Quad Mesh Density Control using Optimal Transport

Submission ID 1015

Abstract

Surface quadrilateral mesh generation plays a fundamental role in CAD/CAE fields. It is crucial to control the density of the meshing structures in practice.

In this work, a novel method for controlling the density of quad meshes is proposed based on the Optimal Transport (OT) theory. This method has a rigorous mathematical foundation and efficient and robust computational algorithms. It provides the designers with the flexibility to control the meshing density by prescribing arbitrary probability density functions on the surface. This work demonstrates the efficiency and efficacy of the method using topological disk models, in principle, the method can be generalized to surfaces with arbitrary topologies.

1 Introduction

Quad meshes, which represent surfaces using quadrilateral elements, are widely used in computer graphics, geometric modeling, and engineering applications due to their alignment with surface features and suitability for tasks such as texture mapping, finite element analysis, and simulation. A key challenge in these applications is efficiently controlling mesh density to balance computational cost and detail preservation. Specifically, too many elements in low-detail regions increase computational costs, while insufficient density in high-curvature regions leads to loss of detail and accuracy. This task becomes even more critical when the density must be adjusted in a user-defined, localized manner—a capability that is not adequately addressed by existing methods.

In this work, we address this gap by introducing a novel framework that enables user-controlled quad mesh density adjustments in specific regions, while maintaining structural integrity and alignment with geometric features. Our approach is based on OT theory and offers a robust mechanism for tailoring mesh density to application-specific needs. The outline of our algorithm is as follows: we use 2D mesh conformally mapped from 3D facial surfaces as our input. As we make sure that the triangle mesh stays Delaunay at all times, we adjust a height vector to change the dual of our mesh, which is a power diagram, representing the desired Brenier's potential. The algorithm goes back and shrinks the step length once it encounters empty power cells or non-convex faces (oriented in different directions) and stops when the dual face's area is close enough to our target area. Finally, we can adopt Zheng's [32] method to generate the quad-mesh from our triangle mesh with new density.

This work's primary contribution lies in enabling precise, user-defined control over quad mesh density in localized areas, addressing a pressing need in applications that require a balance between computational efficiency and geometric fidelity.

As shown in Fig 1, the old man's facial surface is conformally mapped onto a planar rectangular domain using the surface Ricci flow method [12, 11] (upper-left), then we tessellate the rectangle by regular grids on the plane and pull it back to the 3D surface to produce a quad-mesh of the surface (lower left). In order to increase the mesh density around the nose, we apply the OT map. The upper row shows the intermediate results during the OT process, the nose area gets larger and larger. The corresponding quad-meshes are shown in the lower row, the quad-meshes surrounding the nose are getting denser and denser.

Our method can change the density of the parameterization to an arbitrary user-defined distribution. Traditional remeshing techniques often rely on heuristic rules [22, 24] that can lead to uneven density distributions or distorted elements. In contrast, our OT-based framework provides a mathematically rigorous way to transport one distribution to another while minimizing the cost of transformation. The primary contribution of this work is the introduction of an OT-based method for selective density modification in quad-meshes, allowing users to concentrate or reduce mesh density in targeted areas. This capability offers greater flexibility for tasks that require adaptive mesh refinement, such as surface parametrization, finite element analysis, and high-quality rendering. The OT framework also ensures that mesh transformations are smooth and well-aligned with the original geometry, avoiding common pitfalls such as element distortion or irregular mesh transitions.

Our framework aligns well with the geometric features of the surface, minimizing distortion during transformations. Future extensions of this work will focus on integrating curvature-aware strategies, where mesh density dynamically increases in high-curvature regions and decreases in flatter areas. This enhancement will further improve the adaptability and efficiency of the method, making it particularly useful for applications such as adaptive simulation, surface parametrization, and digi-
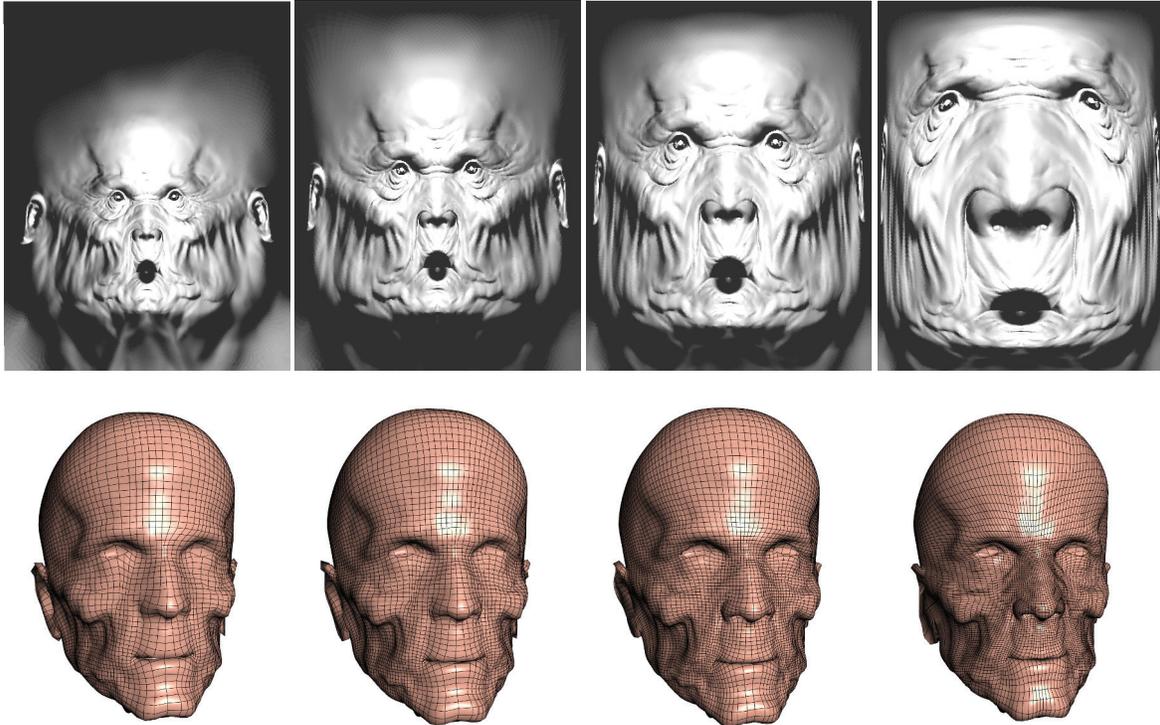
Figure 1: Quad mesh density control by OT map.

tal fabrication. The proposed OT-based approach offers a powerful tool for precise mesh refinement while allowing density control at a specific area defined by the user and contributes to advance state-of-the-art mesh generation and optimization techniques. The outline of this paper is that we introduce the necessary theoretical background in section 3 and the algorithm in section 4, and we display some image results in section 5.

2 Related Work

Mapping 3D Riemannian geometry to a 2D plane, known as parameterization, is crucial in many fields, including computer graphics, medicine, and the transportation industry. Various techniques have been proposed, including cut-based flattening, projection-based methods, and conformal mapping. When the surface is very irregular and complex, conformal mapping is often the default method to ensure the existence of such a mapping. Another advantage is its preservation of angles, allowing local shapes to be preserved. The left figure on figure 2 demonstrates that all 90-degree angles are preserved on the conformal parameterization.

