Local Barycentric Coordinates

Juyong Zhang[1*] Bailin Deng[2*] Zishun Liu[1] Giuseppe Patanè[3] Sofien Bouaziz[2] Kai Hormann[4] Ligang Liu[1]

[1]University of Science and Technology of China [2]École polytechnique fédérale de Lausanne
[3]CNR - Istituto di Matematica Applicata e Tecnologie Informatiche [4]Università della Svizzera italiana

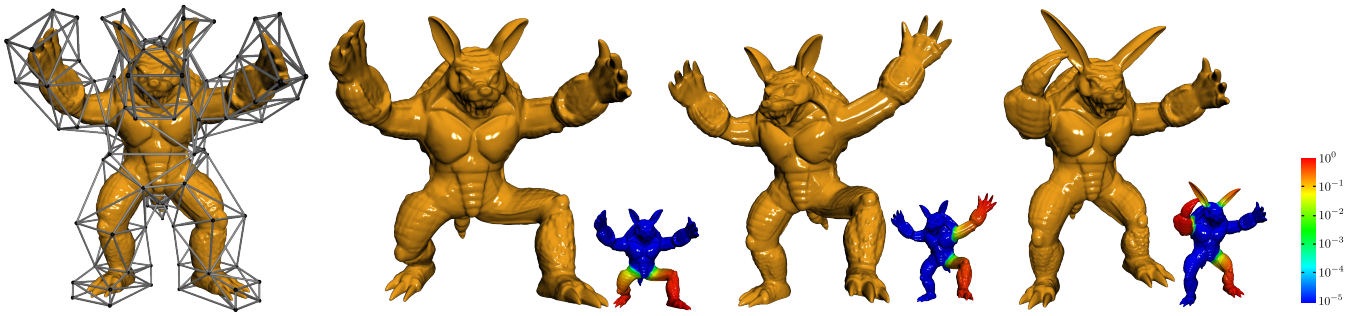


Figure 1: *Using LBC for 3D cage-based manipulation allows for local, smooth, and shape-aware deformations. Only parts near the manipulated control points are deformed, as indicated by the logarithmic color-coding of the displacement magnitude.*

Abstract

Barycentric coordinates yield a powerful and yet simple paradigm to interpolate data values on polyhedral domains. They represent interior points of the domain as an affine combination of a set of control points, defining an interpolation scheme for any function defined on a set of control points. Numerous barycentric coordinate schemes have been proposed satisfying a large variety of properties. However, they typically define interpolation as a combination of *all* control points. Thus a *local* change in the value at a single control point will create a *global* change by propagation into the whole domain. In this context, we present a family of *local barycentric coordinates* (LBC), which select for each interior point a small set of control points and satisfy common requirements on barycentric coordinates, such as linearity, non-negativity, and smoothness. LBC are achieved through a convex optimization based on total variation, and provide a compact representation that reduces memory footprint and allows for fast deformations. Our experiments show that LBC provide more local and finer control on shape deformation than previous approaches, and lead to more intuitive deformation results.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Geometric algorithms, languages, and systems;

Keywords: barycentric coordinates, total variation, locality, smoothness, shape deformation, image warping.

Links: DL PDF CODE

*Joint first authors. Email: juyong@ustc.edu.cn, bailin.deng@epfl.ch

1 Introduction

Barycentric coordinates provide a simple and convenient way to interpolate values from a set of control points over the interior of a domain, using weighted combinations of values associated with the control points. Due to their simplicity and efficiency, they have been successfully applied to various problems in computer graphics, including image composition and warping [Farbman et al. 2009], shape deformation [Ju et al. 2005; Lipman et al. 2007], texture mapping [Desbrun et al. 2002], and synthesis [Takayama et al. 2010]. Current barycentric coordinates typically are of global nature, meaning that the interpolated value depends on many, potentially *all*, control points. This implies two main drawbacks. The first one is the lack of locality and control over a deformation. For example, in design tasks such as shape and image deformation, where the users directly manipulate control points, editing just one control point potentially influences the whole design, yielding a counter-intuitive behavior. Even worse, manually achieving any localized edit might be impossible since it would involve manipulating a large amount of control points to achieve the desired deformation. The second drawback is scalability. Most practical applications store barycentric coordinates using one scalar value per control point for every vertex of the target domain. For high-resolution shapes with many control points, this leads to high memory consumption. Furthermore, the interpolation is computationally expensive: it involves a weighted sum of all control points for each interior vertex. Thus, barycentric coordinates with locality provide benefits in terms of storage requirements as well as computational cost.

Overview and contributions. This paper introduces a novel method to derive *local barycentric coordinates* (LBC), which depend only on a small number of control points. LBC are computed by minimizing a target functional based on *total variation* (TV), subject to a set of constraints that ensure desired properties such as partition of unity, reproduction, and non-negativity. The TV energy, widely used for image smoothing and reconstruction [Rudin et al. 1992], induces locality and regularity of the computed coordinates. The resulting LBC are local, meaning that each control point only influences a nearby region. As a result, LBC induce lower computational cost for applications such as cage-based deformation, since each point on the target shape is only determined by a small number

of control points. A main advantage of our formulation is that it leads to a convex optimization problem, whose global minimum can be efficiently computed. Moreover, the TV energy penalizes local extrema in the coordinate functions. Local extrema are typically due to high order Laplacian energy terms used to improve the smoothness of the resulting barycentric coordinates. While previous work [Jacobson et al. 2012] reduces oscillations of the coordinate functions by introducing more constraints, the TV energy inhibits local extrema without the need for extra terms.

Our optimization automatically determines a local influence region for each control point, and only needs to be performed once for a given cage. In contrast to [Li et al. 2010], LBC do not require a priori selection of influence regions of the control points. Unlike the Poisson-based approach in [Landreneau and Schaefer 2010], the locality and smoothness of LBC are induced by TV minimization, without requiring user-defined parameters to manage the number of control points influencing a vertex during mesh deformation. LBC achieve better locality than *bounded biharmonic weights* (BBW) [Jacobson et al. 2011], while also satisfying the reproduction property that is not available from BBW.

We apply LBC to mesh deformation, where contrary to previous barycentric coordinate approaches, our method allows both high-quality deformations and control at different levels of granularity without the need for fixing an influence region manually. Control over the locality of deformations is an important feature as natural surfaces and images are intrinsically multi-scale.

2 Related work

Barycentric coordinates. Barycentric coordinates, introduced by Möbius [1827], remain an active area of research in computer graphics and mathematics with numerous applications in image and geometry processing. While being uniquely defined for simplices, numerous extensions of barycentric coordinates have been presented for convex [Pinkall and Polthier 1993; Dasgupta and Wachspress 2008] and arbitrary polygons [Hormann and Floater 2006] and curves [Schaefer et al. 2007]. Variants include mean value [Floater 2003; Ju et al. 2005; Hormann and Floater 2006], Green [Lipman et al. 2008], and Poisson [Li and Hu 2013] coordinates. Barycentric coordinates have been enriched with constraints on their regularity and accuracy [Li et al. 2013], harmonicity [Joshi et al. 2007; Jacobson et al. 2011; Weber et al. 2012], positiveness [Lipman et al. 2007; Hormann and Sukumar 2008], with generalizations to the complex plane [Weber et al. 2009; Weber and Gotsman 2010; Weber et al. 2011]. However, to the best of our knowledge, no existing barycentric coordinate scheme explicitly enforces locality.

Local deformation using control weights. Despite the large number of control weight schemes that have been proposed, few of them address the issue of locality for deformation. Li et al. [2010] propose a cage-free local deformation method with vertex handles, using an umbrella-shaped shell and a local influence region associated to the vertex. The influence region needs to be specified by the user, and the umbrella shell needs to be constantly updated during deformation. Unlike this method, our optimization approach automatically determines a local influence region for each control point, and this optimization only needs to be done once for a given cage. Landreneau and Schaefer [2010] introduce a Poisson-based approach to reduce the number of control points influencing a vertex during mesh deformation by limiting the number of nonzero control weights for the vertex to a user-specified threshold. This threshold needs to be properly chosen, in order to achieve weight reduction without sacrificing their smoothness. On the contrary, in our method the locality and smoothness properties are induced by the TV-based target functional, without requiring a proper threshold value.

Jacobson et al. [2011] propose a constrained optimization approach to compute *bounded biharmonic weights* (BBW) that are both smooth and local. The constraints they enforce are similar to those from our method, except that they do not impose the reproduction property. As a result, BBW are not barycentric coordinates, and do not provide linear precision for interpolation. In comparison, LBC are more local than BBW, while achieving the linear precision property.

García et al. [2013] propose a multi-cage mesh deformation approach called **Cages*, which achieves local deformation by employing independent control weight schemes in each subcage and blending the weights from different subcages. As pointed out by the authors, their method complements existing control weight schemes rather than competing with them, and the user is free to choose the control weight scheme of each subcage. As a result, when employing LBC for the subcages, *Cages will benefit from the locality of LBC and provide more localized deformations with lower memory footprints, compared to other subcage control weight schemes.

Local interpolation schemes. Within the literature of scattered data interpolation, there exist schemes that enforce local influence from the input sample points. For example, Sibson [1981] develop the natural neighbor interpolation schemes where the interpolated values are computed as a linear combination of the sample values, with the linear combination coefficients determined from the Voronoi diagram of the sample points; the Voronoi diagram determines a local influence region for each sample point, and the interpolation schemes achieve up to C^1 continuity. For higher order continuity, Hiyoshi and Sugihara [2000] extend Sibson's method using recursive integration. When considering cage vertices as sample points for these Voronoi-based interpolation schemes, the interpolation coefficient functions share many desirable properties with LBC, such as partition of unity, reproduction, and non-negativity. However, since the Voronoi diagrams are computed from the Euclidean distance, such approaches can produce coefficients functions that are local with respect to the ambient Euclidean metric but non-local with respect to the distance inside the cage, which leads to unintuitive deformations. On the contrary, LBC enforce locality within the cage, and provide more intuitive control.

