SUPPLEMENTAL MATERIAL

A DETAILED DERIVATION OF OPERATORS

Our Hessian operator consists of 4 matrices which are multiplied together. The first just represents the gradient on faces, in the intrinsic orthonormal basis with $\hat{\mathbf{x}}$ pointing along the edge between edge ij:

$$(\mathbf{Gu})_f = \begin{pmatrix} \frac{-1}{l_{ij}}u_i + \frac{1}{l_{ij}}u_j \\ \frac{l_{ki}\cos\theta_i - l_{ij}}{l_{ij}l_{ki}\sin\theta_i}u_i - \frac{\cos\theta_i}{l_{ij}\sin\theta_i}u_j + \frac{1}{l_{ki}\sin\theta_i}u_k \end{pmatrix}, \tag{28}$$

where the face f has vertices i, j, k, and θ_i is the tip angle at vertex i.

The second matrix takes differences of the previously-found gradients across a halfedge between the two faces. Explicitly, if θ_{f_i,f_j} is the angle the basis at face f would need to be rotated by to align with the basis at face f_j, the halfedge h_{ij} (on f_i corresponding to the edge shared between f_i and f_j) gets the matrix entries

$$\begin{aligned} (\mathbf{C})_{h_{ij},f_j} &= \left[\mathbf{R}(\theta)\right] \\ (\mathbf{C})_{h_{ij},f_i} &= \left[-\mathbf{I}_{2\times 2}\right] \end{aligned} \tag{29}$$

The third matrix divides the previous difference by the length of the dual edge between two centroids and performs a tensor product. Let e.g. $t_{ij,x}$ correspond to the x entry of the unit vector pointing from the centroid of f to the centroid at the face across halfedge ij in triangle f, and $l^\star_{ij}$ correspond to the intrinsic distance between these centroids. Additionally, let e.g. $c_{ij,x}$ correspond to the x entry in the difference taken over halfedge ij in face f. Then

$$(\mathbf{T})_f(c)_f = \begin{pmatrix} T_{xx} \\ T_{xy} \\ T_{yx} \\ T_{yy} \end{pmatrix}_f \quad (c)_f = \left(\hat{T}_1 \mid \hat{T}_2 \mid \hat{T}_3 \mid \hat{T}_4 \mid \hat{T}_5 \mid \hat{T}_6\right) \begin{pmatrix} c_{jk,x} \\ c_{jk,y} \\ c_{ki,x} \\ c_{ki,y} \\ c_{ij,x} \\ c_{ij,y} \end{pmatrix}_f$$

with

$$\hat{T}_1 = \left(\tfrac{1}{l^\star_{jk}}t^3_{jk,x} \quad \tfrac{1}{l^\star_{jk}}t^2_{jk,x}t_{jk,y} \quad \tfrac{1}{l^\star_{jk}}t^2_{jk,x}t_{jk,y} \quad \tfrac{1}{l^\star_{jk}}t_{jk,x}t^2_{jk,y}\right)^\top$$
$$\hat{T}_2 = \left(\tfrac{1}{l^\star_{jk}}t^2_{jk,x}t_{jk,y} \quad \tfrac{1}{l^\star_{jk}}t_{jk,x}t^2_{jk,y} \quad \tfrac{1}{l^\star_{jk}}t_{jk,x}t^2_{jk,y} \quad \tfrac{1}{l^\star_{jk}}t^3_{jk,y}\right)^\top$$
$$\hat{T}_3 = \left(\tfrac{1}{l^\star_{ki}}t^3_{ki,x} \quad \tfrac{1}{l^\star_{ki}}t^2_{ki,x}t_{ki,y} \quad \tfrac{1}{l^\star_{ki}}t^2_{ki,x}t_{ki,y} \quad \tfrac{1}{l^\star_{ki}}t_{ki,x}t^2_{ki,y}\right)^\top$$
$$\hat{T}_4 = \left(\tfrac{1}{l^\star_{ki}}t^2_{ki,x}t_{ki,y} \quad \tfrac{1}{l^\star_{ki}}t_{ki,x}t^2_{ki,y} \quad \tfrac{1}{l^\star_{ki}}t_{ki,x}t^2_{ki,y} \quad \tfrac{1}{l^\star_{ki}}t^3_{ki,y}\right)^\top$$
$$\hat{T}_5 = \left(\tfrac{1}{l^\star_{ij}}t^3_{ij,x} \quad \tfrac{1}{l^\star_{ij}}t^2_{ij,x}t_{ij,y} \quad \tfrac{1}{l^\star_{ij}}t^2_{ij,x}t_{ij,y} \quad \tfrac{1}{l^\star_{ij}}t_{ij,x}t^2_{ij,y}\right)^\top$$
$$\hat{T}_6 = \left(\tfrac{1}{l^\star_{ij}}t^2_{ij,x}t_{ij,y} \quad \tfrac{1}{l^\star_{ij}}t_{ij,x}t^2_{ij,y} \quad \tfrac{1}{l^\star_{ij}}t_{ij,x}t^2_{ij,y} \quad \tfrac{1}{l^\star_{ij}}t^3_{ij,y}\right)^\top$$