However, in order to preserve angles, conformal mapping often distorts areas. As we can see in figure 3, on the left is the face surface model in 3D, and in the middle is its conformal mapping to a rectangle
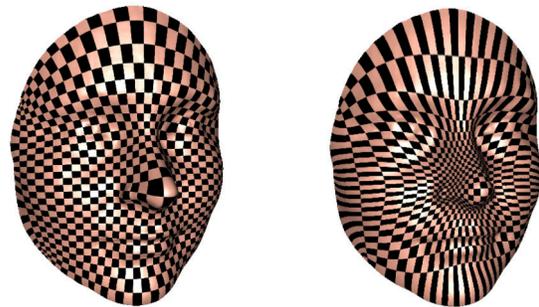
Figure 2: Conformal mapping (left) preserves angles.

in 2D. The nose is shrunk to a smaller area. In many situations, this is undesirable. For instance, a uniform data distribution on the 2D mapping does not correspond to a uniform data distribution. Hence, the need for an area-transforming map is obvious.

OT can help achieve this goal. It can preserve and also change area arbitrarily [31]. The right image of figure 2 and 3 shows an area-distorting and an area-preserving mapping. The nose is of the right size and it retains a uniform distribution in the case of area-preserving map.

An area-preserving mapping can be achieved

Figure 3: *left*: Alex face in 3D; *middle*: parameterization of face in 2D disk through conformal mapping; *right*: area-preserving mapping.

through the composition of OT and conformal mapping. Conformal mapping is often performed through the Ricci flow [16, 15]. OT on the 2D disk can be done using the semi-discrete OT algorithm by Gu [13].

In the recent works of Lei et al. (2021) [32], a novel framework for generating quadrilateral meshes on surfaces with complicated topologies has been introduced based on the Abel-Jacobi theory in algebraic geometry and surface Ricci flow theory. These works give a solid theoretical approach to handling surfaces with realistic topologies. The current work aims to locally adjust the density for the quad-meshes obtained by their method. Therefore, our current proposed method is flexible and general to surfaces with various topologies in reality [32, 3, 4].

Recent works [21, 20, 5] published in the graphics community propose to generate quadrilateral meshes with minimal distortions based on conformal parameterizations and moving frames. Technically, these methods are very promising and achieved impressive results. However, all of these works lack theoretic rigor. As stated in the works of Lei et al. the singularities of a quadrilateral mesh must satisfy the Abel-Jacobi condition, and this condition is missing from these works. Furthermore, in the recent work of Lei [19], it is proven that cross fields are not equivalent to quad-meshes, there are special cross fields with singularities that cannot be converted to quad-meshes.

The works in [2, 25] give thorough surveys for surface quad-mesh and volume hex-mesh generation mainly in the graphics field, but skipped the recent works in computational mechanics fields, therefore these surveys didn't include the most recent advancements in the structured mesh generation field.

There are more recent works on surface mappings [8, 26, 10] which optimize different criteria, such as injectivity, scale-optimality. However, the proposed work requires the full controllability of area distortion, therefore, OT map is the best candidate, due to its theoretic rigor [14]

For mesh density controlling tasks, several heuris-

tic approaches aim to adapt quad-mesh density by adjusting mesh elements according to geometric features or computational requirements. Advancing front algorithms, such as Q-Morph [23], are widely used to propagate a front through the mesh and generate quads based on local topology and curvature. Q-Morph utilizes an indirect triangulation method to determine element sizes and alignments, helping to maintain mesh quality while achieving the desired density in critical regions. Similarly, Laplacian smoothing and vertex-based operations (e.g., edge-swap and splitting) are common strategies for improving mesh regularity by dynamically redistributing vertices. These methods aim to increase density in high-curvature regions and reduce it in flatter areas, but they often rely on trial-and-error rules with limited theoretical guarantees [6].

OT has found numerous applications in computer graphics and geometric modeling due to its ability to measure distances between distributions and facilitate transformations. Previous works have extensively applied OT in texture mapping [7], shape registration [27], and surface correspondence [9]. Zhao et al. [31] utilized OT in the preservation of surface area, and Su et al. [28] extended it to the preservation of volume. An et al. [1] focused on adaptive sampling on the parameter domain to fit to the geometric complexity of the input surface in order to produce high quality, local adaptive triangular meshes. However, OT's potential for mesh modification, particularly in controlling partial mesh densities, remains relatively underexplored.

Our approach builds upon these efforts by leveraging OT for targeted density manipulation in quad meshes by introducing an OT-based framework that modifies partial density in meshes. This approach leverages the Brenier map to ensure optimal transformation, providing significant improvements in flexibility and precision over existing methods.

3 Theoretical Foundation

This section provides a concise overview of the theoretical foundations of Optimal Transport. OT is a mathematical framework for determining the most efficient way to transport mass from one probability distribution to another, minimizing a specified cost function. For a comprehensive exploration of OT and its principles, refer to the detailed discussion in [18].

3.1 Optimal Transport Monge first raised the optimal mass transport problem, which asks for the optimal way to move a pile of mass from one place to another that results in the minimum transport cost. OT theory essentially studies the problem of transforming one distribution to another in the most economical way.

Let X and Y be two metric spaces with probability measures μ and ν respectively, and assume X and Y have equal total measure:

$$\int_X d\mu = \int_Y d\nu$$

Then we want a transport map $T : X \rightarrow Y$ that is *measure preserving*: for any measurable set $B \subset Y$, we have:

$$\mu(T^{-1}(B)) = \nu(B)$$

and we call ν the push-forward of μ by T and denote $\nu = T_{\#}\mu$.

Let the *transport cost* for moving $x \in X$ to $y \in Y$ be $c(x,y)$, $c(x,y) : X \times Y \rightarrow \mathbb{R}$. Then the total transport cost is

$$C(T) = \int_X c(x, T(x))d\mu(x)$$

in most cases, $c(x,y)$ is either the L^1 cost $|x - y|$ or the L^2 cost $|x - y|^2$. OT theory therefore studies measure-preserving maps with the minimum total transport cost.

3.1.1 Monge and Kantorovich's Approach

Monge's Optimal Mass transport is in the most fundamental form: given c, μ, ν, find $W_c(\mu, \nu)$ among all measure preserving transport map T, where

$$W_c(\mu, \nu) = \min_T \{ \int_X c(x, T(x))d\mu(x) : T_{\#}\mu = \nu \}$$

There are some problems with Monge's approach: for example, it could not represent splitting a single mass into two halves. As a result, when X is a single point 0 with mass 1 and Y contains two points, each with mass 0.5, there does not exist a transport map.

Kantorovich [17] relaxed Monge's problem in the 1940s and proved the solution's existence and uniqueness. He introduced a joint measure $\rho(A, B)$ that can represent all measure-preserving transport, called a transport plan. $\rho(A, B)$ can be seen as the portion of mass in pile A that is moving to pile B, represented as a joint distribution. We have $\rho(A*Y) = \mu(A), \rho(X \times B) = \nu(B)$ for all $A \subseteq X$ and $B \subseteq Y$. The projection to X and Y are denoted π_x and π_y respectively: $\pi_x\rho = \mu, \pi_y\rho = \nu$. The total cost thus becomes

$$C(\rho) = \int_{X \times Y} c(x, y)d\rho(x, y)$$

and therefore Monge-Kantorovich problem aims to find an OT plan ρ among all transport plans that minimize the cost $C(\rho)$:

(3.1)
$$W_c(\mu, \nu) = \min_{\rho} \{ \int_{X \times Y} c(x, y)d\rho(x, y) : \\ \pi_x\rho = \mu, \pi_y\rho = \nu \}$$

where $W_c(\mu, \nu)$ denotes the minimum cost $C(\rho)$ among all transport plans.

