient optimal transportation on geometric domains. ACM Transactions on Graphics (ToG), 34(4), 1-11.

2. Santambrogio, F. (2015). Optimal transport for applied mathematicians (Vol. 87, p. 353). Cham: Springer International Publishing.
3. Shannon, C. E. (1948). A mathematical theory of communication. *The Bell system technical journal, 27*(3), 379-423.
4. Ali, S. M., & Silvey, S. D. (1966). A general class of coefficients of divergence of one distribution from another. *Journal of the Royal Statistical Society: Series B (Methodological), 28*(1), 131-142.
5. Gabriel Peyré. Numerical tour : Entropic regularisation of optimal transport. 2019.
6. Cuturi, M. (2013). Sinkhorn distances: Lightspeed computation of optimal transport. Advances in neural information processing systems, 26.
7. Benamou, J. D. (2015). Iterative Bregman projections for regularized transportation problems. SIAM. J. Sci. Comp., to appear.
8. Gabriel Peyré. Numerical tour : Mesh denoising. 2019.
9. Gabriel Peyré. Numerical tour : Mesh parametrisation. 2019.