The fourth matrix $\mathbf{M}_F$ is a matrix containing facewise areas, repeated 4 times for each face.

Then the four intrinsic Hessian entries corresponding to a function $\mathbf{u}$ are given by the entries of

$$\mathbf{Hu} = \mathbf{M}_F\mathbf{TCGu} \tag{30}$$

Each matrix entry corresponding to boundary halfedges is zeroed out.

B FLAT L^1 HESSIAN BOUNDARY CONDITIONS IN 2D

Here, we derive the boundary conditions for the L^1 Hessian in a (nice enough) subset Ω of $\mathbb{R}^2$, assuming $\|Hu\|_F \neq 0$. We start with Equation (12), partially converting it to indices:

$$\int_\Omega \frac{1}{\|Hu\|_F}\partial_i\partial_j u \cdot \partial_i\partial_j\eta \, dx = 0\,. \tag{31}$$

Integrating by parts twice (once in i and once in j) yields

$$\begin{aligned} \int_{\partial\Omega} \frac{1}{\|Hu\|_F}\mathbf{n}^\top(Hu)(\nabla\eta) - \eta \cdot \left(\nabla \cdot \frac{Hu}{\|Hu\|_F}\right) \cdot \mathbf{n}\, dS \\ + \int_\Omega \eta \cdot H^*\left(\frac{Hu}{\|Hu\|_F}\right) dx = 0\,. \end{aligned} \tag{32}$$

The integration over Ω is precisely the interior pointwise condition given in Equation (13). Here we focus on understanding the boundary terms using two characteristic types of variation η, and subsequently drop the last term from (32).

B.1 Case 1: Variation is zero on $\partial\Omega$ and $\nabla\eta \propto \mathbf{n}$

First, we consider variations which are zero on the boundary, but which only have gradient in the normal direction. Such variations satisfy $\nabla\eta = g\mathbf{n}$ where g is a scalar function. The boundary integration becomes

$$\int_{\partial\Omega} \frac{1}{\|Hu\|_F}\mathbf{n}^\top(Hu)(g\mathbf{n}) - 0 \cdot \left(\nabla \cdot \frac{Hu}{\|Hu\|_F}\right) \cdot \mathbf{n}\, dS = 0 \tag{33}$$

and, since g is arbitrary,

$$\mathbf{n}^\top(Hu)\mathbf{n} = 0\,. \tag{34}$$

This is the same condition as appears in the L^2 scenario of Stein et al. [2018b].

B.2 Case 2: Variation is nonzero on $\partial\Omega$ and $\nabla\eta \propto \mathbf{t}$

In this case, the variation is nonzero on the boundary, and we also impose that the gradient of the variation is proportional to the tangent vector ($\nabla\eta = (\nabla\eta \cdot \mathbf{t})\mathbf{t}$ where $\mathbf{t}$ is unit length), and we impose the integrability condition $\oint_{\partial\Omega} \nabla\eta \cdot \mathbf{t}\, dS = 0$ ($\nabla\eta \cdot \mathbf{t}$ is continuous on the boundary). In coordinates, this gives

$$\int_{\partial\Omega} \left(\frac{1}{\|Hu\|_F}\mathbf{n}^\top(Hu)\mathbf{t}\right)(\mathbf{t}_i \cdot \partial_i\eta) - \eta \cdot \left(\nabla \cdot \frac{Hu}{\|Hu\|_F}\right) \cdot \mathbf{n}\, dS = 0\,. \tag{35}$$