Since equation 3.1 is a linear program, it has a dual formulation, known as Kantorovich dual problem:

$$W_c(\mu, \nu) = \max_{\varphi, \psi} \{ \int_X \varphi(x)d\mu(x) + \int_Y \psi(y)d\nu(y) : \\ \varphi(x) + \psi(y) \leq c(x, y) \}$$

We give a brief proof:

Proof. Let $\Pi(X, Y)$ denote the space of all transport plans: $\Pi(X, Y) = \{ \gamma \in \rho(X, Y) : (\pi_x)_{\#}\gamma = \mu, (\pi_y)_{\#}\gamma = \nu \}$, where π_x, π_y are the marginal distribution of X and Y respectively, then we have:

$$\sup_{\varphi, \psi} \int_X \varphi d\mu + \int_Y \psi d\nu - \int_{X \times Y} (\varphi(x) + \psi(y))d\gamma = 0$$

for all $\gamma \in \Pi(X, Y)$ and equals infinity otherwise. Under certain conditions, we can write:

$$\inf_{\gamma} \int_{X \times Y} c d\gamma + \sup_{\varphi, \psi} \int_X \varphi d\mu + \int_Y \psi d\nu \\ - \int_{X \times Y} (\varphi(x) + \psi(y))d\gamma$$

as

$$\sup_{\varphi, \psi} \int_X \varphi d\mu + \int_Y \psi d\nu + \inf_{\gamma} \int_{X \times Y} c d\gamma \\ - \int_{X \times Y} (\varphi(x) + \psi(y))d\gamma$$

which is

$$\sup_{\varphi, \psi} \int_X \varphi d\mu + \int_Y \psi d\nu + \inf_{\gamma} \int_{X \times Y} (c - \varphi(x) - \psi(y))d\gamma$$

so we can write the original program as

$$\max_{\varphi, \psi} \{ \int_X \varphi(x)d\mu(x) + \int_Y \psi(y)d\nu(y) \}$$

with constraint: $\varphi(x) + \psi(y) \leq c(x, y)$. $\qquad \square$

Equivalently, ψ can be replaced by the *c-transform* of φ: $\varphi^c(y) = \inf_{x \in X}(c(x, y) - \varphi(x))$:

$$W_c(\mu, \nu) = \max_{\varphi} \left\{ \int_X \varphi(x)d\mu(x) + \int_Y \varphi^c(y)d\nu(y) \right\}$$

we call φ the *Kantorovich potential*. If the transport cost is L^2: $c(x, y) = \frac{1}{2}|x - y|^2$, then there's a relationship between the c-transform and the classical Legendre transformation of φ:

$$\frac{1}{2}|y|^2 - \varphi^c = \left(\frac{1}{2}|x|^2 - \varphi \right)^*$$

3.1.2 Brenier's Approach By the end of 1980's, Brenier showed that

THEOREM 3.1. *Suppose X and Y are the Euclidean space $\mathbb{R}^n$, and the transportation cost is the quadratic Euclidean distance $c(x,y) = |x-y|^2$. If μ is absolutely continuous and μ and ν have finite second order moments, then there exists a convex function $u : X \to \mathbb{R}$, its gradient map ∇u gives the solution to the Monge's problem, where u is called Brenier's potential. Furthermore, the optimal mass transportation map is unique.*

This theorem converts Monge's problem to solving the pde:

$$det\left(\frac{\partial^2 u}{\partial x_i \partial x_j}(x)\right) = \frac{\mu(x)}{\nu \circ \nabla u(x)}$$

u is called *Brenier potential*. When $c(x,y) = \frac{1}{2}|x-y|^2$, Brenier's potential u and Kantorovich's potential φ are related:

$$u(x) = \frac{x^2}{2} - \varphi(x)$$

3.2 Convex Geometry and Discrete OT OT in the discrete case maps one discrete distribution to another. With L^2 transport cost, it has an intrinsic connection with convex geometry theories, especially Alexandrov's theory. Specifically, finding an OT map in the discrete case is equivalent to constructing a convex polytope with user-defined face volume and normal vectors. Alexandrov's theory is a generalization of Minkowski's theory.

THEOREM 3.2. (MINKOWSKI) *Suppose $\{n_1, ..., n_k\}$ are unit vectors which span $\mathbb{R}^n$ and $\nu_1, ..., \nu_k > 0$ so that $\sum_{i=1}^{k} \nu_i n_i = 0$. There exists a compact convex polytope $P \subset \mathbb{R}^n$ with exactly k codimension-1 faces $F_1, ..., F_k$ so that n_i is the outward normal vector to F_i and the volume of F_i is ν_i. Furthermore, such P is unique up to parallel translation.*

Minkowski's proof is variational. The unbounded convex polytope was considered and solved by A.D. Alexandrov and his student A. Pogorelov.

THEOREM 3.3. (ALEXANDROV) *Suppose Ω is a compact convex polytope with non-empty interior in $\mathbb{R}^n$, $\{n_1, ..., n_k\} \subset \mathbb{R}^{n+1}$ are distinct k unit vectors, the $(n+1)^{th}$ coordinates are negative, and $\nu_1, ..., \nu_k > 0$ so that $\sum_{i=1}^{k} \nu_i = vol(\Omega)$. Then there exists convex polytope $P \subset \mathbb{R}^{n+1}$ with exact k codimension-1 faces $F_1, ..., F_k$ so that n_i is the normal vector to F_i and the intersection between Ω and the projection of F_i is with volume ν_i. Furthermore, such P is unique up to vertical translation.*

By viewing the normal vectors as the target of the transport map, and volumes as the target measure, we can see that constructing such a polytope is indeed equivalent to finding an OT map. Alexandrov's proof was based on algebraic topology and was non-constructive. Gu et al.[30] gave a variational proof for the generalized Alexandrov's theorem in terms of convex functions.

Given $p_1, ..., p_k \in \mathbb{R}^n$ and $h = (h_1, ..., h_k) \in \mathbb{R}^k$, a piecewise linear function $u_h(x)$ is defined as

$$u_h(x) = \max_i\{\langle x, p_i \rangle + h_i\};$$

Then dual function $u^*(y)$ has domain $Conv(p_1, ... p_k)$ and

$$u^*(y) = \min\left\{-\sum_{i=1}^{k} t_i h_i | t_i \ge 0 \sum_{i=1}^{k} t_i = 1, \sum_{i=1}^{k} t_i p_i = y\right\}$$

Therefore, if $p_i \notin Conv(p_1, ..., p_{i-1}, p_{i+1}, ..., p_k)$, then $u^*(p_i) = -h_i$.

The graph of u is a convex polytope in $\mathbb{R}^{n+1}$, and its projection on $\mathbb{R}^n$ induces a cell decomposition where each cell is a closed convex polytope

$$W_i(h) = \{x \in \mathbb{R}^n | \nabla u_h(x) = p_i\}$$

It's well known that some cells defined this way may be unbounded or empty. The volume $w_i(h)$ of each cell, given a probability measure μ on Ω, is

$$w_i(h) = \mu(W_i(h) \cap \Omega) = \int_{W_i(h) \cap \Omega} d\mu$$

THEOREM 3.4. (GU-LUO-SUN-YAU) *Let Ω be a compact convex domain in $\mathbb{R}^n$, $\{y_1, ..., y_k\}$ be a set of distinct points in $\mathbb{R}^n$ and μ a probability measure on Ω. Then for any $\nu_1, ..., \nu_k > 0$ with $\sum_{i=1}^{k} \nu_i = \mu(\Omega)$, there exists $h = (h_1, ..., h_k) \in \mathbb{R}^k$, unique up to adding a constant $(c, ..., c)$, so that $w_i(h) = \nu_i$, for all i. The vectors h are exactly the minimum points of the convex function*

$$(3.2) \qquad E(h) = \int_0^h \sum_{i=1}^{k} w_i(\eta) d\eta_i - \sum_{i=1}^{k} h_i \nu_i$$

on the open convex set

$$H = \{h \in \mathbb{R}^k | w_i(h) > 0, \forall i\}$$

Furthermore, ∇u_h minimize the quadratic cost

$$\int_\Omega |x - T(x)|^2 d\mu(x)$$

among all transport maps $T_{\#}\mu = \nu$, where ν is the Dirac mass $\sum_{i=1}^{k} \nu_i \delta_{y_i}$

