Eurographics Symposium on Geometry Processing 2012
Eitan Grinspun and Niloy Mitra
(Guest Editors)

Volume 31 (2012), Number 5

Smooth Shape-Aware Functions with Controlled Extrema

Alec Jacobson[1] Tino Weinkauf[2] Olga Sorkine[1]

[1]ETH Zurich, Switzerland [2]MPI Informatik, Saarbrücken, Germany

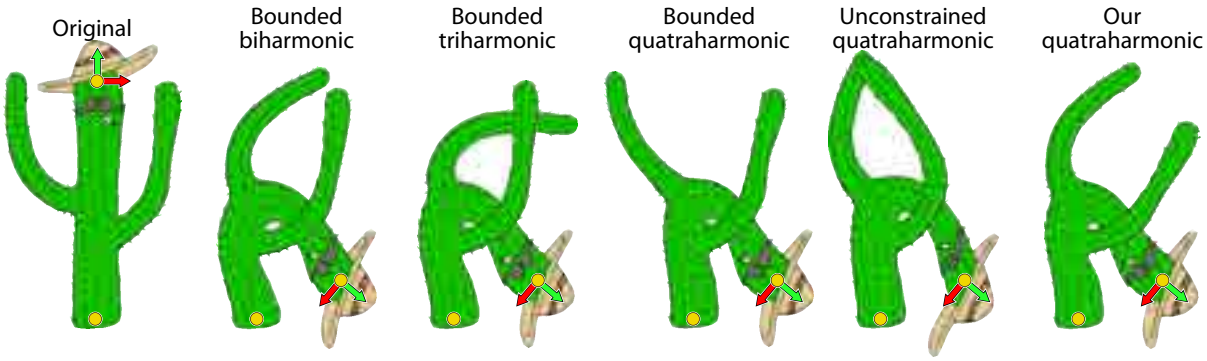

Figure 1: *Shape deformation: Recent works emphasize the importance of bounded control, but simply adding constant bounds to shape-aware smoothness energies of increasing order encourages more and more oscillation. Our framework efficiently optimizes such high-order energies as* $\int_{\mathcal{M}} \|\nabla^4 f\|^2$ *while ensuring against spurious local extrema.*

Abstract

Functions that optimize Laplacian-based energies have become popular in geometry processing, e.g. for shape deformation, smoothing, multiscale kernel construction and interpolation. Minimizers of Dirichlet energies, or solutions of Laplace equations, are harmonic functions that enjoy the maximum principle, ensuring no spurious local extrema in the interior of the solved domain occur. However, these functions are only C^0 *at the constrained points, which often causes smoothness problems. For this reason, many applications optimize higher-order Laplacian energies such as biharmonic or triharmonic. Their minimizers exhibit increasing orders of continuity but lose the maximum principle and show oscillations. In this work, we identify characteristic artifacts caused by spurious local extrema, and provide a framework for minimizing quadratic energies on manifolds while constraining the solution to obey the maximum principle in the solved region. Our framework allows the user to specify locations and values of desired local maxima and minima, while preventing any other local extrema. We demonstrate our method on the smoothness energies corresponding to popular polyharmonic functions and show its usefulness for fast handle-based shape deformation, controllable color diffusion, and topologically-constrained data smoothing.*

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Geometric algorithms, languages, and systems

1. Introduction

Smooth, shape-aware functions have become a cornerstone of geometry processing. They are often obtained by minimizing discrete differential energies, and are used in a wide range of applications, e.g. detail-preserving shape deformation, data interpolation on manifolds, multiscale shape analysis, mesh segmentation and even image processing. Some applications require functions with high-order smoothness or sophisticated control of the boundary conditions, in which case energies involving high-order differential quantities are used.

Energies associated with polyharmonic partial differential equations are particularly popular, since they are relatively easy to discretize on meshes using discrete gradient and Laplace operators [MDSB03], whose properties are well-studied [WMKG07]. Solutions of polyharmonic equations exhibit increasing smoothness at fixed values, but unfortunately also show increasingly wild oscillations that are difficult or

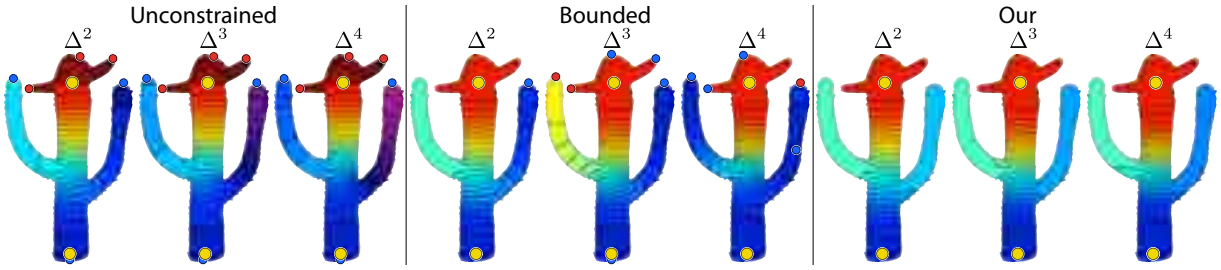

Figure 2: *Various weight functions for the control point in the head of the cactus used in Figure 1, with highlighted local minima (blue dots) and maxima (red dots). Our framework prohibits local extrema during optimization, wrangling the oscillations.*

impossible to control with boundary values alone. Recent work [JBPS11] investigated weight functions for propagating handle transformations in real-time shape deformation; it was shown that minimizing the Laplacian energy with constant bounds produces functions that capture the good qualities of biharmonic functions while avoiding negative values that lead to unintuitive deformation results. The bounds effectively dampen the natural oscillations of the biharmonic solution while maintaining smoothness.

While bounds by definition prevent function values outside the intuitive range, they do not prevent all unintuitive oscillations. In Figure 1, the *Cactus* is deformed using bounded biharmonic functions, but the top of his left arm unintuitively stays put when the control point in the head is moved. This is due to a local minimum in the associated weight function (see Figure 2). Increasing the order of the smoothness energy and keeping the constant bounds only makes the oscillations worse to the detriment of the final deformation. We show how the local extrema introduced by these oscillations occur frequently even if constant bounds are in place.

In this work, we formally define the ideal problem we would like to solve: minimize high-order, shape-aware smoothness energies with guarantees on the locations and values of local extrema. The resulting nonlinear optimization problem turns out to be impractically difficult due to nonlinear inequality constraints. Accordingly, we provide a framework to simplify the constraints in a way that not only ensures we find a feasible solution, but converts the problem into a computationally tractable one.

Our robust optimization allows us to consider smoothness energies associated with higher-order PDEs. We demonstrate the usefulness of guarantees on the absence of new extrema and enabling user control of the placement and values of extrema in the context of weight-based shape deformation, color diffusion and data smoothing.

2. Background

We survey a representative selection of works that utilize smooth, shape-aware functions. Especially relevant to us are works that comment on the oscillatory nature of polyharmonic functions and/or impose topological constraints.