Integration by parts on the leftmost term (a line integral on the boundary, where the derivative is $\mathbf{t} \cdot \nabla$) results in an integral over $\partial\partial\Omega$ (the empty set) and an integral over $\partial\Omega$,

$$-\int_{\partial\Omega} \eta \cdot \left(\nabla\left(\mathbf{t}^\top \frac{Hu}{\|Hu\|_F}\mathbf{n}\right) \cdot \mathbf{t} + \left(\nabla \cdot \frac{Hu}{\|Hu\|_F}\right) \cdot \mathbf{n}\right) dS = 0\,. \tag{36}$$

Finally, to convert this to a pointwise boundary condition, it remains to give a class of variations which satisfy $\nabla\eta = (\nabla\eta \cdot \mathbf{t})\mathbf{t}$ and $\oint_{\partial\Omega} \nabla\eta \cdot \mathbf{t}\, dS = 0$ and which converge to a point measure at some boundary point x_0. Let η be a smooth positive bump function on $\partial\Omega$ which is compactly supported in a ball of radius h around x_0 and which integrates to 1 over $\partial\Omega$; its gradient automatically fulfills the integrability condition. We extend η into the domain by keeping it constant in the inward normal direction over a small enough thickening of the boundary, and then by smoothly filling the interior of the domain from the inner boundary of this strip. This variation is within the class of variations we have restricted ourselves to (even for arbitrarily small support around x_0), and still satisfies Equation (36). So, pointwise,

$$\nabla\left(\mathbf{t}^\top \frac{Hu}{\|Hu\|_F}\mathbf{n}\right)\cdot\mathbf{t} + \left(\nabla\cdot\frac{Hu}{\|Hu\|_F}\right)\cdot\mathbf{n} = 0\,. \tag{37}$$

This condition is similar to the L^2 scenario of Stein et al. [2018b], with the difference of an additional denominator appearing.

C COMPUTING MODES

To compute the modes in Fig. 5, we alternate between

$$\begin{aligned} \mathbf{u}_i \leftarrow \operatorname*{argmin}_{\mathbf{u}} &\ \frac{1}{2}\mathbf{u}^\top\mathbf{L}\mathbf{u} + \mu\|\mathbf{H}\mathbf{u}\| \\ &\text{s.t.}\mathbf{u}_j^\top\mathbf{M}\mathbf{u} = 0 \quad \forall j < i \\ &\mathbf{c}_{i-1}^\top\mathbf{M}\mathbf{u} = 1\,, \text{ and} \\ \mathbf{c}_i \leftarrow &\ \frac{\mathbf{u}_i}{\sqrt{\mathbf{u}_i^\top\mathbf{M}\mathbf{u}_i}} \end{aligned} \tag{38}$$

until convergence, where $\mathbf{c}_0$ is initialized randomly, and $\mu = 1000000$. To find modes of the L^2 energy, we apply eigsh [Virtanen et al. 2020] to the L^2 matrix of Stein et al. [2018b].

D TRANSFORMING THE OPTIMIZATION PROBLEM INTO CONIC FORM

To optimize (21), we rearrange it into the form

$$\operatorname*{argmin}_{\mathbf{u}} \|\mathbf{H}\mathbf{u}\|_{\varphi,1} + \alpha\mathbf{u}^\top\mathbf{M}\mathbf{u} - 2\mathbf{u}_0^\top\mathbf{M}\mathbf{u} \qquad \text{s.t. } \mathbf{A}\mathbf{u} = \mathbf{b}\,, \tag{39}$$

where $\mathbf{H}$ is a matrix mapping $\mathbf{u}$ to its four intrinsic Hessian entries, and $\mathbf{M}$ is the (Voronoi) lumped mass matrix of Ω (see Supplemental Material A). The norm $\|\cdot\|_{\varphi,1}$ corresponds to taking the standard Euclidean norm (*not* the squared norm) of the vector with the four entries corresponding to each face, and then summing the norms over all faces. The nonlinear terms in the objective can be rewritten as standard conic problem constraints by introducing variables $r = 2\alpha\mathbf{u}^\top\mathbf{M}\mathbf{u}$ and $\mathbf{z} = \|(\mathbf{H}\mathbf{u})|_f\|$ (where $|_f$ represents taking the 4 entries of our linear Hessian operator that correspond to face f), and by Cholesky decomposing $2\alpha\mathbf{M} = \mathbf{L}\mathbf{L}^\top$

$$\begin{aligned} &\operatorname*{argmin}_{\mathbf{u}} \mathbf{1}^\top\mathbf{z} + r - 2\mathbf{u}_0^\top\mathbf{M}\mathbf{u} \\ &\text{s.t.} \begin{cases} A\mathbf{u} = \mathbf{b}, \\ (\mathbf{z})_f \geq \sqrt{\sum_i^4 \left((\mathbf{H}\mathbf{u})|_f\right)_i^2} \quad \forall f \\ r \geq \sum_j (\mathbf{L}^\top\mathbf{u})_j^2 \end{cases} \end{aligned} \tag{40}$$