3.3 Power diagram Before getting to the algorithm itself, we recall power diagrams. Convex subdivisions of a piecewise linear convex function $u_h(x)$ are exactly the same as a power diagram. Suppose $\{y_1, ..., y_k\}$ are k points in $\mathbb{R}^n$, and $\{m_1, ..., m_k\}$ are k real numbers. Then *power distance* given a point y_i and a power weight m_i is

$$pow(x, y_i) = |x - y_i|^2 - m_i$$

and a *power diagram* is the cell decomposition of $\mathbb{R}^n$

$$\mathbb{R}^n = \bigcup_{i=1}^{k} W_i(m)$$

where each cell is a convex polytope

$$W_i(m) = \{x \in \mathbb{R}^n | pow(x, y_i) \leq pow(x, y_j), \forall j\}$$

The weighted Delaunay triangulation is the Poincaré dual of power diagram, and when the power weights are all 0, then the power diagram becomes Delaunay decomposition. Note that if $pow(x, y_i) \leq pow(x, y_j)$, then

$$|x - y_i|^2 - m_i \leq |x, y_j|^2 - m_j$$
$$x^2 - 2xy_i + y_i^2 - m_i \leq x^2 - 2xy_j + y_j^2 - m_j$$
$$\langle x, y_i \rangle - \frac{1}{2}(y_i^2 - m_i) \geq \langle x, y_j \rangle - \frac{1}{2}(y_j^2 - m_j)$$

setting $h_i = -\frac{1}{2}(y_i^2 - m_i)$, we have the piecewise linear convex function $u_h(x) = \max_i \{\langle x, y_i \rangle + h_i\}$

3.4 Convex optimization Now, we have all the tools to explain the gradient and the Hessian of the energy defined in equation 3.2. By definition

$$\nabla E(h) = (w_1(h) - \nu_1, w_2(h) - \nu_2, ..., w_k(h) - \nu_k)^T$$

The Hessian of energy $E(h)$ is given by

$$\frac{\partial^2 E(h)}{\partial h_i \partial h_j} = \frac{\partial(w_i(h))}{\partial h_j} = -\frac{\mu(W_i(h) \cap W_j(h) \cap \Omega)}{|y_j - y_i|}$$

that is, suppose edge e_{ij} is in the weighted Delaunay triangulation, connecting y_i, y_j, then there exists a unique dual cell D_{ij} in the power diagram and

$$\frac{\partial^2 E(h)}{\partial h_i \partial h_j} = -\frac{\mu(D_{ij})}{|e_{ij}|}$$

which is the volume ratio between dual cells, and the diagonal elements of the Hessian is

$$\frac{\partial^2 E(h)}{\partial^2 h_i} = \frac{\partial(w_i(h))}{\partial h_i} = -\sum_{j:j \neq i} \frac{\partial(w_j(h))}{\partial h_i}$$

We prove this by first proving a lemma.

LEMMA 3.1. *Suppose X is a compact domain in $\mathbb{R}^n$, $f : X \to \mathbb{R}$ is a non-negative continuous function and $\tau(x, t) : \{(x, t) \in X \times \mathbb{R} | 0 \leq t \leq f(x)\} \to \mathbb{R}$ is continuous. For each $t \geq 0$, let $f_t(x) = \min\{t, f(x)\}$. Then $W(t) = \int_X (\int_0^{f_t(x)} \tau(x, s) ds) dx$ satisfy*

$$\lim_{t \to t_0^+} \frac{W(t) - W(t_0)}{t - t_0} = \int_{x | f(x) \geq t_0} \tau(x, t_0) dx$$

and

$$\lim_{t \to t_0^-} \frac{W(t) - W(t_0)}{t - t_0} = \int_{x | f(x) > t_0} \tau(x, t_0) dx$$

it is differentiable at t_0 iff $\int_{x | f(x) = t_0} \tau(x, t_0) dx = 0$

Proof. Let $G_t(x) = \int_0^{f_t(x)} \tau(x, s) ds$, and M be the upper bound for $\tau(x, t)$ on its domain. Since $min(a, b) - min(a, c) \leq |b - c|$, we have $|f_t(x) - f_{t_0}(x)| \leq |t - t_0|$. For any $t \neq t_0$:

$$\left| \frac{G_t(x) - G_{t_0}(x)}{t - t_0} \right| = \frac{|\int_{f_{t_0}(x)}^{f_t(x)} \tau(x, s) ds|}{|t - t_0|}$$
$$\leq \frac{M}{|t - t_0|} |f_t(x) - f_{t_0}(x)| \leq M$$

fix t_0 and x. If $f(x) < t_0$, then for t sufficiently close to t_0, $G_t(x) = \int_0^{f_t(x)} \tau(x, s) ds$, and $\lim_{t \to t_0} \frac{G_t(x) - G_{t_0}(x)}{t - t_0} = 0$. If $f(x) > t_0$, then for t sufficiently close to t_0, $G_t(x) = \int_0^{f_{t_0}(x)} \tau(x, s) ds$, and $\lim_{t \to t_0} \frac{G_t(x) - G_{t_0}(x)}{t - t_0} = \lim_{t \to t_0} \frac{1}{t - t_0} \int_{f_{t_0}(x)}^{f_t(x)} \tau(x, s) ds = \tau(x, t_0)$. If $f(x) = t_0$, the above calculations showed that $\lim_{t \to t_0^+} \frac{G_t(x) - G_{t_0}(x)}{t - t_0} = \tau(x, t_0)$, and $\lim_{t \to t_0^-} \frac{G_t(x) - G_{t_0}(x)}{t - t_0} = 0$. Then by Lebesgue dominated theorem, we have

$$\lim_{t \to t_0^+} \frac{W(t) - W(t_0)}{t - t_0} = \lim_{t \to t_0^+} \int_X \frac{G_t(x) - G_{t_0}(x)}{t - t_0} dx$$
$$= \int_{\{x | f(x) \geq t_0\}} \tau(x, t_0) dx$$
$$\lim_{t \to t_0^-} \frac{W(t) - W(t_0)}{t - t_0} = \lim_{t \to t_0^-} \int_X \frac{G_t(x) - G_{t_0}(x)}{t - t_0} dx$$
$$= \int_{\{x | f(x) > t_0\}} \tau(x, t_0) dx$$

Hence the lemma is established. $\square$

Fix $a < b$, then $\{(x, t) \in X \times \mathbb{R} | a \leq f(x), a \leq t \leq min(f(x), b)\}$ is called a cap domain with base $\{x | f(x) \geq a\}$ and top $\{x | f(x) \geq b\}$ of height $(b - a)$

associated with function f.