Total variation. In this paper, we compute LBC by minimizing the sum of total variation of the barycentric coordinate functions for all control points. For a function f defined on a domain $\Omega \subseteq \mathbb{R}^D$, its TV is defined as

$$J_f = \int_\Omega |\nabla f(x)| dx, \tag{1}$$

where $|\nabla f|$ is the ℓ_2-norm of the gradient ∇f. TV is a popular tool for image processing tasks, such as denoising, reconstruction, and segmentation [Chambolle et al. 2010]. Two key properties of TV have led to its success. First, TV provides a measure of oscillation, thus TV minimization reduces oscillation from the resulting function [Chan et al. 2011]. This is the foundation of the image denoising method from Rudin et al.'s seminal paper [1992], as well as many TV-based image smoothing schemes [Chan et al. 2006]. In our formulation, this property inhibits local extrema in LBC. The other key property is that for the characteristic function of a set, its TV equals the perimeter of the set [Evans and Gariepy 1992]. Therefore, TV is widely used in image segmentation as a regularization term for boundary curves [Chan and Vese 2001; Goldstein et al. 2010]. This property is crucial for the locality and smoothness of LBC. In particular, it relates TV to the length/area of levelset curves/surfaces, which is also related to the area/volume of the domain bounded by

them due to the isoperimetric inequality [Osserman 1978]. Besides image applications, TV has also been applied to geometry processing problems such as surface fairing and reconstruction [Elsey and Esedoglu 2009]. Despite being non-smooth, TV regularization is convex and leads to convex optimization problems, which can be efficiently handled by various numerical algorithms [Weiss et al. 2009; Goldstein and Osher 2009; Wu and Tai 2010].

3 Local barycentric coordinates

Let $\mathbf{c}_1, \ldots, \mathbf{c}_n$ be a set of control points in $\mathbb{R}^2$ or $\mathbb{R}^3$ which are the vertices of a closed control cage, and let Ω be the domain bounded by the cage. Our goal is to find a function $w_i \colon \Omega \mapsto \mathbb{R}$ for each $\mathbf{c}_i$, such that $[w_1(\mathbf{x}), \ldots, w_n(\mathbf{x})]$ is a set of generalized barycentric coordinates of $\mathbf{x} \in \Omega$ with respect to the control points $\{\mathbf{c}_i\}$. These coordinate functions are used for interpolating function values $f(\mathbf{c}_1)$, …, $f(\mathbf{c}_n)$, given at the control points over the interior of Ω by

$$f(\mathbf{x}) = \sum_{i=1}^{n} w_j(\mathbf{x}) f(\mathbf{c}_i). \tag{2}$$

Such an interpolation scheme enables shape deformation on Ω using the control points as handles. For the quality of the interpolation, we are interested in coordinate functions with the following properties:

1. *Reproduction*: $\sum_{i=1}^{n} w_i(\mathbf{x})\mathbf{c}_i = \mathbf{x},\ \forall \mathbf{x} \in \Omega$;
2. *Partition of unity*: $\sum_{i=1}^{n} w_i(\mathbf{x}) = 1$;
3. *Non-negativity*: $w_i(\mathbf{x}) \geq 0\ \ \forall i$;
4. *Lagrange property*: $w_i(\mathbf{c}_j) = \begin{cases} 0, & \text{if } i \neq j, \\ 1, & \text{otherwise;} \end{cases}$
5. *Linearity*: functions $\{w_i\}$ are linear on cage edges and faces;
6. *Smoothness*: functions $\{w_i\}$ vary smoothly on Ω;
7. *Locality*: a control point only influences its nearby regions, and a point $\mathbf{x} \in \Omega$ is influenced by a small number of control points, i.e., the vector $[w_1(\mathbf{x}), \ldots, w_n(\mathbf{x})]$ is *sparse*.

Here, the reproduction and partition of unity properties are the defining properties for barycentric coordinates and imply linear precision of the interpolation, i.e., linear functions can be reproduced from their values on the control points by an interpolation using barycentric coordinates. Non-negative coordinates prevent unintuitive deformations that result from inconsistent deformation directions between the control points and the cage interior [Jacobson et al. 2011].

3.1 Formulation

Achieving all the target properties listed above is not an easy task. While properties 1–6 are satisfied by many existing barycentric coordinate schemes, the locality property adds complexity and has received much less attention in previous work. The main challenge here is that the coordinate functions are globally coupled: reducing the influence region for one control point might lead to more global influence of other control points. Therefore, all control points must be considered simultaneously to achieve a barycentric coordinate scheme with local influence. To this end, we formulate our coordinate functions as the solution of a constrained optimization problem. The constraints correspond to a subset of the target properties defined above, while the target functional induces locality for the solution functions. Since locality implies sparsity of the vector $\mathbf{w}(\mathbf{x}) = [w_1(\mathbf{x}), \ldots, w_n(\mathbf{x})]$, a straightforward choice is to employ a sparsity term in the target functional while imposing smoothness constraints, similar to [Rustamov 2011]. The sparsity of $\mathbf{w}(\mathbf{x})$ is measured by its ℓ_0-norm, which is the number of its nonzero components. Due to the combinatorial complexity of minimizing the ℓ_0-norm, it is often relaxed using the ℓ_p-norm with $0 < p \leq 1$ [Bach et al. 2012], leading to the following target energy

$$F_{\ell_p} = \sum_{i=1}^{n} \int_{\Omega} |w_i(\mathbf{x})|^p. \tag{3}$$

Setting $p = 1$ is the most popular approach for inducing sparsity, since it leads to a convex term that is effective for finding the sparsest solutions for many problems [Bruckstein et al. 2009]. However, it is not suitable for our case, because the non-negativity and partition of unity properties imply that $\sum_{i=1}^{n} |w_i(\mathbf{x})| = 1\ \forall \mathbf{x} \in \Omega$; as a result, F_{ℓ_1} always equals the area/volume of Ω, regardless of the actual values of the coordinate functions. Thus F_{ℓ_1} is unable to reveal sparsity among the candidate functions. On the other hand, choosing $p \in (0, 1)$ provides a tighter relaxation of the ℓ_0-norm [Chartrand 2007], and captures sparsity better than the ℓ_1-norm. But the optimization problem becomes nonconvex, making it difficult to find the global minimum [Ge et al. 2011].

To make the optimization approach effective, we prefer a target functional that reflects locality and smoothness for the coordinate functions while still being convex. For a function w_i and a given value s, denote by $\{w_i > s\} := \{\mathbf{x} \mid w_i(\mathbf{x}) > s\}$ and $\{w_i = s\} := \{\mathbf{x} \mid w_i(\mathbf{x}) = s\}$ the *superlevel set* and the *level set* of s, respectively. Locality requires the area/volume of the superlevel set $\{w_i > 0\}$ to be small, while for smoothness it is necessary that all curves/surfaces $\{w_i = \text{const}\}$ are smooth. We make the following observations in the 2D case:

- For a domain in $\mathbb{R}^2$, its perimeter L provides an upper bound on its area A, due to the isoperimetric inequality $4\pi A \leq L^2$;
- For a domain with fixed area, its perimeter indicates the regularity of its boundary curve, with the minimum perimeter achieved by a circular boundary;
- If a level set curve $\{w_i = s\}$ exists, then it is the boundary curve of the superlevel set $\{w_i > s\}$.

Similar observations are made in 3D as well. They motivate us to induce the locality and smoothness of w_i by minimizing the sum of the perimeters of superlevel sets $\{w_i > s\}$ for all s. In this way, the perimeter of each superlevel set regularizes the smoothness of its boundary level curve/surface, while the perimeter of $\{w_i > 0\}$ penalizes the area/volume of the influence region. It turns out that this sum is exactly the TV of w_i. Specifically, let $P(\{w_i > s\}; \Omega)$ be the perimeter of $\{w_i > s\}$ on the domain Ω. Then, the TV of w_i satisfies the following *coarea formula* [Ambrosio et al. 2000]:

$$\int_{\Omega} |\nabla w_i| = \int_{-\infty}^{+\infty} P(\{w_i > s\}; \Omega)\, ds. \tag{4}$$

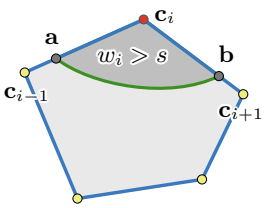
Utilizing this strategy, our formulation minimizes the sum of TV for all coordinate functions, subject to the constraints of reproduction, partition of unity, non-negativity, linearity, and Lagrange property:

$$\begin{aligned} \min_{w_1, \ldots, w_n} \ & \sum_{i=1}^{n} \int_{\Omega} |\nabla w_i| \\ \text{s.t. } & \sum_{i=1}^{n} w_i(\mathbf{x})\mathbf{c}_i = \mathbf{x},\ \sum_{i=1}^{n} w_i = 1,\ w_i \geq 0,\ \forall \mathbf{x} \in \Omega, \\ & w_i(\mathbf{c}_j) = \delta_{ij}\ \forall i, j, \\ & w_i \text{ is linear on cage edges and faces } \forall i. \end{aligned} \tag{5}$$

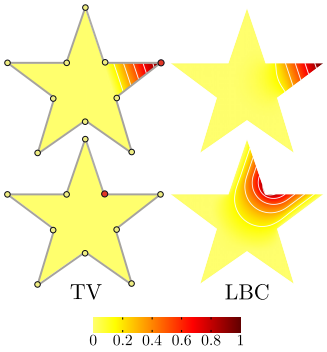
Thanks to the convexity of TV, this is a convex optimization problem, and its global minimum can be computed efficiently.

3.2 Properties of LBC

Our formulation is effective in ensuring all the desired properties of the coordinate functions. As constraints, properties 1-5 are automatically satisfied by the optimization result. In the following, we focus on the locality and smoothness properties.