Shape deformation. Smooth, shape-aware influence functions are profusely used for shape deformation. The basic way of deforming the geometry $\mathbf{x}$ of a 2D or 3D shape $\mathcal{M}$ is by combining the propagated influences of affine transformations T_j provided at user-controlled deformation handles:

$$\mathbf{x}_i' = \sum_{j=1}^{H} f_j(\mathbf{x}_i)\, T_j\, \mathbf{x}_i \qquad (1)$$

where $\mathbf{x}_i$ are vertices of the mesh $\mathcal{M}$. An influence function f_j should attain the maximum value of 1 in the shape region that is most affected by the handle j, and decay towards 0 away from that region (reaching exactly 0 on points fully associated with other handles). If f_j has negative values, the deformed shape will unintuitively move in the "opposite direction" to the prescribed T_j. Thus, f_j should be *bounded* within $[0,1]$. Moreover, as we illustrate in this work, if the influence functions have additional local extrema (besides at the constrained handle regions), the deformation behavior is unintuitive as well, causing parts of the shape to "lag behind" or move too fast relative to neighboring parts. We colloquially refer to such behavior as *non-monotonic*.

The formulation in (1) is common in real-time skinning deformations [MTLT88,KCZO08], as the execution of (1) is parallel and extremely fast on the GPU. Several works proposed automatic computation of skinning weights f_j. [WSLG07] uses harmonic functions that are provably monotonic and bounded but have only C^0 smoothness near constrained boundary. [BP07] solves a Poisson equation whose right-hand side is not guaranteed to be monotonic and thus might produce non-monotonic weight functions. Bounded Biharmonic Weights [JBPS11] use biharmonic functions, which are smooth at the handles, and constrain them to be bounded within $[0,1]$. Although Jacobson et al. claim to observe a lack of spurious maxima in their weights, we show a number of examples where this is not the case: especially in shapes with appendages (see, e.g. Figures 1, 3).

Linear variational surface-based deformation methods [BS08] can also be expressed in the above form (1), as shown in the seminal work of Botsch and Kobbelt [BK04]. This is possible when all mesh vertices belonging to one handle are assigned the same affine transformation. The influence functions f_j are then the columns of the inverted system

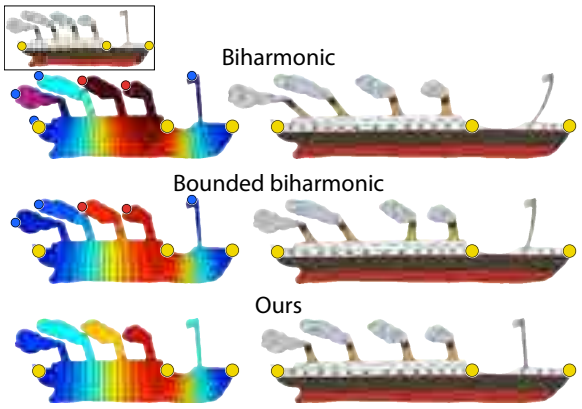

Figure 3: *The* Titanic *(inset) is stretched with 3 control points. Biharmonic and bounded biharmonic weights introduce extrema in the chimneys and flagpole, causing unintuitive response. Our solution (right) is smooth and intuitively deforms the appendages as if rigidly attached to the shape. The weight function for the middle control point is shown for each (left).*

matrix stemming from the variational optimization (referred to as "bases" in [BK04]). Specifically, Botsch and Kobbelt defined families of influence functions that are the solutions of polyharmonic PDEs $\Delta^k f = 0$, providing C^{k-1} continuity at the constrained handles. Jacobson et al. [JTSZ10] refined the approach to allow direct control over boundary derivatives for the biharmonic and the triharmonic cases. Alternatively, triharmonic radial basis functions can be used as weight functions [BK05], replacing sparse large system solves by small and dense ones. In all these cases, the functions are often not monotonic nor even bounded, as discussed in [JBPS11].

Nonlinear deformation methods often employ model reduction based on skinning, such that the transformations T_j become the degrees of freedom in the nonlinear optimization. The weight functions play an important role in the design of the reduced model and greatly affect the final deformation quality. For example, [HSL*06] uses generalized barycentric coordinates and [AFTCO07] employs harmonic functions (using their isolines as reduced-space handles).

When the control handles are vertices of a cage mesh enclosing the shape to be deformed, and the transformations T_j are just translations, the weight functions f_j are called generalized barycentric coordinates (see e.g. [JSW05, JMD*07, HS08, MS10]). Significant attention has been devoted to the boundedness and locality problems [LKCOL07, JMD*07], but we argue the same problems of unintuitive control arise with local extrema. While simple filters may be used to induce boundedness [LS08], such filters will at best downplay spurious local extrema, but will not in general remove them.

A large class of weight functions are constructed to decay with the (geodesic) distance from their handle [SMW06, ZG07]. Deformations with such functions suffer from the "fall-off" effect, characterized by extraneous minima away from the handles [JS11]. These functions are unable to form

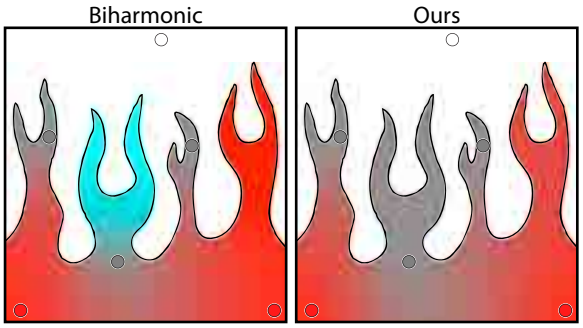

Figure 4: *Colors (specified at the small circles) are diffused using the biharmonic functions of [FSH11], but these are unbounded and extrapolate colors not present in the constraints. Our interpolation explicitly prohibits local extrema and implicitly imposes bounds.*

large regions where one handle's weight function is significantly dominant. These effective plateaus in weight functions are necessary for naturally-looking deformations. In constrast, our method allows dominant regions and we demonstrate that through our choice of smoothness energy we may encourage such behavior, leading to intuitive deformation control.

Boundary value interpolation. Replacing deformed vertex positions $T_j\mathbf{x}$ in Eq. (1) with general values, we obtain an interpolation problem, which has numerous uses in graphics:

$$\mathbf{v}(\mathbf{x}) = \sum_{j=1}^{H} f_j(\mathbf{x})\,\mathbf{v}_j\,. \tag{2}$$

Smoothness and monotonicity of the interpolation basis functions f_j remains important. For example, if $\mathbf{v}_j$'s are RGB colors, one obtains color diffusion, useful in image colorization [LLW04], seamless cloning [PGB03] and vector graphics design [OBW*08]. The mentioned approaches employ harmonic functions f_j; [Geo04] cites possible use of triharmonic and quatraharmonic functions for seamless image cloning, but settles on biharmonic functions. [FSH11] define biharmonic Diffusion Curves to enable diverse control of the color gradients and in particular smoothness at the provided user constraints. Their biharmonic functions can be negative and have prevalent local extrema, leading to unexpected results (see Figure 4). Their proposal to use hard clamping when final color values exceed the displayable range hurts smoothness and does not necessarily avoid local extrema, which are present even when imposing explicit bounds (see Figure 11).

Data smoothing. Scalar data smoothing is often required for subsequent processing, effective visualization and analysis. Polyharmonic RBFs are employed as low-pass filters for smoothing and reconstruction of scattered data, such as range images [CBC*01, CBM*03]; however, no explicit control over the resulting topology, e.g. guarantees of monotonicity, locations and values of extrema, is provided. Carr et al. [CSvdP04] work with the contour tree of the input data set, which enables them to remove small topological features. Gingold and Zorin [GZ06] prevent the formation of new

topological features by controlling the data isocontours while using common iterative filtering methods, such as Laplacian smoothing or anisotropic diffusion.