To show the elements of the Hessian matrix, let $h' = (h_1, ..., h_{i-1}, h_i + \delta, h_{i+1}..., h_k)$. For small $\delta > 0$, by definition, $W_i(h) \subset W_i(h')$ and $W_j(h') \subset W_j(h)$. If $W_i(h) \cap W_j(h) \cap \Omega = \emptyset$, then for small δ, $W_j(h) = W_j(h')$. Hence $\frac{\partial W_j(h)}{\partial h_i} = 0$. If $W_i(h) \cap \Omega$ and $W_j(h) \cap \Omega$ share a co-dimension 1 face F, then $cl(W_j(h) - W_j(h'))$ is a cap domain with base F associated with function f defined on F. The height of the domain is $\frac{\delta}{|y_i - y_j|}$, and f is piecewise linear convex, so any $(n-1)$ dimensional Lebesgue measure of the form $\{x \in F | f(x) = t\}$ is 0. Furthermore, for $\delta > 0$, by definition

$$\frac{w_j(h) - w_j(h')}{\delta} = \frac{1}{\delta} \int_{W_j(h) \cap \Omega - W_j(h') \cap \Omega} \mu(x) dx$$
$$= \frac{1}{\delta} \int_F \int_0^{f_t(x')} \tau(x', s) ds dx'$$

where $x' \in F$ are the Euclidean coordinates, $\tau(x', s)$ is μ expressed in new coordinates. Then by the previous lemma, we have

$$\lim_{\delta \to 0^+} \frac{w_j(h) - w_j(h')}{\delta} = \frac{1}{|y_i - y_j|} \int_F \mu|_F dA$$

where dA is the area form on F. With a negative δ, we can see that this equation hold as well with $cl(W_j(h') - W_j(h))$ has cap F, and we can obtain the same formula as non-diagonal elements of the Hessian of our energy in 3.2. If $W_i(h)$ and $W_j(h)$ share a face of dimension less than $n-1$, this equation holds as well with a zero measure top or base.

4 Algorithms

The discrete OT algorithm has several important parts: Delaunay triangulation and Dual graph calculation. We will introduce these two algorithms first.

We first map the 3D model onto the 2D unit disk through conformal methods such as Ricci flow [16], the 2D parameterization Ω is a source triangle mesh consisting of vertices V_s, edges E_s, faces F_s and half-edges H_s. It will be used as our input.

Input: A source triangle mesh $\Omega = (V_s, E_s, F_s, H_s)$, a target triangle mesh $P = (p_1, p_2, ..., p_n)$, and user-defined target area for each vertex $\nu = (\nu_1, \nu_2, ..., \nu_{|V_s|})$, where $\sum_{i=1}^{|V_s|} \nu_i =$ area of P.

Output: The source mesh in target shape P and vertices with target area ν.

Notation: for any vertex v, v.vertexfaces represents all faces that have v as one of its vertex, for any edge e, e.edgefaces represents all faces that have e as one of its edge, the logic is the same for all similar expression.

4.1 Delaunay triangulation The Delaunay triangulation algorithm 4.1 checks if the current mesh is Delaunay by checking if all edges are local Delaunay. If not, we will try to flip that edge by removing it and connecting the other two vertices of the two faces that share this edge. All half-edges in all faces are oriented counter-clockwise, thus any edge that results in a clockwise-oriented face after flipping is not flippable.

ALGORITHM 4.1. Delaunay triangulation

```
     function DELAUNAY TRIANGULATION(Ω)
         for e ∈ all non-boundary edges in Es do
3:           if e is not local Delaunay then
                 illegal edge stack.push(e)
             end if
6:       end for
         for e ∈ illegal edge stack do
             if e is not local Delaunay then
9:               stack.pop(e)
                 for face f1, f2 has edge e do
                     if f1, f2 are counter-clockwise orient
         then
12:                      flip e
                         stack.push(other 4 edges of f1, f2)
                     end if
15:              end for
             end if
         end for
18:      if e is Delaunay ∀e ∈ Es then
             return False
         else
21:          return True
         end if
     end function
```

4.2 Dual graph Algorithm 4.2 calculates the dual of the mesh. Dual points are defined to be the unit normal of each face, dual edges are edges that connect dual points of neighboring faces, and dual faces are then naturally defined but restricted inside the target domain.

ALGORITHM 4.2. Dual calculation

```
     function DUAL CALCULATION(Ω)
         for f ∈ Fs do
3:           f: ax+by-c=z
             (a,b,c) = f.dualpoint
         end for
6:       for v ∈ Vs do
             for finv.vertexFaces do
                 connect f.dualpoint
9:               v.dualface = resulting polygon
                 if v.dualface > target domain then
                     clip v.dualface
```

12:                     **end if**
                    end for
                end for
15: **end function**

4.3 Semi-Discrete OT
Now we can move on to the main algorithm.

ALGORITHM 4.3. SEMI-DISCRETE Optimal Transport Map

    function SEMI-DISCRETE OT(Ω, P, ν)
        Translate and scale Ω s.t. $\Omega \subset P$;
3:    **for** each v in Ω **do**
        v.area += v.vertexfaces.area/3
    end for
6:    scale v.area s.t. sum is Ω area
    Initialize a height vector $h_i^0 = \frac{1}{2}|v_i|^2$
    Call function Delaunay triangulation 4.1
9:    Call function Dual Calculation 4.2
    repeat
        Compute gradient of energy E:

$$E(h^n) = \int_0^{h^n} \sum_{i=1}^{k} v_i.\text{dualface.area } dh_i - \sum_{i=1}^{k} \nu_i h_i^n$$

$$\nabla E(h^n) = v_i.\text{dualface.area} - v_i.\text{area}$$

12:    Compute the Hessian matrix for $E(h^n)$

$$\frac{\partial^2 E(h^n))}{\partial h_i \partial h_j} = \frac{\partial w_i(h^n)}{\partial h_j} = -\frac{e_{ij}.\text{length}}{e_{ij}.\text{dualedge.length}}$$

$$\frac{\partial^2 E(h^n))}{\partial^2 h_i} = \frac{\partial w_i(h^n)}{\partial h_i} = -\sum_{j:i \neq j} \frac{\partial w_j(h^n)}{\partial h_i};$$

        Solve $Hess(h^n)d = \nabla E(h^n)$
        Set initial step length $\lambda = 1$
15:    $h^{n+1} \leftarrow h^n + \lambda d$;
        Call Delaunay triangulation 4.1
        if return is `False` **then**
18:        Roll back, $\lambda = \lambda/2$
        end if
        if any power cell is empty **then**
21:        Roll back, $\lambda = \lambda/2$
        end if
        Compute the error $\epsilon = \sum$ v.targetarea - v.area,
24:    **until** $\epsilon \leq$ some threshold
    v.coordinates = v.dualface.center
    Output new Ω
27: **end function**

4.4 Image illustration
We provide some image illustrations. Figure 4 shows the input of the discrete OT algorithm. On the left we have the face mesh in 3D, and in the middle is its conformal projection onto the 2D plane. Our target domain in this case is a rectangle with all edges 1.1 times bigger than our source. The target domain can be any convex shape.
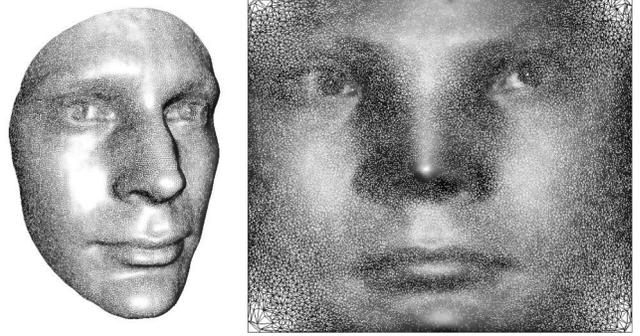
Figure 4: OT algorithm input

Figure 5 represents the Delaunay triangulation of the mesh after edge flipping, it is the 2D mesh in 3D with our height vector. Delaunay triangulation is done based on orientation in 3D.
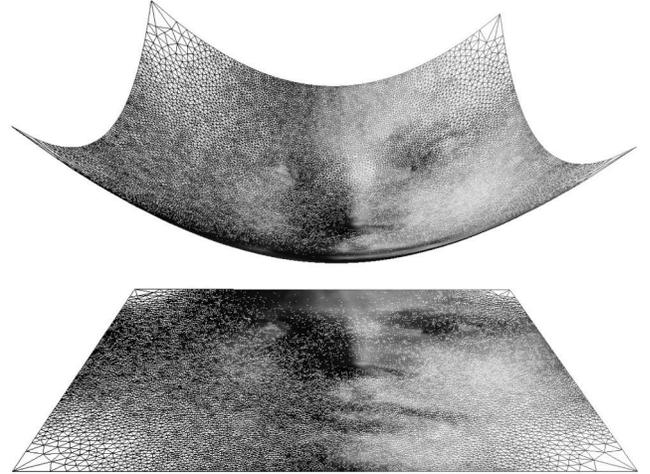
Figure 5: Delaunay triangulation

The target mesh P is a 2D rectangular region while the source mesh Ω is the input mesh after parameterization (2D). Fig. 6 illustrates the concept of domain Ω and the probability density ν. Suppose $\nu(x,y) = f(x,y)dxdy$, we use the color temperature to visualize the density function $f(x,y)$.