The objective in this problem is linear in the variables $\mathbf{z}, r$, and $\mathbf{u}$, and the constraints correspond to a linear constraint, a quadratic conic constraint, and a rotated conic constraint respectively. It can be efficiently solved using black-box conic solvers.

E SEGMENTATION DISCRETIZATION AND OPTIMIZATION

We follow Weill–Duflos et al. [2023] by applying a smoothness energy to vertex-based functions in order to minimize a face-based quantity (v). Employing the face-based DEC Laplacian matrix $\mathbf{L}$ (using extrinsic distance between centroids) [Desbrun et al. 2005], the Voronoi mass matrix $\mathbf{M}$, and the diagonal matrix of face areas $\mathbf{M}_F$, we discretize the functional as

$$\begin{aligned} \mathbf{A}(\mathbf{u},\mathbf{v}) = \alpha(\mathbf{u}-\mathbf{u}_0)^\top\mathbf{M}(\mathbf{u}-\mathbf{u}_0) + (\mathbf{v}^2)^\top\|\mathbf{H}\mathbf{u}\|_\phi + \lambda\varepsilon\mathbf{v}^\top\mathbf{L}\mathbf{v} + \\ \frac{\lambda}{4\varepsilon}(\mathbf{v}^\top\mathbf{M}_F\mathbf{v} - \mathbf{2}^\top\mathbf{M}_F\mathbf{v})\,dx\,. \end{aligned} \tag{41}$$

where $\mathbf{v}^2$ is the elementwise square of $\mathbf{v}$, and $\|\cdot\|_\phi$ represents taking Frobenius norm over the 4 entries corresponding to a face. In this energy functional, λ penalizes length between segmentation regions, and α encourages solutions to closely approximate the original function. (41) is solved using an alternating scheme. For a fixed ε, we follow the procedure

$$\begin{aligned} \mathbf{v}^{(i+1)} &\leftarrow \operatorname*{argmin}_{\mathbf{v}} \mathbf{v}^\top\left(\operatorname{diag}(\|\mathbf{H}\mathbf{u}^{(i)}\|_\phi) + \lambda\varepsilon\mathbf{L} + \frac{\lambda}{4\varepsilon}\mathbf{M}_F\right)\mathbf{v} - \frac{\lambda}{4\varepsilon}\mathbf{2}^\top\mathbf{M}_F\mathbf{v} \\ \mathbf{u}^{(i+1)} &\leftarrow \operatorname*{argmin}_{\mathbf{u}} \alpha\mathbf{u}^\top\mathbf{M}\mathbf{u} + \mathbf{1}^\top(\operatorname{diag}(\mathbf{v}^{(i+1)})^2\|\mathbf{H}\mathbf{u}\|_\phi - 2\alpha\mathbf{u}_0^\top\mathbf{M}\mathbf{u} \end{aligned}$$

before halving ε and resolving, repeating until ε is below a threshold ε_2. The fixed $\mathbf{u}$ iteration amounts to a quadratic solve in $\mathbf{v}$, whereas the fixed $\mathbf{v}$ iteration amounts to a conic program solve in $\mathbf{u}$. To apply the segmentation procedure to geometry, we set $\mathbf{u}$ to be the vertex coodinates $(\mathbf{x},\mathbf{y},\mathbf{z})$.

At last, we apply postprocessing to find a true segmentation of the mesh elements. We first take the mean of $\mathbf{v}$ between the faces adjacent to each edge. We then cut the mesh along edges with $\mathbf{v}$ smaller than some threshold. We then merge the resulting connected components with fewer than τ faces into their neighboring components based on a majority vote along their boundaries; τ is a hyperparameter that should be chosen dependent on mesh resolution.

F PARAMETERS

Figure 1. The Cherry parameters were $\alpha_{\text{ours}} = 6.1479$ for our energy and $\alpha_{L^1} = 16.17$ for [Stein et al. 2018b] L^1. The Tete segmentation used the parameters $\lambda = 0.25, \alpha = 1000, \varepsilon_{\text{init}} = 0.1, \varepsilon_{\text{end}} = 0.001, \varepsilon_{\text{inner loop}} = 0.00001$ and used a cut threshold of 0.5 and merge threshold τ of 20 faces.