Locality. The Lagrange and linearity properties of our coordinate functions define Dirichlet boundary conditions. In the 2D case, the coordinate function w_i attains value 1 at control point $\mathbf{c}_i$ and decays linearly to zero along its neighboring cage edges, while vanishing on other cage edges. For a continuous function w_i under such boundary conditions, the boundary of the superlevel set $\{w_i > s\}$ $(0 < s < 1)$ connects two points $\mathbf{a}$, $\mathbf{b}$ on the cage where $w_i(\mathbf{a}) = w_i(\mathbf{b}) = s$ (inset above). If we minimize the TV of w_i over Ω *under its boundary conditions only*, then the boundary of $\{w_i > s\}$ contains the shortest path between $\mathbf{a}$ and $\mathbf{b}$ in Ω, because TV penalizes its length (inset, top left). In particular, if $\mathbf{c}_i$ is a convex vertex of the cage, and no other control point lies on the triangle $\mathbf{c}_i\mathbf{c}_{i-1}\mathbf{c}_{i+1}$, then the resulting level set curve $\{w_i = s\}$ is a straight line segment connecting $\mathbf{a}$ and $\mathbf{b}$; it follows that w_i is linear on the triangle $\mathbf{c}_i\mathbf{c}_{i-1}\mathbf{c}_{i+1}$, and vanishes outside the triangle (inset, top left). If $\mathbf{c}_i$ is a concave vertex, then the shortest path between $\mathbf{a}$ and $\mathbf{b}$ lies on the cage, implying a degenerate case where w_i vanishes in the interior of Ω (inset, bottom left). In either case, TV effectively induces locality. Similar arguments apply to the 3D case. In our formulation, all coordinate functions are optimized simultaneously under global constraints. Since they are globally coupled, the resulting functions have larger support compared to unconstrained TV minimization (inset, right). Nevertheless, they are still local compared to other barycentric coordinate schemes, due to the effect of TV (Fig. 2).



One benefit of the TV formulation is that it measures the superlevel set perimeters within the cage, and the resulting solution functions are local with respect to the distance inside the cage. In comparison, Voronoi-based interpolation schemes such as [Sibson 1981] and [Hiyoshi and Sugihara 2000] determine the influence of sample points based on the Euclidean distance, which might be different from the distance inside the cage. As a consequence, Voronoi-based interpolation schemes can lead to influence regions that are local in the ambient space but non-local within the cage, which is unintuitive for cage-based deformation (Fig. 3).

The difference in locality between LBC and harmonic coordinates can also be understood from the perspective of sparsity optimization. For a coordinate function w_i to have local influence, it needs to vanish over a large region, which also means that the gradient ∇w_i is zero in this region. Thus, *a necessary condition for local support of w_i is that the gradient norm function $H_i(\mathbf{x}) = |\nabla w_i(\mathbf{x})|$ vanishes over a large domain*. The TV of w_i is the ℓ_1-norm of $H_i(\mathbf{x})$. Since the ℓ_1-norm regularization promotes sparsity and penalizes the sizes of nonzero regions for the resulting function [Bach et al. 2012], our optimization for LBC produces coordinate functions with large regions of vanishing gradient (Fig. 4 top), satisfying the above

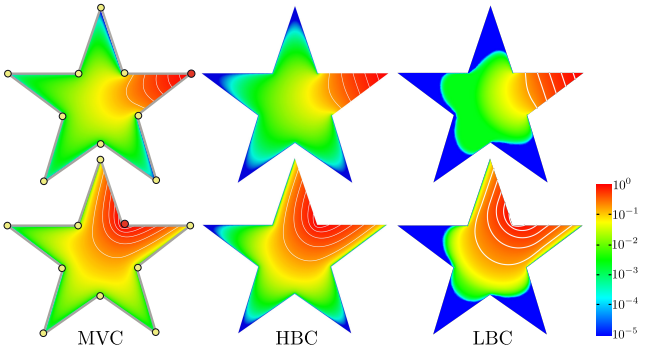


Figure 2: *Comparison of locality between mean value coordinates (MVC), harmonic coordinates (HBC), and local barycentric coordinates (LBC). The color-coding shows the coordinate function values for a convex (top) and a concave (bottom) control point in red.*

necessary condition of locality in terms of the gradient norm. Moreover, since the ℓ_1-norm regularization allows for large values of the resulting function in its nonzero region, LBC have large gradients in neighboring regions of the control point. Thus LBC decrease rapidly when moving away from the control point, resulting in local influence (Fig. 4, bottom). On the other hand, harmonic coordinates [Joshi et al. 2007] are harmonic functions. They are critical points of the Dirichlet energy, thus minimizing the ℓ_2-norm of the gradient norm function. ℓ_2-norm regularization penalizes large values of the resulting function and inhibits its sparsity [Bach et al. 2012]. This prevents harmonic coordinates from having large regions of zero gradient, which violates the above necessary condition of locality and leads to their global influence. In fact, since harmonic coordinates are analytic functions and non-constant, their zero level sets can only have zero measure [Krantz and Parks 2002].

Remark. In Fig. 4, we choose $10^{-3}/n$ as the threshold to indicate negligible influence from a control point, based on the following observation: if all n control vertices have the same influence at a point, then their coordinate functions have value $1/n$, due to the partition of unity and non-negativity properties. Thus the threshold $10^{-3}/n$ is chosen to take into account the total number of control vertices. This threshold value is used throughout the whole paper.

Smoothness. At a point $\mathbf{x} \in \Omega$, if the solution function w_i for problem (5) has nonzero gradient, then it will satisfy the following

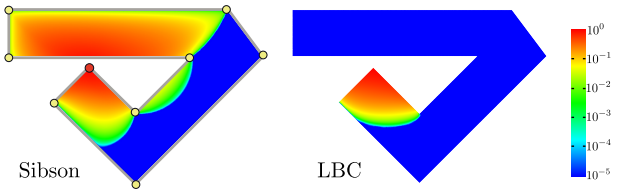


Figure 3: *Comparison between LBC and Sibson's C^0 natural neighbor interpolation scheme. The color-coding shows respectively the interpolation coefficient function (for Sibson's method) and the coordinate function (for LBC) for the red vertex. Using Sibson's method, the red vertex influences the other end of the cage due to their proximity according to the Euclidean distance. On the contrary, with LBC it only influences nearby regions according to the distance inside the cage, providing more intuitive control.*

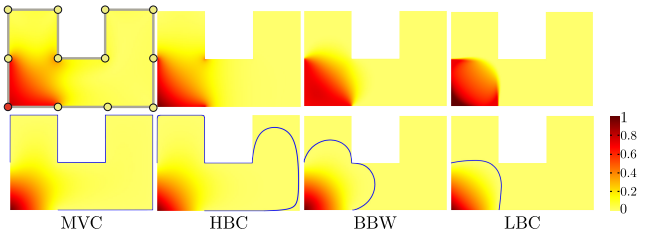


Figure 4: *Comparison of gradient norm and influence region. The color-coding shows the weight function values (bottom row) for the red control point and their normalized gradient norm (top row). The bottom row also shows the level set curves for the threshold value $10^{-3}/n$ (in blue). Compared to other weighting schemes, LBC have the smallest region with values greater than $10^{-3}/n$.*

Euler–Lagrange equation (see Appendix A for its derivation)

$$\nabla \cdot \frac{\nabla w_i(\mathbf{x})}{|\nabla w_i(\mathbf{x})|} + \boldsymbol{\gamma}_1(\mathbf{x}) \cdot \mathbf{c}_i + \gamma_2(\mathbf{x}) + \gamma_3(\mathbf{x}) = 0, \qquad (6)$$

where $\boldsymbol{\gamma}_1, \gamma_2, \gamma_3$ are the Lagrange multiplier functions for the constraints of reproduction, partition of unity, and non-negativity properties, respectively. Since Eq. (6) involves second derivatives of w_i, the function w_i needs to be at least C^1 at $\mathbf{x}$. On the other hand, if $\mathbf{x}$ is in the interior of a region where $\nabla w_i = \mathbf{0}$, then w_i is constant in a neighborhood of $\mathbf{x}$ and thus C^1 as well. Therefore, w_i is guaranteed to be C^1 except at the boundaries of regions where its gradient vanishes, and the set of non-C^1 points in Ω has zero measure.

Remark. Eq. (6) is only applicable to points with nonzero gradients, since the term $\nabla \cdot (\nabla w_i / |\nabla w_i|)$ is not well-defined when $\nabla w_i = \mathbf{0}$ as the ℓ_2-norm function is not differentiable at zero. This prevents Eq. (6) from ensuring C^1 continuity of w_i at the boundaries of regions with vanishing gradients. Global C^1 continuity can be achieved by relaxing the target functional in (5) to

$$\sum_{i=1}^{n} \int_{\Omega} f_\epsilon(\nabla w_i), \qquad (7)$$

where the function $f_\epsilon : \mathbb{R}^D \mapsto \mathbb{R}$ is defined as

$$f_\epsilon(\mathbf{u}) = \begin{cases} |\mathbf{u}|, & \text{if } |\mathbf{u}| \geq \epsilon, \\ -\dfrac{|\mathbf{u}|^3}{3\epsilon^2} + \dfrac{|\mathbf{u}|^2}{\epsilon} + \dfrac{\epsilon}{3}, & \text{otherwise,} \end{cases} \qquad (8)$$

for some small $\epsilon > 0$. The function f_ϵ is C^1 and convex, and bounds the ℓ_2-norm function from above: $f_\epsilon(\mathbf{u})$ is different from $|\mathbf{u}|$ only when $|\mathbf{u}| < \epsilon$, with the maximum difference being $\epsilon/3$, achieved at $\mathbf{u} = \mathbf{0}$. Thus, the functional in (7) provides a tight upper bound for the original target functional in (5), and its minimization still penalizes the superlevel set perimeters, albeit not as strongly as TV. The Euler–Lagrange equation of this relaxed problem becomes

$$\nabla \cdot f'_\epsilon(\nabla w_i(\mathbf{x})) + \boldsymbol{\gamma}_1(\mathbf{x}) \cdot \mathbf{c}_i + \gamma_2(\mathbf{x}) + \gamma_3(\mathbf{x}) = 0, \qquad (9)$$

which is satisfied at every interior point of Ω. Then w_i is C^1 everywhere due to the presence of its second derivatives. This, however, comes at the price of locality: due to the gap between the relaxed functional in (7) and the original TV-based functional, the solution to the relaxed problem is less local. The larger ϵ is, the more global the solution functions become (Fig. 5). Nevertheless, with a small value of ϵ, the solution functions are still more local than BBW, while achieving global C^1 continuity. In the remainder of this paper, all LBC examples are computed without relaxation.