Figure 7 shows the unclipped upper envelop with an infinity vertex added. The infinity vertex is used to help calculate dual points of the boundary faces and dual face of the boundary vertex. While figure 8 shows the upper envelope clipped within the target domain using Sutherland-Hodgman algorithm. It is also equivalent to
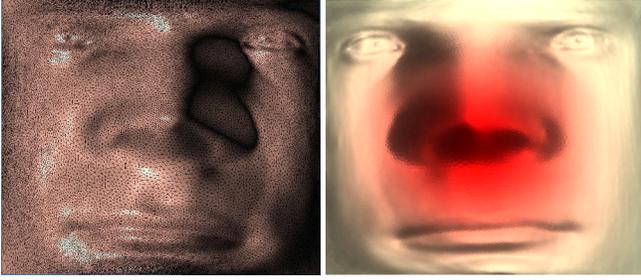
Figure 6: *Left:* The source triangle mesh Ω in the rectangular target shape P with target area ν. *Right:* Color temperature representing the density. Red reflects a higher density and yellow corresponds to lower.

a power diagram.
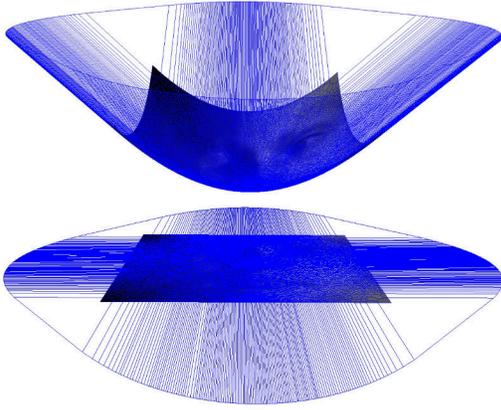
Figure 7: The final Brenier potential $u(x) : \Omega \to \mathbb{R}$, represented as the unclipped upper envelop of the supporting planes. Its projection on the plane induces the planar power diagram of the sample vertices.

5 Experimental Results

All the computational algorithms are implemented using generic C++ on Windows platform using Visual Studio 2022. All the experiments are conducted on a laptop with Intel(R) Core(TM) CPU i9-12900H @2.9GHz with 14 cores and 32GB of memory.

The testing 3D meshes include human facial surfaces (the Alex and Sophie) acquired by our 3D scanning system based on phase-shifting structure light with depth resolution 0.1 mm, and facial surfaces created by modeling tool (the oldman model). The scanned models are preprocessed to remove geometric and topological noises. In order to test the robustness of our method, the surface models are not remeshed and with the original triangulation.

The input models are conformally mapped onto
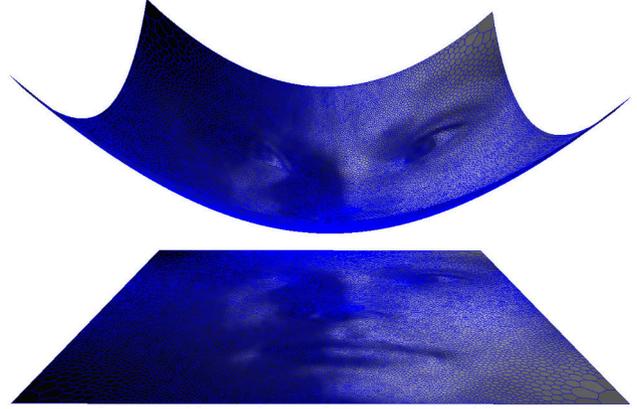
Figure 8: The clipped final Brenier potential $u(x) :$ $\Omega \to \mathbb{R}$ and its projection onto the planar domain Ω, which induces a power diagram of Ω, each power cell is mapped to the target vertex, which gives the semi-discrete optimal transportation map.

a planar rectangle using the Ricci flow algorithm for extremal length [16]. First, four corner vertices on the surface boundary are manually selected. Although Wise et al [29] proposed several methods without the need to pick points manually. However, it is difficult to adopt this method to structured meshes due to the topological barriers. Second, we set the target curvature for all vertices: zeros for interior vertices, zeros for boundary vertices except the four corners, and $\pi/2$'s for the corner vertices. Third, we use Ricci flow method to conformally deform the Riemannian metric to realize the target curvature. By discrete Ricci flow theory, the solution exists and is unique. Then we isometrically embed the surface onto the plane, the image is a planar rectangle.

After we obtain the conformal mapping, we compute the OT map. First, we enlarge the planar rectangle by a factor 1.1. Second, we normalize the total surface area to the total area of the scaled rectangle. Then for each vertex v_i, we compute the areas of the adjacent triangular faces and divide them by 3 to obtain the target measure ν_i associated with the vertex. We use the conformal mapping image $\varphi(v_i)$ as the target point, then construct the target measure as

$$\nu = \sum_{i=1}^{n} \nu_i \delta(y - \varphi(v_i)).$$

Fourth, we run the OT algorithm to compute the OT map from the scaled rectangle the discrete measure ν. This produces a power cell decomposition of Ω. Finally, each vertex is mapped to the corresponding power cell center.

In order to quantitatively measure the quality of our computational results, we have conducted the numerical testings as follows: the optimal transportation map is approximated by a piecewise linear map from the input triangle mesh to the output mesh, we compute the ratio between the area of the target triangle and the initial triangle on the surface mesh, and plot the histogram of the logarithm of the ratios. As shown in Fig. 9, it is clear that the histogram highly is concentrated near the 0.54 point, showing that OT map preserves relative area proportions despite a uniform scaling.
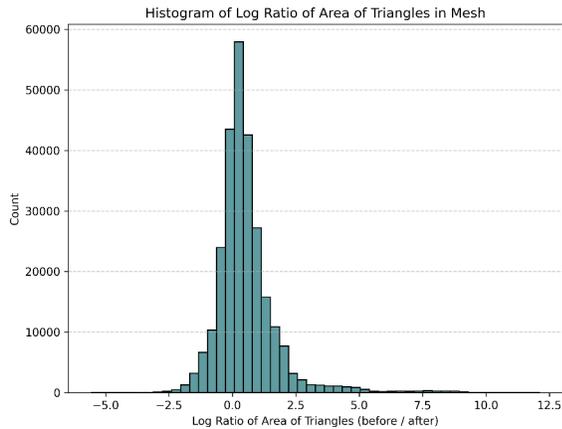
Figure 9: Histogram of log ratio of the areas of triangles in Old man's face mesh.

The results of the experiment are summarized in the table 1, which shows the complexity of the surfaces and the corresponding run time.