The two most related approaches to our method in the data smoothing context are [BEHP04] and [WGS10]. Bremer et al. iteratively simplify the Morse-Smale complex of the data and produce a corresponding scalar function after each cancellation step by Laplacian smoothing of the data within each Morse-Smale cell, until no interior critical points are left. Their iterative procedure is time-consuming and produces only C^0 continuity along separatrices. Weinkauf et al. first perform persistence-based simplification of the input Morse-Smale complex, and then fit a C^1 function that adheres to the complex by constrained minimization of a weighted sum of the Laplacian energy and a data term. They optimize each cell subject to nonlinear monotonicity constraints, extracted from a harmonic function computed in the cell. Our use of the maximum principle of harmonic functions is conceptually similar, but we devise a sparse set of linear inequality constraints and convert our energy optimization to a conic programming problem. As a result, our optimization is $\approx 1000\times$ faster (see Section 4). Both methods [BEHP04] and [WGS10] fully constrain the entire Morse-Smale complex, which may be beneficial for visualization, but could be too restrictive in cases where the precise locations or values of saddle points and separatrices are not relevant for the application. Our method constrains locations and values of the extrema, the locations of the saddles, and the combinatorial connectivity between these points in the Morse-Smale complex. The values of saddles and the locations of separatrices may be enforced in our framework but we find it useful to leave these free, allowing for a wider range of feasible solutions.

Mesh and image processing. Smooth, shape-aware (and in particular polyharmonic) functions find many additional uses, such as mesh segmentation [ZT10], design of fair Morse functions [NGH04] or multiscale kernels [Rus11]. Chen et al. [CFL11] used topological constraints to control the number of connected component in an image segmentation. Our framework is general and applicable in any such contexts.

3. Method

Let our domain be the triangle mesh $\mathcal{M}$ and let $\mathcal{N}(i)$ denote the 1-ring neighbors of vertex i. Our goal is to find a piecewise linear function $f : \mathcal{M} \to \mathbb{R}$, which interpolates values at specified minima locations $\mathbf{p}_i, i \in \mathcal{K}_{\min}$ and maxima locations $\mathbf{p}_i, i \in \mathcal{K}_{\max}$, with corresponding fixed values $g_i \in \mathbb{R}$. We also want f to be *monotonic*, i.e., no other extrema in $\mathcal{M}$.

Many functions fulfill those conditions. As often done in data interpolation, we introduce an energy functional $E(f)$ which measures the quality of f for a given application. Laplacian-based energies are often employed as a smoothness regularization term [BK04, JBPS11] and are of the form:

$$E_{L^k}(f) = \int_{\mathcal{M}} \|\nabla^k f\|^2 \text{ for } k = 2,3,\ldots \tag{3}$$

Details of discretization may be found in [JTSZ10].

For applications like data smoothing, we may also introduce a data energy term E_D. In the simplest form, E_D measures in a least-squares sense the deviation from some data function $h : \mathcal{M} \to \mathbb{R}$:

$$E_D(f) = \sum_{i \in \mathcal{M}} \|f_i - h_i\|^2. \tag{4}$$

In general, we consider any energy functional E, but typically we are concerned with combinations of a smoothness term and possibly a data term:

$$E(f) = \gamma_L E_L(f) + \gamma_D E_D(f) \tag{5}$$

where γ_L and γ_D balance the influences of the energies.

3.1. Ideal optimization

We may formulate the ideal problem as an energy minimization with nonlinear, non-differentiable inequality constraints:

$$\arg\min_f E(f) \tag{6}$$

$$\text{s.t.} \quad f_i = g_i \qquad \forall i \in \mathcal{K}_{\min} \cup \mathcal{K}_{\max} \tag{7}$$

$$f_j > f_i \qquad \forall j \in \mathcal{N}(i), \forall i \in \mathcal{K}_{\min} \tag{8}$$

$$f_j < f_i \qquad \forall j \in \mathcal{N}(i), \forall i \in \mathcal{K}_{\max} \tag{9}$$

$$f_i > \min_{j \in \mathcal{N}(i)} f_j \qquad \forall i \notin \mathcal{K}_{\min} \cup \mathcal{K}_{\max} \tag{10}$$

$$f_i < \max_{j \in \mathcal{N}(i)} f_j \qquad \forall i \notin \mathcal{K}_{\min} \cup \mathcal{K}_{\max} \tag{11}$$

The constant equality constraints (7) simply enforce that the values of the known extrema are interpolated. The linear inequality constraints (8) and (9) ensure that the prescribed extremal points are local minima and maxima, respectively. The nonlinear inequality constraints (10) and (11) enforce that all unknown values are greater than their minimum neighbor and smaller than their maximum neighbor.

We assume that the given set of extrema does not contradict the Morse inequalities. For example, a non-constant function on a topological disk must have at least one minimum. We also assume that prescribed extrema are not immediate neighbors and there exists one minimum smaller than all maxima and vice-versa.

Given a quadratic energy E, this optimization problem would be simple to solve if not for the nonlinear inequality constraints. The other constraints are linear and at worst produce a quadratic programming problem. Trying to optimize the ideal problem directly with commercial "black-box" nonlinear optimization software [MAT12] shows discouraging convergence. Often feasible solutions are not found, and even if the solver does converge, performance renders this option useless for most applications. For example, on a 16-vertex mesh, the optimization takes 3 seconds to converge, and only does so when provided with a close and feasible initial guess.

Many nonlinear optimization methods allow a "constraint violation tolerance" parameter. However, our topological constraints are highly sensitive. Even the slightest violation can allow oscillations preferred by the unconstrained energy, producing potentially unbounded spurious hills and valleys.

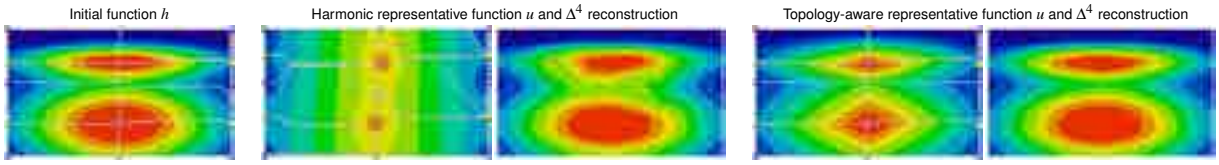

Figure 5: *Two different representative functions are compared for the application of smoothing an initially given function. The harmonic function (middle) is visualized together with its Morse-Smale complex, which is clearly different to the original topology. The constraints derived from this fight against the data term during the reconstruction, which leads to a poor result. The topology-aware representative function (right) leads to good reconstruction results, since it respects the original topology.*

3.2. Constraint simplification

In light of our inability to solve the ideal optimization problem in (6) in a reasonable amount of time, we propose a method for simplifying the nonlinear inequality constraints (10) and (11) into a larger set of *linear* inequality constraints. Suppose we have a *representative* function $u : \mathcal{M} \to \mathbb{R}$ which satisfies constraints (8), (9), (10), and (11). We replace the nonlinear inequality constraints (10) and (11) with linear inequality constraints requiring the *direction* (but not magnitude) of ∇f be aligned with $\nabla u / \|\nabla u\|$. In essence we enforce the *monotonicity* or, loosely, the topology of u onto our optimized solution f. As long as we choose u intelligently and construct our linear constraints from $\nabla u / \|\nabla u\|$ carefully, the optimization in (6) becomes convex and thus efficiently solvable, and finds an acceptable solution.