Figure 2. The Cathedral parameters were $\alpha_{\text{ours}} = 100$ for our energy, $\alpha_{L^2} = 0.05$ for L^2, $\alpha_{L^1} = 100$ for [Stein et al. 2018b] L^1, and $\alpha_{\Delta^1} = 25$ for the L^1 edge Laplacian energy.

Figure 3. The flowed cube was run for 50 iterations at fidelity weight 100. The denoised cube ran for 10 iterations at fidelity weight 500. The hole filled cube ran for 100 iterations at fidelity weight 100.

Figure 4. For the Springer, we ran our flow for 20 iterations at fidelity weight 500. For Koala, we ran our flow for 40 iterations at fidelity weight 500.

Figure 5. For our compressed modes on the Square and Catenoid, we used $\mu = 1000000$ and $\varepsilon = 5 \times 10^{-10}$ for the inner loop stopping criterion.

Figure 7. For the Skull, we used $\alpha_{\text{ours}} = 44.675350189208984$ for our energy, $\alpha_{L^2} = 0.11566162109375$ for L^2, $\alpha_{L^1} = 78.5$ for [Stein et al. 2018b] L^1. For this figure, the smoothing parameters were searched for by evaluating our optimization procedure at 5 equally spaced points across a large starting range, then recursing on the 3-point subinterval whose center point admitted the closest L^2 solution to the ground truth.

Figure 13. We used the authors' command line C++ implementation of [Liu and Jacobson 2019], with a parameter $\lambda = 1$. Our Hand ran for 43 iterations at a fidelity weight 500.

Figure 14. For Nefertiti, the fidelity parameters are set to $\eta = 1250$ for 25 steps, $\eta = 500$ for 10 steps, and $\eta = 50$ for 1 step.

Figure 15. The Dodecahedron used $\lambda = 0.3, \alpha = 1000, \varepsilon_{\text{init}} = 0.1, \varepsilon_{\text{end}} = 0.001, \varepsilon_{\text{inner loop}} = 0.00001$, a cut threshold of 0.6, and a merge limit of 4. The Fandisk used $\lambda = 0.5, \alpha = 1000, \varepsilon_{\text{init}} = 0.1, \varepsilon_{\text{end}} = 0.01, \varepsilon_{\text{inner loop}} = 0.00001$, a cut threshold of 0.305, and a merge limit of 4. The Moai used $\lambda = 0.1, \alpha = 1000, \varepsilon_{\text{init}} = 0.1, \varepsilon_{\text{end}} = 0.0001, \varepsilon_{\text{inner loop}} = 0.00001$, a cut threshold of 0.7, and a merge limit of 30.

Figure 16. The Horseshoe flowed for 80 iterations at a fidelity weight 100. Evora flowed for 40 iterations at a fidelity weight 1000.

Figure 17. The Building used $\lambda = 0.03, \alpha = 0.5, \varepsilon_{\text{init}} = 0.1, \varepsilon_{\text{end}} = 0.001, \varepsilon_{\text{inner loop}} = 0.00001$, a cut threshold of 0.925, and a merge limit of 20.

Figure 18. For the Moai λ test, all parameters except λ were kept the same as in the Moai subfigure of Figure 15.

Figure 19. The segmentation for the Cow used $\lambda = 0.35, \alpha = 1000, \varepsilon_{\text{init}} = 0.1, \varepsilon_{\text{end}} = 0.0001, \varepsilon_{\text{inner loop}} = 0.0001$, a cut threshold of 0.95, and merge limit 4. The segmentation for the Stravinsky Fountain character used $\lambda = 2, \alpha = 10000, \varepsilon_{\text{init}} = 0.1, \varepsilon_{\text{end}} = 0.0001, \varepsilon_{\text{inner loop}} = 0.0001$, as well as a cut threshold of 0.95 and a merge threshold of 4.

Figure 20. The Cubesphere and UV Sphere both ran for 250 iterations at fidelity weight 500.

G SUPPLEMENTAL MATHEMATICA CALCULATIONS

A supplemental file is attached to this work giving the formula for the (intrinsic) distances and vector entries between a face's centroid and its adjacent faces' centroids as computed in Mathematica from only the edge lengths of a generic triangle.