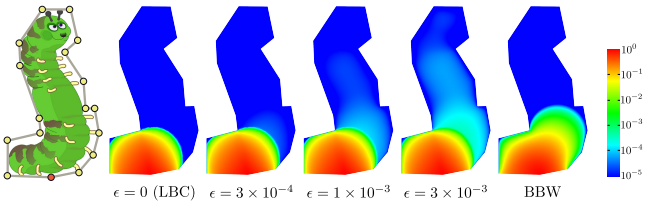


Figure 5: *Relaxing the TV according to Eq.* (7) *guarantees global C^1 smoothness, at the expense of more global influence. For a small value of ϵ, the result is still more local than BBW.*

No local extrema. For cage-based deformation, it is desirable that the weight functions are free of local extremal values in the interior of the cage, to prevent unintuitive deformation. In all our experiments, we observed that LBC have no local extrema. This is not surprising, since TV measures oscillation, and hence its minimization inhibits local extremal values. On the contrary, optimizing energies with higher order differential operators often induces local extrema [Jacobson et al. 2012].

3.3 Controlling locality

When minimizing the TV energy, the penalty for the gradient norm is uniform across the whole domain Ω. To obtain coordinate functions with more local support, it is preferable to penalize the gradient norm based on the distance to the control points within the cage. Ideally, for a control point $\mathbf{c}_i$, its coordinate function w_i should decrease quickly to zero when moving away from $\mathbf{c}_i$ and remain zero at regions far away from $\mathbf{c}_i$. In other words, it is desirable that the gradient norm $|\nabla w_i|$ is large at regions close to $\mathbf{c}_i$, and vanishes far away from $\mathbf{c}_i$. Accordingly, regions farther away from $\mathbf{c}_i$ should receive a higher penalty for $|\nabla w_i|$. Based on these considerations, we extend our formulation by incorporating a spatially varying weight function $\phi_i : \Omega \mapsto [0, 1]$ into the evaluation of TV. Specifically, the target functional in (5) is modified to

$$\sum_{i=1}^{n} \int_{\Omega} \phi_i |\nabla w_i|, \qquad (10)$$

with ϕ_i determined by the normalized distance $D_i(\mathbf{x})$ from $\mathbf{x}$ to $\mathbf{c}_i$,

$$\phi_i(\mathbf{x}) = \tau(D(\mathbf{x})), \qquad (11)$$

where $\tau : [0, 1] \mapsto [0, 1]$ is a continuous function, and

$$D_i(\mathbf{x}) = \frac{g_i(\mathbf{x})}{\max\limits_{\mathbf{y} \in \Omega} g_i(\mathbf{y})}, \qquad (12)$$

with $g_i(\cdot)$ being the geodesic distance to $\mathbf{c}_i$ within Ω. Using a monotonically increasing function τ, this weighting scheme imposes a larger penalty for points further away. Fig. 6 shows the results of weighted TV minimization with different choices of τ. The case where $\tau \equiv 1$ is equivalent to unweighted TV minimization. It shows that a monotonically increasing τ leads to more local support than the unweighted case. We can even choose a monotonically decreasing τ, which results in more global support. Thus, the choice of weighting schemes enables the user to fine-tune the locality of optimization results, providing more flexibility in our framework.

The effects of weighting can also be interpreted using a weighted version of the coarea formula that generalizes Eq. (4). For a weighting function ϕ_i, we denote by $P_{\phi_i}(\{w_i > s\}; \Omega)$ the weighted perimeter of the superlevel set $\{w_i > s\}$, which is evaluated by locally scaling the length/area element at each point $\mathbf{x}$ of its boundary

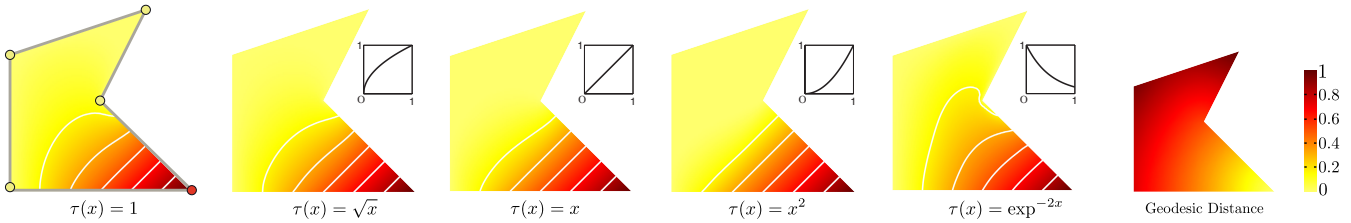


Figure 6: *Weighted formulation of LBC. Using weighted TV, we can control the level of locality for LBC. From left to right, we show the effect of different functions τ in* (11)*. Monotonically increasing functions τ lead to more local support, while monotonically decreasing functions τ lead to more global support. The rightmost image shows the normalized distance function* (12) *for the control point.*

curve/surface by $\phi_i(\mathbf{x})$. It can be shown [Grasmair 2010] that

$$\int_\Omega \phi_i |\nabla w_i| = \int_{-\infty}^{+\infty} P_{\phi_i}(\{w_i > s\}; \Omega)\, ds. \tag{13}$$

Thus the weighting function controls the contribution to the superlevel set perimeters at different locations in Ω. In 2D, with a monotonically increasing τ, the superlevel set boundaries closer to control point $\mathbf{c}_i$ contribute less to the weighted perimeter. Therefore, compared to the unweighted results, the superlevel set boundaries of w_i curve towards $\mathbf{c}_i$, leading to more local support (see Fig. 6). On the other hand, with a monotonically decreasing τ, the superlevel set boundaries curve away from $\mathbf{c}_i$, since they are subject to less penalty for their length when being farther away from $\mathbf{c}_i$. In 3D, the effect on level surfaces is similar.

4 Numerical optimization

LBC are computed by numerically solving a convex optimization problem. To this end, we first triangulate the domain Ω, such that the triangulation vertices include sample points in the interior of the domain Ω and on the cage, as well as all the control points. Each w_i is represented as a function that is linear within each cell (triangle in 2D or tetrahedron in 3D) and fully determined by its values at the triangulation vertices. In this way, the gradient of w_i is constant on each cell, and linear with respect to the values of w_i at the vertices of the cell. Let $\mathcal{C}$ be the set of cells in the triangulation. Then the target functional (10) can be discretized as

$$\sum_{s\in\mathcal{C}} \sum_{i=1}^{n} \phi_i^s A_s \|\nabla_s w_i\|_2, \tag{14}$$

where A_s is the area/volume of cell s, $\nabla_s w_i$ is the gradient of w_i in cell s, and ϕ_i^s is the value of the weighting function ϕ_i at the centroid of s. For triangulation vertices lying on the cage, the values of w_i are already determined according to the boundary conditions due to the Lagrange and linearity properties. Thus the optimization variables are the values of the coordinate functions $\{w_i \mid i = 1, \ldots, n\}$ at the interior vertices $\{\mathbf{v}_j \mid j = 1, \ldots, m\}$, which we denote by a matrix $\mathbf{W} \in \mathbb{R}^{m\times n}$. Each row $\mathbf{W}^j$ of $\mathbf{W}$ collects the values of all coordinate functions at vertex $\mathbf{v}_j$, while each column $\mathbf{W}_i$ collects the values of function w_i at all interior vertices. Since $\nabla_s w_i$ is affine with respect to $\mathbf{W}_i$, it can be written as $\nabla_s w_i = \mathbf{G}_s \mathbf{W}_i + \mathbf{e}_i^s$ with a sparse matrix $\mathbf{G}_s \in \mathbb{R}^{D\times m}$ $(D = 2, 3)$ and a vector $\mathbf{e}_i^s \in \mathbb{R}^D$ that represents the contribution from the boundary vertices. If no vertex of the cell lies on the cage, then $\mathbf{e}_i^s = \mathbf{0}$. The discretized optimization problem becomes

$$\begin{aligned} \min_{\mathbf{W}} \quad & \sum_{s\in\mathcal{C}} \sum_{i=1}^{n} \phi_i^s A_s \|\mathbf{G}_s \mathbf{W}_i + \mathbf{e}_i^s\|_2 \\ \text{s.t. } & \mathbf{W}\mathbf{K} = \mathbf{B}, \quad \mathbf{W} \geq \mathbf{0}, \end{aligned} \tag{15}$$

where the matrices $\mathbf{K}$ and $\mathbf{B}$ are derived from the reproduction and partition of unity properties,

$$\mathbf{K} = \begin{bmatrix} \mathbf{c}_1^T & 1 \\ \vdots & \vdots \\ \mathbf{c}_n^T & 1 \end{bmatrix} \in \mathbb{R}^{n\times(D+1)}, \mathbf{B} = \begin{bmatrix} \mathbf{v}_1^T & 1 \\ \vdots & \vdots \\ \mathbf{v}_m^T & 1 \end{bmatrix} \in \mathbb{R}^{m\times(D+1)}. \tag{16}$$