Given a monotonic representative function u as described above, we reduce the optimization to:

$$\arg\min_f \; E(f) \tag{12}$$

$$\text{s.t.} \quad f_i = g_i \qquad \forall i \in \mathcal{K}_{\min} \cup \mathcal{K}_{\max} \tag{13}$$

$$(f_i - f_j)(u_i - u_j) > 0 \qquad \forall (i,j) \in \mathcal{E} \tag{14}$$

where $\mathcal{E}$ is a subset of the edges of the domain $\mathcal{M}$. The constraints (14) require that the direction of decrease across every edge of $\mathcal{E}$ in our optimal solution f should match that of the known function u. If $\mathcal{E}$ contains all edges, this is enforced everywhere and the choice of u greatly restricts the shape of f. This is useful if we have high confidence in the representativeness of u (e.g., when it is derived from existing input data). In other situations we only want to capture the monotonicity of u, hence we would like the smallest possible edge set $\mathcal{E}$. To guarantee that our constraints prohibit local extrema, $\mathcal{E}$ must include for each vertex $i \in \mathcal{M} \setminus (\mathcal{K}_{\min} \cup \mathcal{K}_{\max})$ *at least* one edge (i,j) where $u_j < u_i$ and another where $u_j > u_i$. This guarantees that each vertex value f_i in our solution will have one neighbor $f_j < f_i$ and another $f_j > f_i$.

3.3. Choice of representative function

We provide two methods for constructing a valid representative function u satisfying (8), (9), (10), and (11). The first method shows that such a function always exists, ensuring a feasible final solution. The second method takes advantage of situations when initial data is present.

No initial data. Consider a situation where no data is available besides the domain $\mathcal{M}$ and the locations and values of extrema in $\mathcal{K}_{\max}$ and $\mathcal{K}_{\min}$. This is useful, for example, in the context of shape deformation and color interpolation. Taking advantage of the strong maximum principle of harmonic functions, we can always construct a valid representative function u by solving the following Dirichlet problem:

$$\arg\min_u \int_{\mathcal{M}} \|\nabla u\|^2 \tag{15}$$

$$\text{s.t.} \quad u|_{\mathcal{K}_{\min}} = 0 \tag{16}$$

$$u|_{\mathcal{K}_{\max}} = 1. \tag{17}$$

Minimizers of the Dirichlet energy are harmonic functions. Their maximum principle guarantees that, when choosing the Dirichlet boundary conditions $u|_{\mathcal{K}_{\min}} = 0$ and $u|_{\mathcal{K}_{\max}} = 1$, the locations in $\mathcal{K}_{\min}$ and $\mathcal{K}_{\max}$ become minima and maxima, respectively. Thanks to the uniqueness of harmonic functions, u contains no other extrema inside $\mathcal{M}$. This fulfills the necessary conditions for u to be a valid representative function.

Initial data. In other situations, e.g. scalar field smoothing, our input will contain some initial data function h and the objective is to remove some of its extrema while keeping others. This is useful in the context of topologically-constrained smoothing [WGS10], where the Morse-Smale complex of h is simplified based on Forman's discrete Morse theory [For98] such that only the critical points above a user-defined persistence [ELZ02] threshold remain. To reconstruct a smooth function f based on the remaining topology and as close as possible to the initial data function h, we set $\mathcal{K}_{\min}$ and $\mathcal{K}_{\max}$ to contain the minima/maxima of the simplified Morse-Smale complex and construct the representative function u in the same fashion as [WGS10] builds its "preview" function: the critical points are fixed to their original values $u(\mathbf{p}_i) = h(\mathbf{p}_i)$, and all vertices on each separatrix are fixed to a linearly interpolated value between its end points, i.e., a saddle and an extremum. All fixed vertices serve as Dirichlet boundary conditions for solving $\Delta u = 0$ on the domain. The solution u is a harmonic function that has no interior critical points in the domain: since the boundary conditions are monotonic, we obtain a valid scalar field that obeys the prescribed topology.

Note that the solver of [WGS10] required that the values along the separatrices remain fixed. This mandated smoothing

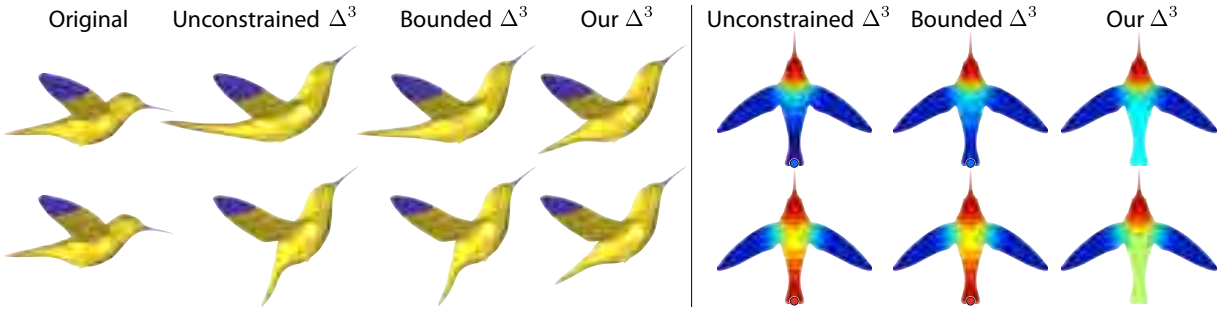

Figure 6: *The* Hummingbird *in her rest pose is deformed using the unconstrained, bounded and our triharmonic weights (top row, left). For this handle configuration, the unconstrained and bounded weights of the nose introduce a local minimum in the tail, whereas ours stay monotonic (top row, right). The bottom row shows the chaotic nature of the unconstrained and bounded weights' oscillations. Slightly larger wing handles reverse the oscillations, now producing a local maximum in the tail. We constrain the solution to be monotonic and thus avoid such chaotic oscillations.*

the locations and values of the separatrices before building another harmonic function and finally optimizing the interiors of the Morse cells. Our framework does not require fixing separatrix values and may optimize the entire domain at once.

Figure 5 compares this topology-aware representative function to the harmonic function created by (15)–(17). The latter places the central saddle point in the middle between the two peaks, thereby disregarding the size and shape of the peaks in the original function. In this example, this creates areas where the gradients of the harmonic and the original function are perpendicular to each other (visible in the isolines). This leads to a poor reconstruction. The topology-aware representative function is built from the original topology, leading to a favorable reconstruction.

Finally, with a valid representative function we may take a minimally sufficient edge set $\mathcal{E}$. For each vertex i in $\mathcal{M}$, we include the edges $\{i,j\}$ and $\{i,k\}$ where u_j and u_k are the smallest and greatest of the neighbors of i. To ensure the topological conditions of the user constraints are met, we add all edges incident on any i in $\mathcal{K}_{\min}$ or $\mathcal{K}_{\max}$.

3.4. Implementation

One could solve (12) with any sparse quadratic programming solver, but we saw major performance improvements ($\approx$100$\times$) when converting our problem to conic programming and using MOSEK [AA00], a sparse, conic programming solver. See Appendix for details of this conversion. For 2D image deformation and color diffusion we use Triangle [She96] to triangulate the interior of the shape outline.

When solving for shape-deformation or interpolation weight functions in (1) and (2), we may append additional linear equality constraints to ensure weights across handles sum to one for every vertex in $\mathcal{M}$. This would tether the optimizations of the weight functions together into one larger problem. However, we observe the same behavior of our weight functions as in [JBPS11]: dropping the partition of unity constraints and normalizing *post-hoc* has little effect on the final weights and thus also the deformation. Thus, in all our experiments, we compute each weight function independently and normalize *post-hoc*. We report the average times for computing each function in Table 1.