We evaluated our proposed method on three facial surfaces. As shown in Fig. 4, the left frame shows the input Alex facial surface; the right frame shows the conformal mapping image on the planar rectangle, it is obvious that the mapping preserves local shapes. Fig. 5 illustrates the initial condition for solving the OT map, the Legendre dual of the initial Brenier potential is $u * (y) = 1/2\|y\|^2$, the projection is the canonical Delaunay triangulation. Fig. 7 shows the final result of the desired Brenier potential and its projected power diagram. Fig 8 illustrates the final clipped Brenier potential and its projected power diagram. Fig. 13 demonstrates the optimization process. The upper left frame is the initial conformal mapping result, the upper right and lower left show the intermediate results during the optimization, the lower right is the final result. We can see that the area surrounding the nose is getting larger and larger. Fig 11 illustrates the quad-meshes based on the intermediate results. We can see the

density of the nose region is getting denser and denser. A clearer depiction of the old man's facial features after utilizing our method is provided in Figure 10.

The proposed mesh adaptation technique was evaluated on three facial models: Alex's face (Figure 11), Sophie's face (Figure 12), and the face of an elderly man (Figure 1). By applying scaling factors to targeted regions, our approach successfully demonstrated its ability to increase quad-mesh density in the central facial areas while reducing it in adjacent regions. This controlled redistribution of density ensures better detail capture in critical areas (such as the nose and eyes) while maintaining computational efficiency by lowering density where fewer details are needed. The results highlight the flexibility of our OT-based framework in achieving adaptive mesh refinement.
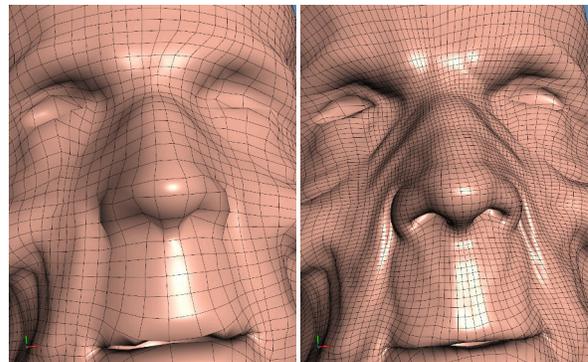
Figure 10: Old man's middle face area zoomed in.

6 Conclusion and Future Work

In this paper, we introduced a novel approach for adjusting the density of quad meshes using Optimal Transport. By leveraging OT's ability to compute mass-preserving mappings, our method ensures precise control over local density distributions while maintaining the structural integrity of the mesh. The proposed framework addresses limitations in traditional remeshing techniques by offering a principled, metric-based solution that adapts mesh densities according to specific criteria. Our results demonstrate that OT provides a robust and flexible tool for selectively altering mesh density, enabling smoother transformations and better alignment with the intended design or modeling goals. This contribution lays the groundwork for new advancements in mesh optimization and geometric modeling by providing a reliable framework for density manipulation.

This method can be applied to enhance the distribution of mesh density by incorporating curvature-based criteria. Our future plan would be to develop an adaptive algorithm that increases mesh density in regions

Model	# Vertices	# Faces	OT runtime	Quad-Mesh generation time
Alex	20956	41908	1.9s	1.2s
Sophie	21046	42088	2.0s	1.5s
Old Man	133604	389703	13.5s	6.8s

Table 1: Complexity of the surface and runtime of OT and quad-mesh generation

with high curvature and reduces it in flatter areas. This extension will improve the efficiency of the mesh by concentrating detail only where necessary, making it particularly useful for applications like finite element analysis, surface parameterization, and high-quality rendering. Additionally, this method can be extended to surfaces with more complex topologies by employing spherical OT and hyperbolic OT for genus zero surfaces and high genus surfaces respectively. These approaches enable the handling of surfaces with complicated topologies and geometries, allowing for effective mesh adaptation and distribution on non-Euclidean geometries.
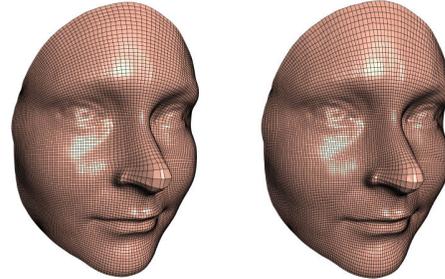
(a) Conformal mapped Sophie face (b) OT mapped Sophie face

(c) OT mapped Sophie's face with increased density on the nose (d) OT mapped Sophie's face with further increased density and decreased density on the forehead and jaw.

Figure 12: Increasing mesh density around nose.

(a) Conformal mapped Alex face (b) OT mapped Alex face
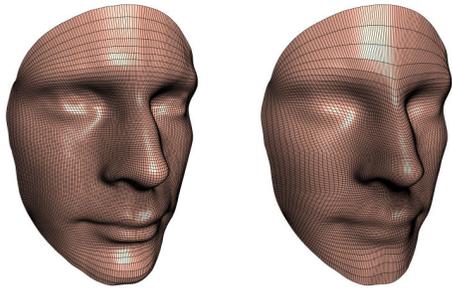
(c) OT mapped Alex's face with increased density on the nose (d) OT mapped Alex's face with further increased density and decreased density on eyes and mouth

Figure 11: Increasing mesh density around nose.
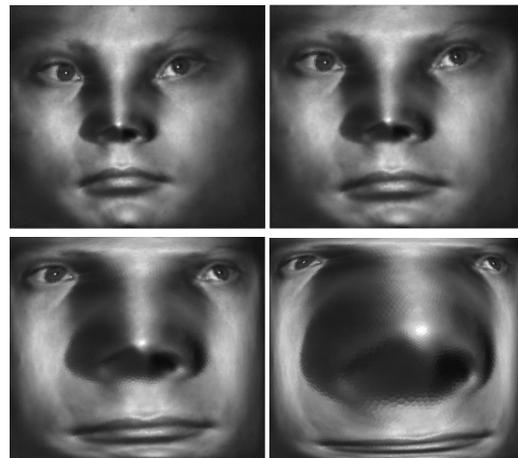
Figure 13: The corresponding OT map of Alex's face.
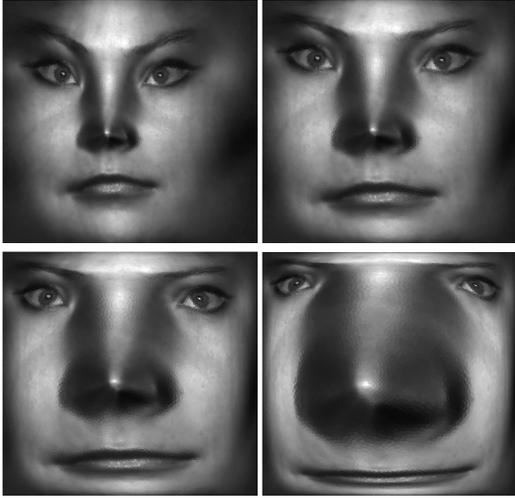
Figure 14: The corresponding OT map of Sophie's face.
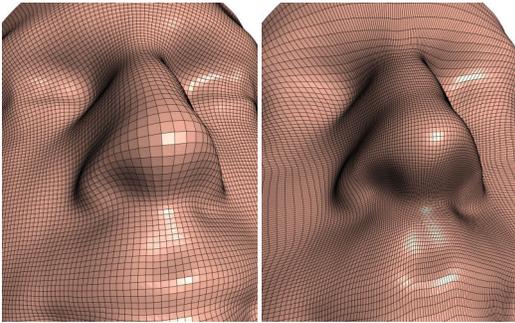
Figure 15: Alex's nose area zoomed in.
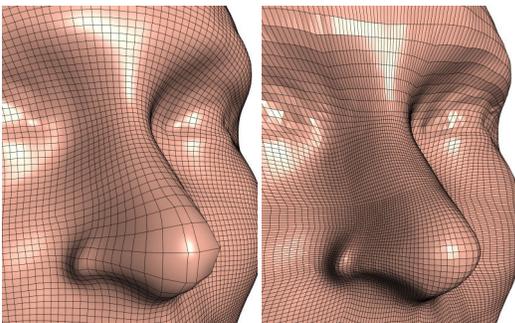
Figure 16: Sophie's nose area zoomed in.