Although this is a convex problem, its target function is not smooth. To solve it efficiently, we first convert the problem into a separable form using variable splitting [Combettes and Pesquet 2011]. Specifically, for each cell s we introduce auxiliary variables $\mathbf{x}_i^s \in \mathbb{R}^D (i = 1, \ldots, n)$ to replace the gradient expression $\mathbf{G}_s \mathbf{W}_i + \mathbf{e}_i^s$ in the target function. Moreover, the constraint $\mathbf{W}\mathbf{K} = \mathbf{B}$ in the original optimization problem implies that $\mathbf{W}$ can be written as $\mathbf{W} = \mathbf{Y}\mathbf{M} + \mathbf{H}$, where the rows of $\mathbf{M}$ form an orthonormal basis of the null space of $\mathbf{K}^T$, and $\mathbf{H} \in \mathbb{R}^{m\times n}$ is the least-norm solution to the linear systems $\mathbf{H}\mathbf{K} = \mathbf{B}$. Both $\mathbf{M}$ and $\mathbf{H}$ can be computed efficiently using the SVD of $\mathbf{K}$, due to the small size of $\mathbf{K}$. Then the original problem is converted into an optimization problem about $\mathbf{W}$ and the auxiliary variables $\{\mathbf{x}_i^s\}$, $\mathbf{Y}$:

$$\begin{aligned} \min_{\mathbf{W},\mathbf{X},\mathbf{Y}} \quad & \sum_{s\in\mathcal{C}} \sum_{i=1}^{n} \phi_i^s A_s \|\mathbf{x}_i^s\|_2 + \sigma(\mathbf{W}) \\ \text{s.t.} \quad & \mathbf{X} = \mathbf{G}(\mathbf{Y}\mathbf{M} + \mathbf{H}) + \mathbf{E}, \quad \mathbf{W} = \mathbf{Y}\mathbf{M} + \mathbf{H}, \end{aligned} \tag{17}$$

where the matrices $\mathbf{X}, \mathbf{E} \in \mathbb{R}^{D|\mathcal{C}|\times n}$, and $\mathbf{G} \in \mathbb{R}^{D|\mathcal{C}|\times m}$ collect all $\mathbf{x}_i^s$, $\mathbf{e}_i^s$, and $\mathbf{G}_s$, respectively, and the indicator function

$$\sigma(\mathbf{W}) = \begin{cases} 0, & \text{if } \mathbf{W} \geq \mathbf{0}, \\ +\infty, & \text{otherwise}, \end{cases} \tag{18}$$

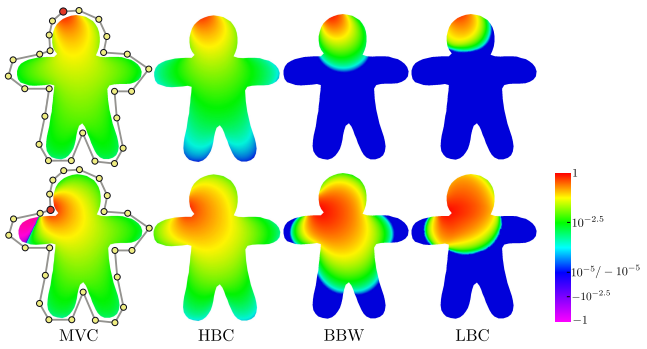


Figure 7: *Comparison of support size between MVC, HBC, BBW, and LBC. The color-coding shows the weight function values for the red control point, indicating better locality of LBC. A full comparison for all control points is presented in the supplementary material.*

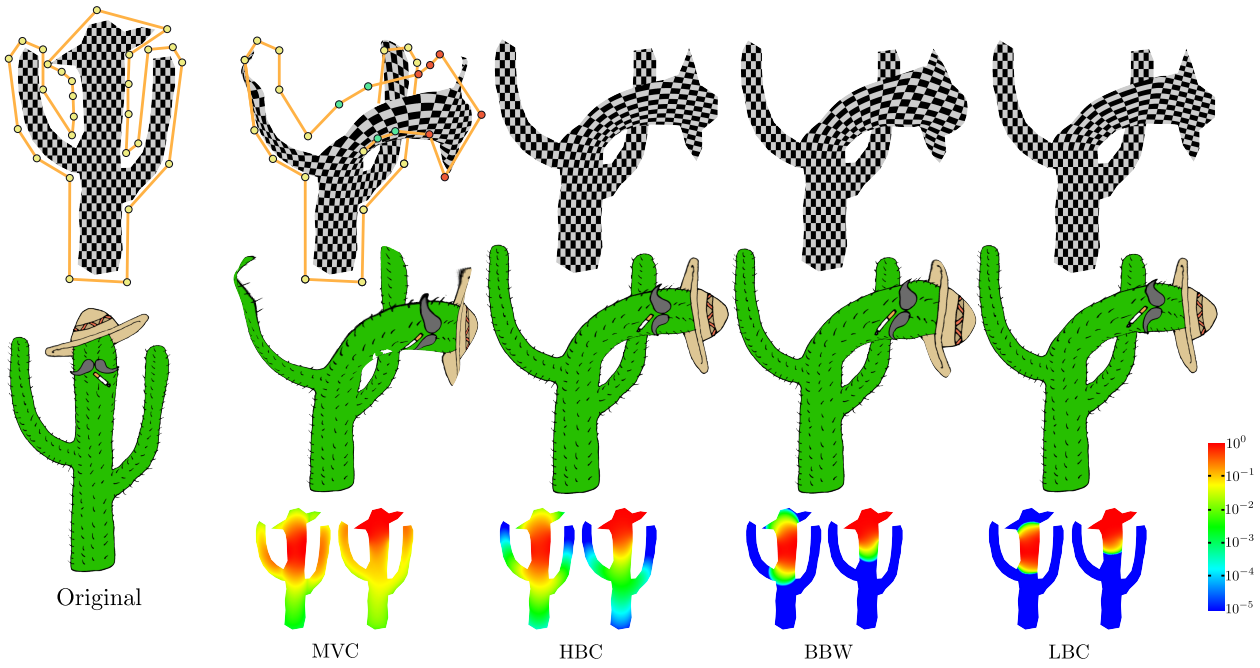


Figure 8: *Deformation of the cactus model with different control weight functions. The control points in red and in green are subject to rigid and non-rigid transformations, respectively. The color-coding shows the absolute sums of weight functions for the green and red control points, respectively. LBC preserve the shape of the hat, since it is only influenced by the red control points, and deformed by the same rigid transformation. Other coordinate schemes distort the hat shape, due to the influence from the green control points.*

enforces the non-negativity constraint on $\mathbf{W}$. Problem (17) has a separable target function and linear side constraints. It can be solved efficiently using the Alternating Direction Method of Multipliers (ADMM) [Boyd et al. 2011], which is popular for separable convex problems. The steps of our ADMM solver are presented in Appendix B. Thanks to the separable structure, each step of the solver handles a set of independent subproblems that can be solved in parallel, which leads to significant speedup on multi-core processors.

5 Results

This section provides examples that demonstrate the main characteristics of LBC, as well as a comparison between LBC and other control weight schemes, including *mean value coordinates* (MVC) [Ju et al. 2005; Hormann and Floater 2006], *harmonic barycentric coordinates* (HBC) [Joshi et al. 2007], *bounded biharmonic weights* (BBW) [Jacobson et al. 2011], and *Poisson-based weight reduction* (PWR) [Landreneau and Schaefer 2010]. In all examples, LBC are computed using weighted TV, where the weight function in Eq. (10) is chosen as $\tau(x) = x^2$.

Locality. Fig. 7 illustrates the locality-inducing effect of TV minimization. The color-coding of the coordinate function values shows the clear advantage of LBC in terms of locality. The comparison for other control points can be found in the supplementary material.

For cage-based deformation, the local influence of LBC provides better preservation of local shapes. This is shown in Fig. 8, where we deform a cactus model. Here, a group of control points around the hat undergoes a rigid transformation, while another set of control points around the neck are subject to a non-rigid transformation. Thanks to the locality of LBC, the control points around the neck have negligible influence on the hat. Hence the hat undergoes the same rigid transformation as the control points around it, and preserves its shape as shown by the checkerboard texture. On the contrary, with other control weight schemes the hat region is also influenced by the control points around the neck, whose non-rigid transformation distorts the hat shape. The non-negativity of LBC also prevents unintuitive deformation, in contrast to the MVC result where negative weights lead to local flipping of the shape. Similarly, Figs. 9 and 10 show the benefit of LBC for enabling local control and producing intuitive deformations. Figs. 1, 11, 12, and 13 illustrate the smoothness and locality of LBC when applied to 3D cage-based

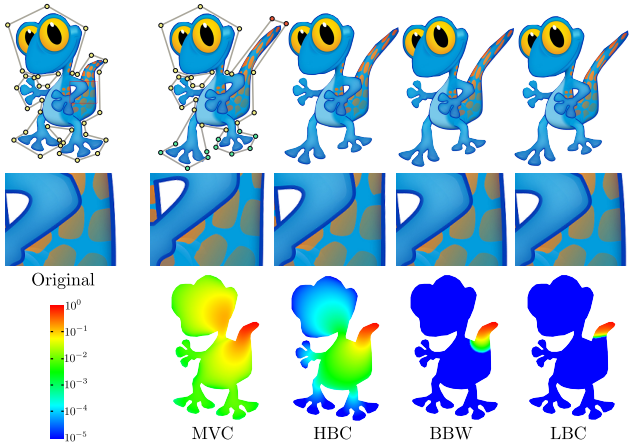


Figure 9: *The gecko image is deformed using control points close to the tail (in red) and the feet (in green). The color-coding shows the absolute sum of control weight functions for the red control points. The elbow and the head remain fixed with LBC, while being deformed by other weight functions. A comparison of the deformations can be found in the supplementary video.*