	$\lvert\mathcal{M}\rvert$	k	Time/f_j	Total Time
Cactus	2403	2	0.1246	0.2493
Cactus	2403	3	1.3135	2.6271
Cactus	2403	4	0.2324	0.4649
Colored J	8229	2	0.2306	0.9225
Hummingbird	14636	3	86.393	259.18
Dino	28136	2	2.4564	7.3693
Combustor	29021	2	5.1707	5.1707
Beetle	38656	2	6.0263	6.0263

Table 1: *Statistics for various examples in the paper.* $\lvert\mathcal{M}\rvert$ *is the number of triangles in the discretized domain,* k *is the order of the corresponding polyharmonic operator,* Time/f_j *is the average optimization time per function in seconds, and* Total Time *is the total optimization time.*

It is worth noting that this normalization technically invalidates any guarantees of monotonicity in the final functions, but we argue that the artifacts arising from non-monotonicity are involved with low-frequency oscillations present in the original functions before normalization. In any case, our contribution is a general framework which easily incorporates creating guaranteed monotonic functions with partition of unity constraints if so desired, just at additional precomputation costs. One would simply add the appropriate linear equality constraints and compute all functions simultaneously as also described in [JBPS11].

4. Experiments and Results

We experimented with quadratic energies corresponding to biharmonic, triharmonic and quatraharmonic equations with and without least-squares data energy terms. We tested our method on an iMac Intel Core i7 3.4GHz computer with

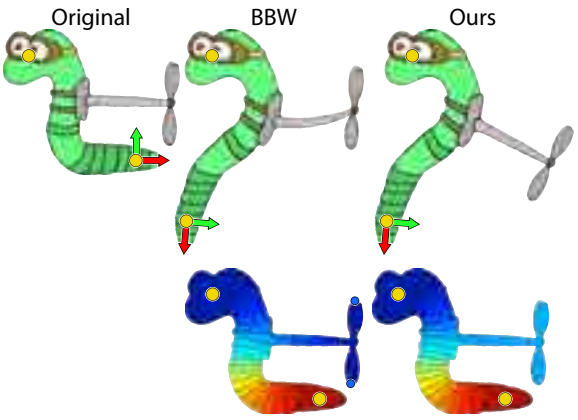

Figure 7: *The* Propeller Worm *in its rest pose (left) is deformed using bounded biharmonic weights (BBW) of [JBPS11] (middle). A local maximum leaves its propeller behind when its tail is bent, also causing the rod to unintuitively squish. Our solution (right) attaches the propeller to the body, giving it near constant weights (visualized below).*

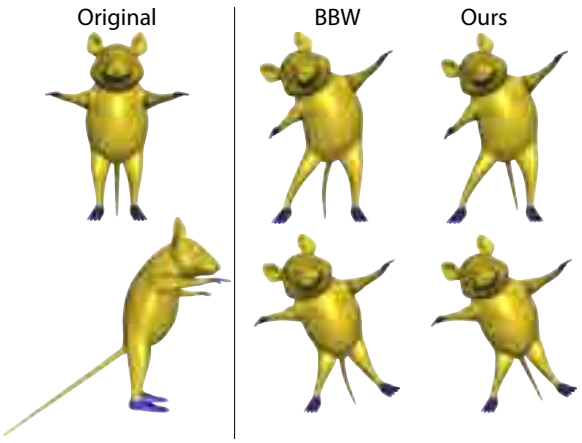

Figure 8: *The* Mouse *in its rest pose (left) is deformed using the biharmonic weights (BBW) of [JBPS11] (middle), which each have an extrema in the tail causing it to wiggle when the handles are deformed. Our weights have no spurious extrema and keep the tail stiff.*

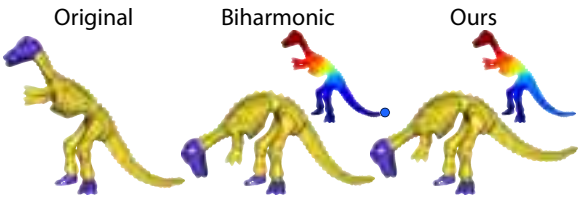

Figure 9: *The* Dino *in its rest pose (left) is deformed using the biharmonic weights of [BK04] (middle). In this example the weights are between* $[0,1]$ *but a local minimum (blue dot) leaves the tail connected to the feet, giving the impression that it is glued to the ground when bending. Our solution finds more intuitive, monotonic weights.*

16GB memory. The computation time measurements of our code are reported in Table 1. The optimization time required depends heavily on the order of the polyharmonic operator in play. For the Laplacian energy, $k = 2$, we show timings on the same order of magnitude as [JBPS11] who solve a simpler quadratic programming problem with only constant bounds. For $k = 3$, our conversion to conic programming results in a rectangular coefficients matrix, which is in turn not as efficiently optimized. Somewhat surprisingly, increasing the order to the next even power, $k = 4$, returns to much faster computation time: the square root of discrete quatraharmonic operator is again square.

We use dual quaternion skinning [KCZO08] to deform the *Cactus* in Figure 1. This avoids distracting shrinkage artifacts present in linear blend skinning. This example demonstrates that increasing the smoothness operator under our framework does not result in wilder oscillations.

We have found that unconstrained and bounded polyharmonic solutions often struggle with long appendages, placing extrema in the propeller in Figure 7 and the chimneys in Figure 3. During exploration of the deformation, local extrema in these appendages are immediately apparent and distracting during interaction. The tail of the *Dino* in Figure 9 feels as if it is glued to the ground when the head is transformed. The tail of the *Mouse* in Figure 8 wiggles when deforming the hands and feet, rather than staying stiff as one might expect. Applying scaling exaggerates the unintuitive nature of local extrema in weight functions: the geodesically distant sousaphone bell is sheared when scaling the head in Figure 12. Whether the additional local extrema are maxima or minima seems largely unpredictable; the oscillations may flip due to slight changes in the boundary definition, see Figure 6.

In contrast, the monotonic weight functions resulting from our method allow intuitive deformations in all these cases. Please refer to the supplemental video to get a better impression of the deformation behavior.

Long features are also difficult for previous methods when blending colors. Oscillations common to biharmonic functions used by [FSH11] lead to interpolation weights outside of $[0,1]$ and may extrapolate colors not present in the user's constraints: red - grey = blue in Figure 4. Placing bounds on these weight functions helps, but local extrema are still present, having the effect that colors fade out and then suddenly reappear somewhere else in the domain (see Figure 11).