References

[1] D. An, N. Lei, T. Zhao, H. Si, and X. Gu, *A moving mesh adaption method by optimal transport*, Proceedings of International Meshing Roundtable.

[2] D. Bommes, B. Levy, N. Pietroni, E. Puppo, C. Silva, and D. Zorin, *Quad-mesh generation and processing: A survey*, Computer Graphics Forum, 32 (2013).

[3] W. Chen, X. Zheng, J. Ke, N. Lei, Z. Luo, and X. Gu, *Quadrilateral mesh generation ii: Meoromorphic quartic differentials and abel-jacobi condition*, Computer Methods in Applied Mechanics and Engineering, 366.

[4] ——, *Quadrilateral mesh generation i: Metric based method*, Computer Methods in Applied Mechanics and Engineering, 356 (2019), pp. 652–668.

[5] G. Coiffier and E. Corman, *The method of moving frames for surface global parametrization*, ACM Trans. Graph., 42 (2023).

[6] J. Docampo Sánchez and R. Haimes, *A regularization approach for automatic quad mesh generation*, in Proceedings of the 28th International Meshing Roundtable, 2019.

[7] A. Dominitz and A. Tannenbaum, *Texture mapping via optimal mass transport*, IEEE Transactions on Visualization and Computer Graphics, 16 (2010), pp. 419–433.

[8] X. Du, D. M. Kaufman, Q. Zhou, S. Kovalsky, Y. Yan, N. Aigerman, and T. Ju, *Isometric energies for recovering injectivity in constrained mapping*, in SIGGRAPH Asia 2022 Conference Papers, SA '22, New York, NY, USA, 2022, Association for Computing Machinery.

[9] M. Eisenberger, A. Toker, L. Leal-Taixé, and D. Cremers, *Deep shells: Unsupervised shape correspondence with optimal transport*, in Advances in Neural Information Processing Systems (NeurIPS), vol. 33, 2020, pp. 7693–7704.

[10] V. Garanzha, I. Kaporin, L. Kudryavtseva, F. Protais, and D. Sokolov, *In the quest for scale-optimal mappings*, ACM Trans. Graph., 43 (2024).

[11] X. Gu, R. Guo, F. Luo, J. Sun, and T. Wu, *A discrete uniformization theorem for polyhedral surfaces ii*, Journal of Differential Geometry (JDG), 109 (2018), pp. 431–466.

[12] X. Gu, F. Luo, J. Sun, and T. Wu, *A discrete uniformization theorem for polyhedral surfaces i*, Journal of Differential Geometry (JDG), 109 (2018), pp. 223–256.

[13] X. Gu, F. Luo, J. Sun, and S.-T. Yau, *Variational principles for minkowski type problems, discrete optimal transport, and discrete monge-ampere equations*, Asian Journal of Mathematics (AJM), 20 (2016), pp. 383–398.

[14] ——, *Variational principles for minkowski type problems, discrete optimal transport, and discrete monge-ampere equations*, Asian Journal of Mathematics (AJM), 20 (2016), pp. 383–398.

[15] X. Gu, Y. Wang, T. F. Chan, P. M. Thompson, and S.-T. Yau, *Genus zero surface conformal mapping and its application to brain surface mapping*, IEEE Transactions on Medical Imaging (TMI), 23 (2004), pp. 949–958.

[16] X. D. Gu and S.-T. Yau, *Computational Geometry: Algorithms and Applications*, International Press of Boston, Inc., 2008.

[17] L. Kantorovich, *On the translocation of masses*, Journal of Mathematical Science, (2006), pp. 1381–1382.

[18] N. Lei and X. Gu, *Computational Conformal Geometry*, International Press and Higher Education Press, 2021.

[19] N. Lei, Y. Zhu, X. Zheng, H. Si, Z. Luo, and X. Gu, *Why cross fields are not equivalent to quadrilateral meshes*, Computer Methods in Applied Mechanics and Engineering, 417 (2023), p. 116442. A Special Issue in Honor of the Lifetime Achievements of T. J. R. Hughes.

[20] Z. Levi, *Seamless parametrization with cone and partial loop control*, ACM Trans. Graph., 42 (2023).

[21] M. Li, Q. Fang, Z. Zhang, L. Liu, and X.-M. Fu, *Efficient cone singularity construction for conformal parameterizations*, ACM Transactions on Graphics, 42 (2023), pp. 1–13.

[22] N. Nakajima, S. Tokumasu, and Y. Kunitomo, *Feature-based heuristics for finite-element meshing using quadtrees and octrees*, Computer-Aided Design, 24 (1992), pp. 677–690.

[23] S. J. Owen, M. L. Staten, S. A. Canann, and S. Saigal, *Q-morph: An indirect approach to advancing front quad meshing*, International Journal for Numerical Methods in Engineering, 44 (1999), pp. 1317–1340.

[24] C. L. Phillips, *A learning heuristic for space mapping and searching self-organizing systems using adaptive mesh refinement*, Journal of Computational Physics, 272 (2014), pp. 799–813.

[25] N. Pietroni, M. Campen, A. Sheffer, G. Cherchi, D. Bommes, X. Gao, R. Scateni, F. Ledoux, J. Remacle, and M. Livesu, *Hex-mesh generation and processing: A survey*, ACM Trans. Graph., 42 (2022).

[26] M. Rabinovich, R. Poranne, D. Panozzo, and O. Sorkine-Hornung, *Scalable locally injective mappings*, ACM Trans. Graph., 36 (2017).

[27] Z. Shen, J. Feydy, P. Liu, A. H. Curiale, R. S. J. Estepar, R. S. J. Estepar, and M. Niethammer, *Accurate point cloud registration with robust optimal transport*, in Advances in Neural Information Processing Systems (NeurIPS), vol. 34, 2021, pp. 5373–5389.

[28] K. Su, W. Chen, N. Lei, J. Zhang, K. Qian, and X. Gu, *Volume preserving mesh parameterization based on optimal mass transportation*, Computer-Aided Design, 82 (2017), pp. 42 – 56. Isogeometric Design and

Analysis.

[29] E. S. Wise, B. T. Cox, , and B. Treeby, *Mesh density functions based on local bandwidth applied to moving mesh methods*, Communications in Computational Physics, (2017), pp. 1286–1308.

[30] S. Yau, X. Gu, J. Sun, and F. Luo, *Variational principles for minkowski type problems, discrete optimal transport, and discrete monge-ampere equations*, The Asian journal of mathematics, (2016), pp. 383–398.

[31] X. Zhao, Z. Su, X. D. Gu, A. Kaufman, J. Sun, J. Gao, and F. Luo, *Area-preservation mapping using optimal mass transport*, IEEE transactions on visualization and computer graphics, 19 (2013), pp. 2838–2847.

[32] X. Zheng, Y. Zhu, W. Chen, N. Lei, Z. Luo, and X. Gu, *Quadrilateral mesh generation iii : Optimizing singularity configuration based on abel–jacobi theory*, Computer Methods in Applied Mechanics and Engineering, (2021), pp. 114–146.