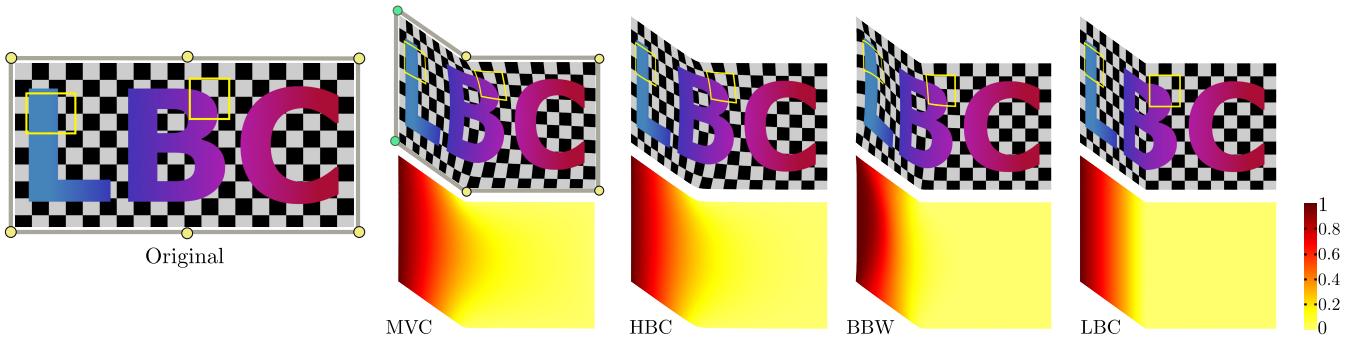


Figure 10: *Deformation of an image by moving two control points (in green), and color-coding for the magnitude of deformation. LBC not only produce a more local deformation, but also preserve the linear features of the texture better.*

deformation. A comparison of 3D deformation using LBC and other control weight schemes can be found in the supplementary video.

The locality of LBC is also verified by statistics of its coordinate function values. Specifically, for a set of control weight functions $w_1, \ldots, w_n$, we denote by $\varphi_i(t)$ the normalized area/volume inside the cage where the magnitude of w_i is no greater than t,

$$\varphi_i(t) = \frac{\eta(\{\mathbf{x} \in \Omega \mid |w_i(\mathbf{x})| \leq t\})}{\eta(\Omega)}, \tag{19}$$

where $\eta(\cdot)$ denotes the area/volume of a set. Then the function

$$P(t) = \frac{1}{n} \sum_{i=1}^{n} \varphi_i(t) \tag{20}$$

indicates the average influence of the control points. Additionally, for a point $\mathbf{x} \in \Omega$, we denote by $\sigma(\mathbf{x})$ the number of control points whose control weight functions at $\mathbf{x}$ are greater than the threshold $10^{-3}/n$ of negligible influence. Then the function

$$Q(k) = \frac{\eta(\{\mathbf{x} \in \Omega \mid \sigma(\mathbf{x}) \geq k\})}{\eta(\Omega)} \quad (\text{for } k \in \mathbb{N}) \tag{21}$$

provides the statistics about the number of control points that influence the cage interior. Figs. 14 and 15 provide the graphs of P and Q for HBC, BBW, and LBC, for the models of Armadillo (Fig. 1), Woody (Fig. 7), Gecko (Fig. 9), and Horse (Fig. 11). The graphs of P show that LBC have small values over large regions, while the graphs of Q show that with LBC each point inside the cage is only influenced by a small number of control points. They clearly indicate the locality of LBC.

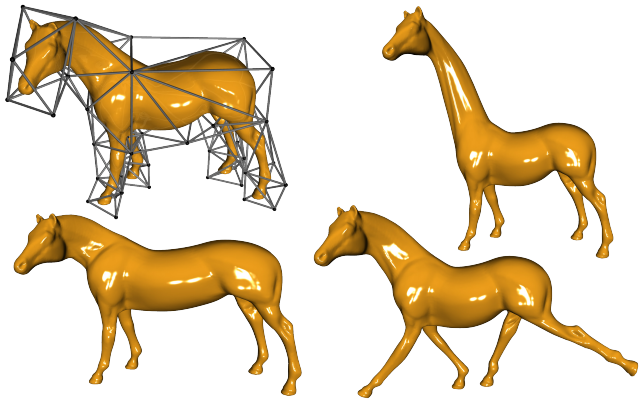


Figure 11: *Cage-based deformation of the horse model illustrates the smoothness of LBC.*

Weight reduction. The above statistics reveal that we can adopt a compact representation for LBC that only stores the control weight values greater than the threshold $10^{-3}/n$. Such a representation not only reduces the amount of required memory storage, but also leads to faster deformation since fewer control points are involved in the interpolation computations. Table 1 lists the memory footprints and deformation timings using the reduced weights from LBC, compared against the dense storage of MVC weights without reduction. Specifically, the dense storage represents MVC weights as a matrix where each columns collects the weights from a control point, and each row collects the weights for a data point. The reduced LBC weights are stored as a sparse matrix of the same dimension, using the compressed sparse column (CSC) format. The deformations are computed by multiplying the weight matrices with a matrix that stores the control point positions. Table 1 shows that the compact representation of LBC brings significant reduction in memory storage and deformation timing for most models. Note that the Logo model requires more memory and time to deform with the reduced weights, because there are only 6 control points, which prevents LBC from being sparse enough to compensate for the storage and computation overheads induced by the CSC data structure.

Compared to other weight reduction schemes such as PWR, LBC are able to improve memory storage and deformation timings without sacrificing the quality of the deformations, thanks to their locality and smoothness (Figs. 12, 16). Note that with PWR, the user needs to specify a set of example poses, as well as the maximum number

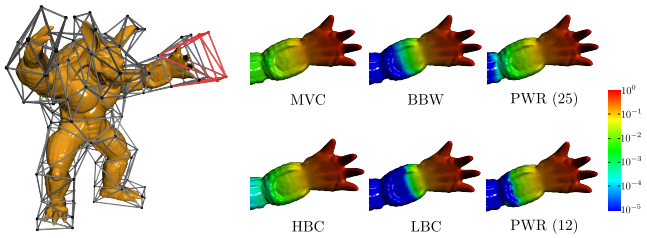


Figure 12: *Local 3D deformation. For the armadillo model, a set of control points near the left hand is moved, with the deformed cage edges shown in red. The color-coding shows the magnitudes of mesh vertex deformations, normalized by the maximum deformation magnitude among all vertices. The PWR parameters* 25 *and* 12 *are the maximum and average values of* Q *in* (21) *for LBC.*

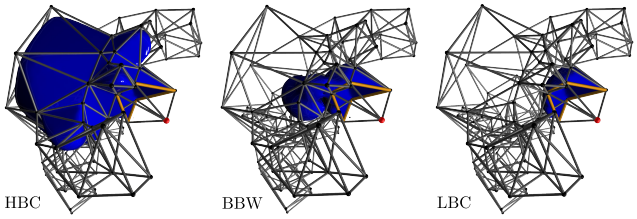


Figure 13: *Comparison of the level set surfaces at value* $10^{-3}/n$ *for the weight functions of a control point (in red), where* n *is the number of control points. TV regularization leads to a smaller area of the level set surface for LBC, resulting in locality.*

of control points that influence an interior point. The example poses we used for PWR can be found in the supplementary material. For the maximum number of control points, we use two values for each model: (i) the maximum value of Q for LBC, which allows PWR to have at least as many weight values as LBC on all interior points; and (ii) the average value of Q for LBC, which requires PWR to achieve the same amount of reduction as LBC. LBC are either as smooth as PWR (Fig. 12), or smoother than PWR (Fig. 16). In both cases, LBC require no more storage than PWR, and achieve better locality.

Linear precision. As a barycentric coordinate scheme, LBC enjoy the linear precision property. This is shown in Fig. 17, where LBC reproduce a linear deformation field from its values at the control points, preserving the linear features of the original image. On the contrary, BBW lack the linear precision property and distort the straight lines of the checkerboard pattern.

Implementation and performance. Our LBC solver is implemented in C++. All linear algebra computations are done using the EIGEN library (`http://eigen.tuxfamily.org`). The triangulation is computed using TRIANGLE (`http://www.cs.cmu.edu/~quake/triangle.html`) in 2D and TETGEN (`http://tetgen.org`) in 3D. The geodesic distance values required for the weighting functions are computed according to [Crane et al. 2013], which amounts to solving a set of linear systems with the same matrix and requires only one factorization of the matrix to solve for different right-hand sides. In each iteration of our solver, independent subproblems are solved in parallel using OpenMP. Table 2 provides the timings of our solver for different models shown in this paper, in comparison to BBW, since BBW also require solving a convex problem. We compute BBW on the same triangulations as LBC. To accelerate the BBW computations, we only compute approximate solutions to the BBW optimization problem by computing the weight functions of each control point independently and normalizing them to enforce partition of unity, as recommended by [Jacobson et al. 2011]. All BBW results are computed using LIBIGL (`http://igl.ethz.ch/projects/libigl/`). The LIBIGL code provides two choices for the backend QP solver, using the active-set method and the MOSEK interior point optimizer (`http://www.mosek.com`), respectively. In all examples, the active set method requires less computation time and produces results that are more local and achieve lower values of the optimization target function. We have also implemented a BBW solver based on the SOCP formulation proposed in [Jacobson et al. 2012], using MOSEK as the backend SOCP solver. It is also outperformed by the active-set code from LIBIGL, in terms of both timing and quality of the results. Thus, all BBW examples are computed with LIBIGL using the active-set solver. The weight functions for different control points are computed in parallel to achieve maximum performance of the BBW solver. Table 2 shows that the speed of our solver is comparable to the BBW solver. Despite the large scale of the optimization problem, the performance of our solver makes it suitable for real-world applications, since the coordinate functions only need to be computed once for each cage. The source code of our solver is available at `https://github.com/bldeng/LBC`.