In Figure 10, we smooth exchange rate data which contains a pronounced, sharp spike at the global maximum. To ensure that the C^0 nature of this spike is not smoothed away, we take advantage of the fact that the parameterized blends of different polyharmonic operators in [BK04] may be reexpressed as discrete quadratic energies. We may then optimize using our method, such that only specified extrema are present in the result. In this example, we choose our blend parameters (λ in

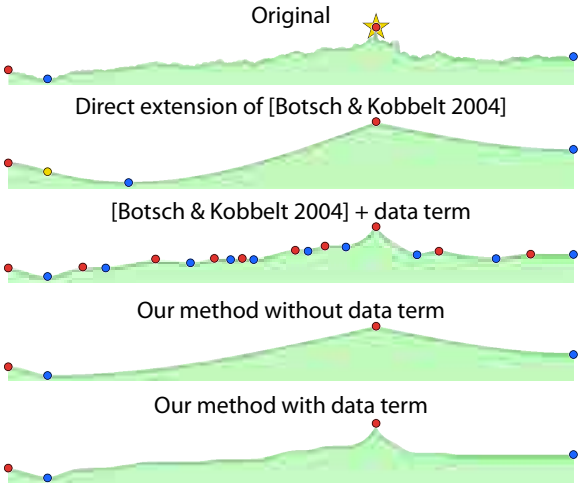

Figure 10: *Top to bottom: Currency exchange data with hundreds of extrema including a sharp, global maximum (star). Blended polyharmonic energies of [BK04] can reproduce the spike, constraining only the values of the global minimum, global maximum, and endpoint values. But this produces new extrema, changing the global min (from yellow dot). Adding a data term introduces even more extrema. Our formulation prohibits new extrema, and with a data term provides a smooth, monotonic representation of the data.*

Section 3 of [BK04]) such that we create a C^0 at the global maximum and C^1 elsewhere.

Smoothing geodesic distance fields is often a tricky balance between achieving desired smoothness and loosing control of the linear spacing of isolines and placement and values of maxima (farthest points). Our framework reconstructs a smoothed geodesic distance field which has the original maxima and no others (see Figure 13). Smoothing noisy data similarly requires care, or original features may be lost while new ones are introduced during smoothing. In Figure 14, we reproduce the topology-based smoothing results of [WGS10] (compare to Figure 10c in their paper) with an optimization time that, for this example, is 1000× faster.

5. Limitations and Future Work

Due to the large number of linear inequality constraints, computation can still be expensive, compared to solving a linear system (e.g. [BK04, FSH11]), or to optimizing a quadratic energy with constant bounds [JBPS11]. But it is important to keep in mind that the ideal problem in Eq. (6) is far more difficult. We would like to investigate other optimization methods that could take advantage of a warm start or other constraint simplification possibilities.

We take advantage of the maximum principle of harmonic functions, but it is well known that obtuse angles may cause weights in a cotangent Laplacian to be negative, thus nullifying the guarantee of the maximum principle [WMKG07]. We sometimes observe this problem *locally* in construction

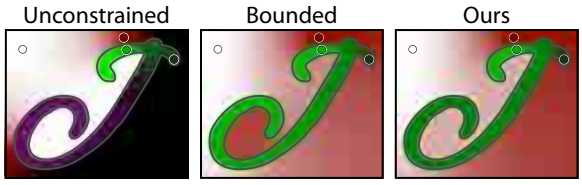

Figure 11: *Colors are diffused using the unconstrained biharmonic functions of [FSH11], resulting in extrapolated (purple) and clipped (black) regions. Placing constant bounds helps, but oscillations with local extrema are still visible. Our solution provides a smooth diffusion without wild oscillations.*

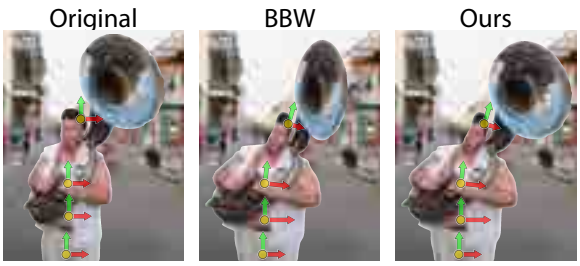

Figure 12: *The* Sousaphonist *in his rest pose (left) is deformed using the bounded biharmonic weights (BBW) of [JBPS11] (middle). Due to local extrema the horn's bell gets an uneven deformation, whereas our local-extrema-free deformation maintains its shape.*

of our representative functions u. While this could potentially lead to actually enforcing local extrema in our final solution, we observe that globally u captures the correct gradient information we need, and problems due to poor discretization can often be safely ignored. Of course, another option is to use a discrete Laplacian with positive weights, but this may come at the cost of other convenient properties [WMKG07].

Finally, our per-edge, linear inequality constraints are inherently discretization dependent. For data smoothing this means, an asymmetric meshing combined with a strong data term may produce unintuitively asymmetric reconstructions. Without a data term (e.g. for deformation or color interpolation) this appears to be a non-issue: the constrained optimization subspace is still quite large.

6. Conclusion

We have shown a framework for constructing smooth, shape-aware functions on 2D and 3D surfaces with guarantees on the placement and values of extrema. We also highlight the typical problematic situations for which our method succeeds over previous work. We believe our work will help promote the continued study of topological constraints in connection with geometry processing applications like deformation and interpolation on manifolds.

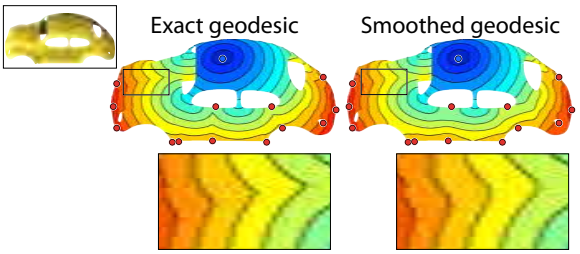

Figure 13: *Sharp cusps appear in exact geodesic distances (left) computed for a point on the top of the* Beetle *(inset). Our framework smooths this field while guaranteeing that all maxima maintain their location and value (right). No new extrema are created.*

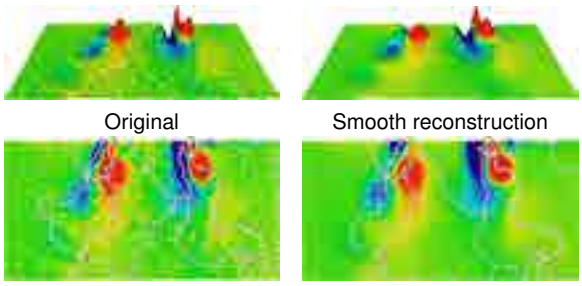

Figure 14: *By adding a data energy term and using a topology-aware representative function, we may use our framework to smooth noisy data. Left: vorticity magnitude derived from an optically measured flow at the outlet of a combustion chamber, with thousands of local extrema. Using persistence-based simplification, we isolate the most important extrema: 9 minima, 4 maxima (blue and red dots). We then smooth the data guaranteeing that these and only these extrema occur in the solution (right). Data courtesy of A. Lacarelle (TU Berlin) [LFG*09].*

Acknowledgement

We thank Kenshi Takayama for his valuable feedback. This work was supported in part by an SNF award 200021_137879 and by a gift from Adobe Systems.

References

[AA00] ANDERSEN E. D., ANDERSEN K. D.: The MOSEK interior point optimizer for linear programming: an implementation of the homogeneous algorithm. In *High Performance Optimization*. Kluwer Academic Publishers, 2000, pp. 197–232. 6, 10

[AFTCO07] AU O. K.-C., FU H., TAI C.-L., COHEN-OR D.: Handle-aware isolines for scalable shape editing. *ACM Trans. Graph. 26*, 3 (2007), 83. 3

[BEHP04] BREMER P.-T., EDELSBRUNNER H., HAMANN B., PASCUCCI V.: A topological hierarchy for functions on triangulated surfaces. *IEEE TVCG 10*, 4 (2004), 385 – 396. 4

[BK04] BOTSCH M., KOBBELT L.: An intuitive framework for real-time freeform modeling. *ACM Trans. Graph. 23*, 3 (2004), 630–634. 2, 3, 4, 7, 8

[BK05] BOTSCH M., KOBBELT L.: Real-time shape editing using radial basis functions. *Comput. Graph. Forum 24*, 3 (2005), 611–621. 3

[BKP*10] BOTSCH M., KOBBELT L., PAULY M., ALLIEZ P., LÉVY B.: *Polygon Mesh Processing*. AK Peters, 2010. 10

[BP07] BARAN I., POPOVIĆ J.: Automatic rigging and animation of 3D characters. *ACM Trans. Graph. 26*, 3 (2007), 72:1–72:8. 2

[BS08] BOTSCH M., SORKINE O.: On linear variational surface deformation methods. *IEEE Transactions on Visualization and Computer Graphics 14*, 1 (2008), 213–230. 2

[CBC*01] CARR J., BEATSON R., CHERRIE J., MITCHELL T., FRIGHT W., MCCALLUM B., EVANS T.: Reconstruction and representation of 3D objects with radial basis functions. In *Proc. ACM SIGGRAPH* (2001), pp. 67–76. 3

[CBM*03] CARR J. C., BEATSON R. K., MCCALLUM B. C., FRIGHT W. R., MCLENNAN T. J., MITCHELL T. J.: Smooth surface reconstruction from noisy range data. In *Proc. ACM GRAPHITE* (2003), pp. 119–127. 3

[CFL11] CHEN C., FREEDMAN D., LAMPERT C. H.: Enforcing topological constraints in random field image segmentation. In *Proc. CVPR* (2011), pp. 2089–2096. 4

[CSvdP04] CARR H., SNOEYINK J., VAN DE PANNE M.: Simplifying flexible isosurfaces using local geometric measures. In *Proc. Visualization* (2004), pp. 497–504. 3

[ELZ02] EDELSBRUNNER H., LETSCHER D., ZOMORODIAN A.: Topological persistence and simplification. *DCG 28*, 4 (2002), 511 – 533. 5

[For98] FORMAN R.: Morse theory for cell-complexes. *Adv. in Math. 134*, 1 (1998), 90–145. 5

[FSH11] FINCH M., SNYDER J., HOPPE H.: Freeform vector graphics with controlled thin-plate splines. *ACM Trans. Graph. 30*, 6 (2011), 166:1–166:10. 3, 7, 8

[Geo04] GEORGIEV T.: Photoshop healing brush: a tool for seamless cloning. In *Proc. ECCV ACV Workshop* (2004), pp. 1–8. 3

[GZ06] GINGOLD Y. I., ZORIN D.: Controlled-topology filtering. In *Proc. SPM* (2006), pp. 53–61. 3

[HS08] HORMANN K., SUKUMAR N.: Maximum entropy coordinates for arbitrary polytopes. *Comput. Graph. Forum 27*, 5 (2008), 1513–1520. 3

[HSL*06] HUANG J., SHI X., LIU X., ZHOU K., WEI L.-Y., TENG S.-H., BAO H., GUO B., SHUM H.-Y.: Subspace gradient domain mesh deformation. *ACM Trans. Graph. 25*, 3 (2006), 1126–1134. 3

[JBPS11] JACOBSON A., BARAN I., POPOVIĆ J., SORKINE O.: Bounded biharmonic weights for real-time deformation. *ACM Trans. Graph. 30*, 4 (2011), 78:1–78:8. 2, 3, 4, 6, 7, 8

[JMD*07] JOSHI P., MEYER M., DEROSE T., GREEN B., SANOCKI T.: Harmonic coordinates for character articulation. *ACM Trans. Graph. 26*, 3 (2007), 71. 3

[JS11] JACOBSON A., SORKINE O.: Stretchable and twistable bones for skeletal shape deformation. *ACM Trans. Graph. 30*, 6 (2011), 165:1–165:8. 3

[JSW05] JU T., SCHAEFER S., WARREN J.: Mean value coordinates for closed triangular meshes. *ACM Trans. Graph. 24*, 3 (2005), 561–566. 3

[JTSZ10] JACOBSON A., TOSUN E., SORKINE O., ZORIN D.: Mixed finite elements for variational surface modeling. In *Proc. SGP* (2010), pp. 1565–1574. 3, 4

[KCZO08] KAVAN L., COLLINS S., ZARA J., O'SULLIVAN C.: Geometric skinning with approximate dual quaternion blending. *ACM Trans. Graph. 27*, 4 (2008), 105:1–105:23. 2, 7

[LFG*09] LACARELLE A., FAUSTMANN T., GREENBLATT D., PASCHEREIT C., LEHMANN O., LUCHTENBURG D., NOACK B.: Spatiotemporal Characterization of a Conical Swirler Flow Field Under Strong Forcing. *Journal of Engineering for Gas Turbines and Power 131* (2009), 031504. 9

[LKCOL07] LIPMAN Y., KOPF J., COHEN-OR D., LEVIN D.: GPU-assisted positive mean value coordinates for mesh deformations. In *Proc. SGP* (2007), pp. 117–124. 3

[LLW04] LEVIN A., LISCHINSKI D., WEISS Y.: Colorization using optimization. *ACM Trans. Graph. 23* (2004), 689–694. 3

[LS08] LANGER T., SEIDEL H.-P.: Higher order barycentric coordinates. *Comput. Graph. Forum 27*, 2 (2008), 459–466. 3

[MAT12] MATLAB: *7.13.0.564 (R2011b)*. The MathWorks Inc., Natick, Massachusetts, 2012. 4

[MDSB03] MEYER M., DESBRUN M., SCHRÖDER P., BARR A. H.: Discrete differential-geometry operators for triangulated 2-manifolds. In *Visualization and Mathematics III*. Springer-Verlag, 2003, pp. 35–57. 1, 10

[MS10] MANSON J., SCHAEFER S.: Moving least squares coordinates. In *Proc. SGP* (2010), pp. 1517–1524. 3

[MTLT88] MAGNENAT-THALMANN N., LAPERRIÈRE R., THALMANN D.: Joint-dependent local deformations for hand animation and object grasping. In *Graphics Interface* (1988), pp. 26–33. 2

[NGH04] NI X., GARLAND M., HART J. C.: Fair morse functions for extracting the topological structure of a surface mesh. *ACM Trans. Graph. 23*, 3 (2004), 613–622. 4

[OBW*08] ORZAN A., BOUSSEAU A., WINNEMÖLLER H., BARLA P., THOLLOT J., SALESIN D.: Diffusion curves: a vector representation for smooth-shaded images. *ACM Trans. Graph. 27*, 3 (2008), 92:1–92:8. 3

[PGB03] PÉREZ P., GANGNET M., BLAKE A.: Poisson image editing. *ACM Trans. Graph. 22*, 3 (2003), 313–318. 3

[Rus11] RUSTAMOV R. M.: Multiscale biharmonic kernels. In *Proc. SGP* (2011), pp. 1521–1531. 4

[She96] SHEWCHUK J. R.: Triangle: Engineering a 2D quality mesh generator and delaunay triangulator. In *Applied Computational Geometry: Towards Geometric Engineering*, vol. 1148 of *Lecture Notes in Computer Science*. Springer-Verlag, 1996, pp. 203–222. 6

[SMW06] SCHAEFER S., MCPHAIL T., WARREN J.: Image deformation using moving least squares. *ACM Trans. Graph. 25*, 3 (2006), 533–540. 3