Model	n	Memory	Timing
ARMADILLO (Fig. 1)	110	16.58%	17.57%
WOODY (Fig. 7)	26	30.80%	41.99%
CACTUS (Fig. 8)	27	23.23%	38.71%
GECKO (Fig. 9)	34	23.93%	31.44%
LOGO (Fig. 10)	6	98.19%	105.46%
HORSE (Fig. 11)	51	20.75%	30.85%

Table 1: *Memory storage and deformation timings using the reduced weights from LBC, relative to those using the dense storage of MVC. Here* n *is the number of control points.*

6 Limitations and future work

Our method is built upon a trade-off between locality and smoothness. TV minimization induces locality, but only guarantees that the result is C^1 almost everywhere. Global C^1 continuity can be achieved with the relaxation in Eq. (7), but at the expense of less

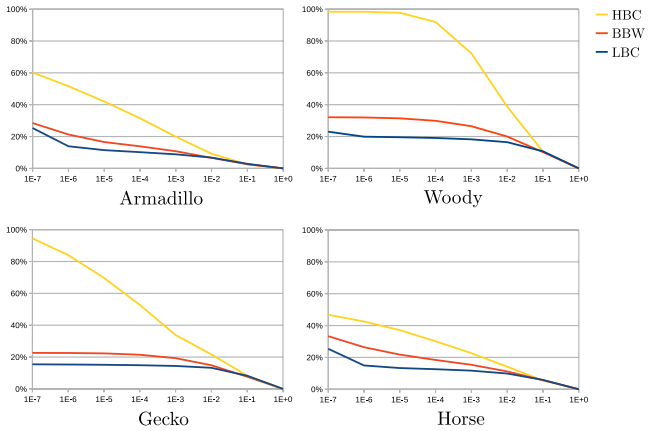


Figure 14: *Graphs of* P *in* (20) *for HBC, BBW and LBC.*

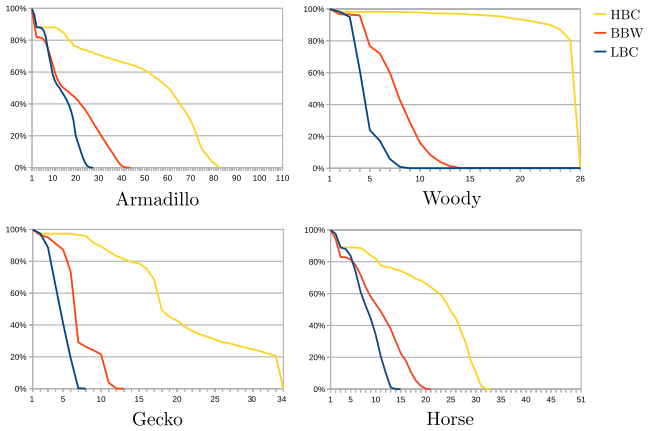


Figure 15: *Graphs of* Q *in* (21) *for HBC, BBW and LBC.*

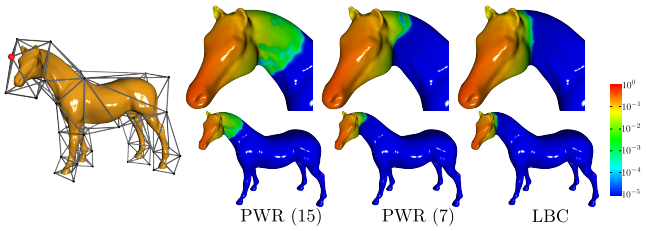


Figure 16: *Comparison of weight reduction using LBC and PWR. The color-coding shows the weight functions for the red control point. The PWR parameters* 15 *and* 7 *are the maximum and average values of* Q *in* (21) *for LBC.*

local results. Neither approach guarantees C^2 continuity. In the future we would like to work on barycentric coordinate schemes with locality as well as higher order continuity.

Since LBC are computed by numerical optimization, we consider the influence from a control point to be negligible when its numerically computed coordinate function value is close enough to zero. It is an open problem to characterize regions where the coordinate functions are exactly zero, which will be interesting future work.

More broadly speaking, despite extensive research results about TV regularization for a single function [Chambolle et al. 2010], not much is known about the case of multiple functions coupled by global constraints like our formulation. Therefore, our work opens up a new avenue of research, which will potentially also provide insight into other problems such as image labeling [Lellmann and Schnörr 2011] and clustering [Bresson et al. 2013].

7 Conclusion

While previous barycentric coordinate schemes define interior points as a combination of all control points or require the user to select an influence region, we introduce a novel method to derive local barycentric coordinates that only depend on a subset of control points while satisfying common requirements for barycentric coordinates. LBC are computed through a convex constrained optimization based on TV. Utilizing the convexity of this formulation, we propose an efficient solver to compute its numerical solutions. The locality of LBC provides various benefits such as lower storage requirement, faster deformations, and local shape preservation, making our method suitable for many applications in computer graphics, such as shape deformation or image composition and warping.

Acknowledgements

We thank Antonin Chambolle and the reviewers for their valuable comments and Hao Li for his help on the video. This work has been supported by NSF of China (61222206, 61303148), NSF of Anhui Province, China (1408085QF119), Specialized Research Fund for the Doctoral Program of Higher Education (20133402120002), the EU FP7 Integrated Project IQmulus (FP7-ICT-2011-318787), Swiss National Science Foundation (grant 200021_137626), and the 100 Talents Program of the Chinese Academy of Sciences.

Model	n	$\lvert\mathcal{V}\rvert$	LBC	BBW
ARMADILLO	110	38,876	1,622.1	5,063.8
WOODY	26	6,069	4.2	3.0
CACTUS	27	12,862	16.0	7.6
GECKO	34	11,554	25.1	6.4
LOGO	6	4,479	0.54	0.75
HORSE	51	24,546	278.9	885.6

Table 2: *Computation timings of LBC and BBW (in seconds), where* n *is the number of control points and* $\lvert\mathcal{V}\rvert$ *is the number of vertices in the triangulation of* Ω.

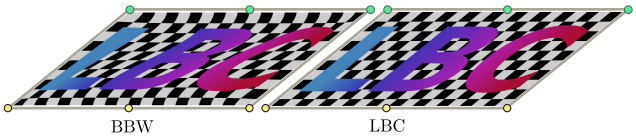


Figure 17: *Linear precision of LBC. For the original image in Fig. 10, its control points are moved according to a linear function. LBC recover the linear deformation function, and preserve the straight lines in the original image. On the other hand, BBW distort the lines, due to their lack of linear precision.*

References

AMBROSIO, L., FUSCO, N., AND PALLARA, D. 2000. *Functions of Bounded Variation and Free Discontinuity Problems*. Oxford University Press.

BACH, F., JENATTON, R., MAIRAL, J., AND OBOZINSKI, G. 2012. Optimization with sparsity-inducing penalties. *Foundations and Trends in Machine Learning 4*, 1, 1–106.

BOYD, S., PARIKH, N., CHU, E., PELEATO, B., AND ECKSTEIN, J. 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning 3*, 1, 1–122.

BRESSON, X., LAURENT, T., UMINSKY, D., AND VON BRECHT, J. 2013. Multiclass total variation clustering. In *Advances in Neural Information Processing Systems*, 1421–1429.

BRUCKSTEIN, A., DONOHO, D., AND ELAD, M. 2009. From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM Rev. 51*, 1, 34–81.

CHAMBOLLE, A., CASELLES, V., CREMERS, D., NOVAGA, M., AND POCK, T. 2010. An introduction to total variation for image analysis. *Theoretical Foundations and Numerical Methods for Sparse Recovery 9*, 263–340.

CHAN, T. F., AND VESE, L. A. 2001. Active contours without edges. *IEEE Trans. Image Process. 10*, 2, 266–277.

CHAN, T., ESEDOGLU, S., PARK, F., AND YIP, A. 2006. Total variation image restoration: Overview and recent developments. In *Handbook of Mathematical Models in Computer Vision*, N. Paragios, Y. Chen, and O. Faugeras, Eds. Springer US, 17–31.

CHAN, R., CHAN, T., AND YIP, A. 2011. Numerical methods and applications in total variation image restoration. In *Handbook of Mathematical Methods in Imaging*, O. Scherzer, Ed. Springer New York, 1059–1094.

CHARTRAND, R. 2007. Exact reconstruction of sparse signals via nonconvex minimization. *IEEE Signal Process. Lett. 14*, 10, 707–710.

COMBETTES, P. L., AND PESQUET, J.-C. 2011. Proximal splitting methods in signal processing. In *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, H. H. Bauschke, R. S. Burachik, P. L. Combettes, V. Elser, D. R. Luke, and H. Wolkowicz, Eds. Springer, 185–212.

CRANE, K., WEISCHEDEL, C., AND WARDETZKY, M. 2013. Geodesics in heat: A new approach to computing distance based on heat flow. *ACM Trans. Graph. 32*, 5, 152:1–152:11.

DASGUPTA, G., AND WACHSPRESS, E. L. 2008. Basis functions for concave polygons. *Comput. Math. Appl. 56*, 2, 459–468.

DESBRUN, M., MEYER, M., AND ALLIEZ, P. 2002. Intrinsic parameterizations of surface meshes. *Comput. Graph. Forum 21*, 4, 209–218.

ELSEY, M., AND ESEDOGLU, S. 2009. Analogue of the total variation denoising model in the context of geometry processing. *Multiscale Modeling & Simulation 7*, 4, 1549–1573.

EVANS, L. C., AND GARIEPY, R. F. 1992. *Measure Theory and Fine Properties of Functions*. CRC Press.

FARBMAN, Z., HOFFER, G., LIPMAN, Y., COHEN-OR, D., AND LISCHINSKI, D. 2009. Coordinates for instant image cloning. *ACM Trans. Graph. 28*, 3, 67:1–67:9.

FLOATER, M. S. 2003. Mean value coordinates. *Comput. Aided Geom. Des. 20*, 1, 19–27.