[WGS10] WEINKAUF T., GINGOLD Y., SORKINE O.: Topology-based smoothing of 2D scalar fields with C^1-continuity. *Comput. Graph. Forum (proc. EuroVis) 29*, 3 (2010), 1221–1230. 4, 5, 8

[WMKG07] WARDETZKY M., MATHUR S., KÄLBERER F., GRINSPUN E.: Discrete Laplace operators: no free lunch. In *Proc. SGP* (2007), pp. 33–37. 1, 8

[WSLG07] WEBER O., SORKINE O., LIPMAN Y., GOTSMAN C.: Context-aware skeletal shape deformation. *Comput. Graph. Forum 26*, 3 (2007), 265–273. 2

[ZG07] ZHU Y., GORTLER S. J.: *3D deformation using moving least squares*. Tech. Rep. Tr-10-07, Harvard University, Cambridge, MA, 2007. 3

[ZT10] ZHENG Y., TAI C.-L.: Mesh decomposition with cross-boundary brushes. *Comput. Graph. Forum 29*, 2 (2010), 527–535. 4

Appendix

We use MOSEK [AA00] to efficiently solve sparse, quadratic progamming problems. Its documentation strongly recommends converting convex quadratic energy minimization with linear inequality constraints, like Eq. (12), into linear energy minimization with conic constraints. We found this to be especially advantageous for our problem. Without loss of generality we assume our energy is of the form: $E(f) = \int_{\mathcal{M}} (\nabla^k f)^2$, which can be discretized as

$$E(\mathbf{f}) = \frac{1}{2}\mathbf{f}^\mathsf{T}\left(\mathbf{L}\mathbf{M}^{-1}\right)^{k-1}\mathbf{L}\mathbf{f} \tag{18}$$

where $\mathbf{L}$ and $\mathbf{M}$ are the familiar cotangent Laplacian and normalized, diagonalized mass matrix, respectively [MDSB03]. We may write $\mathbf{L} = \mathbf{G}^\mathsf{T}\mathbf{A}\mathbf{G}$ where $\mathbf{G}$ is the per-element gradient operator and $\mathbf{A}$ has triangle areas repeated along the diagonal [BKP*10]. For odd k, Eq. (18) becomes:

$$E(\mathbf{f}) = \frac{1}{2}\|\sqrt{\mathbf{A}}\mathbf{G}\left(\mathbf{M}^{-1}\mathbf{L}\right)^{\frac{k-1}{2}}\mathbf{f}\|^2 \tag{19}$$

and for even k:

$$E(\mathbf{f}) = \frac{1}{2}\|\sqrt{\mathbf{M}}^{-1}\left(\mathbf{L}\mathbf{M}^{-1}\right)^{\frac{k}{2}-1}\mathbf{L}\mathbf{f}\|^2. \tag{20}$$

This allows us to write $E(f) = \frac{1}{2}\|\mathbf{F}\mathbf{f}\|^2$ and Eq. (12) becomes:

$$\begin{aligned} \underset{\mathbf{f}}{\text{minimize}} \quad & \frac{1}{2}\|\mathbf{F}\mathbf{f}\|^2 + \mathbf{c}^\mathsf{T}\mathbf{f} + \text{const} \\ \text{subject to} \quad & \mathbf{A}_{leq}^\mathsf{T}\mathbf{f} \le \mathbf{b}_{leq}, \\ & \mathbf{f} \le \mathbf{u}_f, \;\; \mathbf{f} \ge \mathbf{l}_f \end{aligned} \tag{21}$$

where, for the sake of generality, we include linear terms $\mathbf{c}^\mathsf{T}\mathbf{f}$ and a constant term (e.g. arising from data energy terms). The matrix $\mathbf{A}_{leq}$ contains the coefficients of the linear inequality constraints, and $\mathbf{b}_{leq}$ is a vector of zeros. The constant upper and lower bounds $\mathbf{u}_f$ and $\mathbf{l}_f$ are set to the user-defined global maximum and minimum values and aid the optimization.

To convert this to a conic problem, we first introduce a vector of auxiliary variables $\mathbf{t}$ and rewrite as:

$$\begin{aligned} \underset{\mathbf{f},\mathbf{t}}{\text{minimize}} \quad & \frac{1}{2}\|\mathbf{t}\|^2 + \mathbf{c}^\mathsf{T}\mathbf{f} + \text{const} \\ \text{subject to} \quad & \mathbf{F}\mathbf{f} - \mathbf{t} = 0, \\ & \mathbf{A}_{leq}^\mathsf{T}\mathbf{f} \le \mathbf{b}_{leq}, \;\; \mathbf{f} \le \mathbf{u}_f, \;\; \mathbf{f} \ge \mathbf{l}_f\,. \end{aligned} \tag{22}$$

Using a scalar variable v we convert into conic form:

$$\begin{aligned} \underset{\mathbf{f},\mathbf{t},v}{\text{minimize}} \quad & v + \mathbf{c}^\mathsf{T}\mathbf{f} + \text{const} \\ \text{subject to} \quad & \mathbf{F}\mathbf{f} - \mathbf{t} = 0, \;\; \mathbf{A}_{leq}^\mathsf{T}\mathbf{f} \le \mathbf{b}_{leq}, \;\; \mathbf{f} \le \mathbf{u}_f, \;\; \mathbf{f} \ge \mathbf{l}_f, \\ & 2v \ge \sum_i t_i^2 \end{aligned}$$

where the inequality constraint on v forces its value to be inside the cone described by the coordinates of $\mathbf{t}$.

Putting all variables in a column vector, we can write this in matrix form, as we supply it to the solver:

$$\begin{aligned} \underset{\begin{bmatrix}\mathbf{f}^\mathsf{T} & \mathbf{t}^\mathsf{T} & v\end{bmatrix}}{\text{minimize}} \quad & \begin{bmatrix}\mathbf{c}^\mathsf{T} & 0 & 1\end{bmatrix}\begin{bmatrix}\mathbf{f}\\ \mathbf{t}\\ v\end{bmatrix} + \text{const} \\ \text{subject to} \quad & \begin{bmatrix}\mathbf{F} & -\mathbf{I} & 0\\ \mathbf{A}_{leq}^\mathsf{T} & 0 & 0\end{bmatrix}\begin{bmatrix}\mathbf{f}\\ \mathbf{t}\\ v\end{bmatrix} \ge \begin{bmatrix}0\\ -\infty\end{bmatrix} \\ & \begin{bmatrix}\mathbf{F} & -\mathbf{I} & 0\\ \mathbf{A}_{leq}^\mathsf{T} & 0 & 0\end{bmatrix}\begin{bmatrix}\mathbf{f}\\ \mathbf{t}\\ v\end{bmatrix} \le \begin{bmatrix}0\\ \mathbf{b}_{leq}\end{bmatrix} \\ & \begin{bmatrix}\mathbf{f}\\ \mathbf{t}\\ v\end{bmatrix} \le \begin{bmatrix}\mathbf{u}_f\\ \infty\\ \infty\end{bmatrix} \\ & \begin{bmatrix}\mathbf{f}\\ \mathbf{t}\\ v\end{bmatrix} \ge \begin{bmatrix}\mathbf{l}_f\\ -\infty\\ 0\end{bmatrix} \\ & 2v \ge \sum_i t_i^2 \end{aligned}$$

Finally, we constrain known values in $\mathbf{f}$ by setting the corresponding entries in both $\mathbf{l}_f$ and $\mathbf{u}_f$ to be equal to the known values.