GARCÍA, F. G., PARADINAS, T., COLL, N., AND PATOW, G. 2013. *Cages: A multilevel, multi-cage-based system for mesh deformation. *ACM Trans. Graph. 32*, 3, 24:1–24:13.

GE, D., JIANG, X., AND YE, Y. 2011. A note on the complexity of L_p minimization. *Mathematical Programming 129*, 2, 285–299.

GOLDSTEIN, T., AND OSHER, S. 2009. The split Bregman method for L1-regularized problems. *SIAM J. Imaging Sci. 2*, 2, 323–343.

GOLDSTEIN, T., BRESSON, X., AND OSHER, S. 2010. Geometric applications of the split Bregman method: Segmentation and surface reconstruction. *J. Sci. Comput. 45*, 1-3, 272–293.

GRASMAIR, M., 2010. A coarea formula for anisotropic total variation regularisation. NRN Report No. 103.

HIYOSHI, H., AND SUGIHARA, K. 2000. Voronoi-based interpolation with higher continuity. In *Proc. SoCG*, 242–250.

HORMANN, K., AND FLOATER, M. S. 2006. Mean value coordinates for arbitrary planar polygons. *ACM Trans. Graph. 25*, 4, 1424–1441.

HORMANN, K., AND SUKUMAR, N. 2008. Maximum entropy coordinates for arbitrary polytopes. *Comput. Graph. Forum 27*, 5, 1513–1520.

JACOBSON, A., BARAN, I., POPOVIĆ, J., AND SORKINE, O. 2011. Bounded biharmonic weights for real-time deformation. *ACM Trans. Graph. 30*, 4, 78:1–78:8.

JACOBSON, A., WEINKAUF, T., AND SORKINE, O. 2012. Smooth shape-aware functions with controlled extrema. *Comput. Graph. Forum 31*, 5, 1577–1586.

JOSHI, P., MEYER, M., DEROSE, T., GREEN, B., AND SANOCKI, T. 2007. Harmonic coordinates for character articulation. *ACM Trans. Graph. 26*.

JU, T., SCHAEFER, S., AND WARREN, J. 2005. Mean value coordinates for closed triangular meshes. *ACM Trans. Graph. 24*, 3, 561–566.

KRANTZ, S. G., AND PARKS, H. R. 2002. *A Primer of Real Analytic Functions*, 2nd ed. Birkhäuser.

LANDRENEAU, E., AND SCHAEFER, S. 2010. Poisson-based weight reduction of animated meshes. *Comput. Graph. Forum 29*, 6, 1945–1954.

LELLMANN, J., AND SCHNÖRR, C. 2011. Continuous multiclass labeling approaches and algorithms. *SIAM J. Imaging Sci. 4*, 4, 1049–1096.

LI, X.-Y., AND HU, S.-M. 2013. Poisson coordinates. *IEEE Trans. Vis. Comput. Graph. 19*, 2, 344–352.

LI, Z., LEVIN, D., DENG, Z., LIU, D., AND LUO, X. 2010. Cage-free local deformations using green coordinates. *Visual Comput. 26*, 6-8, 1027–1036.

LI, X.-Y., JU, T., AND HU, S.-M. 2013. Cubic mean value coordinates. *ACM Trans. Graph. 32*, 4, 126:1–126:10.

LIPMAN, Y., KOPF, J., COHEN-OR, D., AND LEVIN, D. 2007. GPU-assisted positive mean value coordinates for mesh deformations. In *Proc. SGP*, 117–123.

LIPMAN, Y., LEVIN, D., AND COHEN-OR, D. 2008. Green coordinates. *ACM Trans. Graph. 27*, 3, 78:1–78:10.

MÖBIUS, A. F. 1827. *Der barycentrische Calcul: ein neues Hülfsmittel zur analytischen Behandlung der Geometrie*. Barth.

OSSERMAN, R. 1978. The isoperimetric inequality. *Bulletin of the American Mathematical Society 84*, 6, 1182–1238.

PINKALL, U., AND POLTHIER, K. 1993. Computing discrete minimal surfaces and their conjugates. *Experimental Mathematics 2*, 1, 15–36.

RUDIN, L. I., OSHER, S., AND FATEMI, E. 1992. Nonlinear total variation based noise removal algorithms. *Phys. D 60*, 1-4, 259–268.

RUSTAMOV, R. M. 2011. Multiscale biharmonic kernels. *Comput. Graph. Forum 30*, 5, 1521–1531.

SCHAEFER, S., JU, T., AND WARREN, J. 2007. A unified, integral construction for coordinates over closed curves. *Comput. Aided Geom. Des. 24*, 8-9, 481–493.

SIBSON, R. 1981. A brief description of natural neighbour interpolation. In *Interpreting Multivariate Data*, V. Barnett, Ed. John Wiley & Sons, 21–36.

TAKAYAMA, K., SORKINE, O., NEALEN, A., AND IGARASHI, T. 2010. Volumetric modeling with diffusion surfaces. *ACM Trans. Graph. 29*, 6, 180:1–180:8.

WAN, F. Y. 1995. *Introduction to the Calculus of Variations and Its Applications*, 2nd ed. Chapman & Hall.

WEBER, O., AND GOTSMAN, C. 2010. Controllable conformal maps for shape deformation and interpolation. *ACM Trans. Graph. 29*, 4, 78:1–78:11.

WEBER, O., BEN-CHEN, M., AND GOTSMAN, C. 2009. Complex barycentric coordinates with applications to planar shape deformation. *Comput. Graph. Forum 28*, 2, 587–597.

WEBER, O., BEN-CHEN, M., GOTSMAN, C., AND HORMANN, K. 2011. A complex view of barycentric mappings. *Comput. Graph. Forum 30*, 5, 1533–1542.

WEBER, O., PORANNE, R., AND GOTSMAN, C. 2012. Biharmonic coordinates. *Comput. Graph. Forum 31*, 8, 2409–2422.

WEISS, P., BLANC-FRAUD, L., AND AUBERT, G. 2009. Efficient schemes for total variation minimization under constraints in image processing. *SIAM J. Sci. Comput. 31*, 3, 2047–2080.

WU, C., AND TAI, X.-C. 2010. Augmented Lagrangian method, dual methods, and split Bregman iteration for ROF, vectorial TV, and high order models. *SIAM J. Imaging Sci. 3*, 3, 300–339.

A Euler–Lagrange equations for LBC

To derive the optimality condition for Problem (5), we get rid of the inequality constraint $w_i \geq 0$ by introducing functions $q_i : \Omega \mapsto \mathbb{R}$ $(i = 1, \ldots, n)$ together with constraints $w_i(\mathbf{x}) = [q_i(\mathbf{x})]^2 \ \forall \mathbf{x} \in \Omega$. Problem (5) is then converted into a constrained optimization problem about $\{w_i\}$ and $\{q_i\}$ with pointwise equality constraints,

$$\begin{aligned} \min_{\substack{w_1,\ldots,w_n \\ q_1,\ldots,q_n}} \ & \sum_{i=1}^{n} \int_\Omega |\nabla w_i| \\ \text{s.t.} \ & \sum_{i=1}^{n} w_i(\mathbf{x})\mathbf{c}_i = \mathbf{x}, \ \sum_{i=1}^{n} w_i(\mathbf{x}) = 1, \ \forall \mathbf{x} \in \Omega, \\ & w_i(\mathbf{x}) = [q_i(\mathbf{x})]^2, \ \forall \mathbf{x} \in \Omega, \\ & w_i(\mathbf{x}) = \psi_i(\mathbf{x}), \ \forall \mathbf{x} \in \partial\Omega, \end{aligned} \tag{22}$$

where $\partial\Omega$ is the boundary of Ω, and the function $\psi_i : \partial\Omega \mapsto \mathbb{R}$ represents the boundary values of w_i derived from the Lagrange and linearity properties. Then an optimal solution function w_i should be part of a stationary point for the functional [Wan 1995]

$$\begin{aligned} & I[w_1, \ldots, w_n, q_1, \ldots, q_n, \boldsymbol{\gamma}_1, \gamma_2, \gamma_3] \\ = \ & \int_\Omega [\sum_{i=1}^{n} |\nabla w_i| + \boldsymbol{\gamma}_1 \cdot (\sum_{i=1}^{n} w_i \mathbf{c}_i - \mathbf{x}) + \gamma_2 (\sum_{i=1}^{n} w_i - 1) \\ & + \gamma_3 (w_i - q_i^2) \ ] \, d\mathbf{x}, \end{aligned} \tag{23}$$

where $\boldsymbol{\gamma}_1 : \Omega \mapsto \mathbb{R}^N$, $\gamma_2 : \Omega \mapsto \mathbb{R}$, and $\gamma_3 : \Omega \mapsto \mathbb{R}$ are the Lagrange multiplier functions. The Euler-Lagrange equation of the functional in (23) for w_i is

$$\nabla \cdot \frac{\nabla w_i}{|\nabla w_i|} + \boldsymbol{\gamma}_1 \cdot \mathbf{c}_i + \gamma_2 + \gamma_3 = 0. \tag{24}$$

B ADMM solver for LBC

Using ADMM, we solve Problem (15) by searching for a saddle point of its augmented Lagrangian function

$$\begin{aligned} & L(\mathbf{W}, \mathbf{X}, \mathbf{Y}, \boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2) \\ = & \sum_{s \in \mathcal{C}} \sum_{i=1}^{n} \phi_i^s A_s \|\mathbf{x}_i^s\|_2 + \sigma(\mathbf{W}) + \boldsymbol{\lambda}_1 \bullet (\mathbf{X} - \mathbf{G}\mathbf{Y}\mathbf{M} - \mathbf{J}) \\ & + \boldsymbol{\lambda}_2 \bullet (\mathbf{W} - \mathbf{Y}\mathbf{M} - \mathbf{H}) + \frac{\mu}{2} (\|\mathbf{X} - \mathbf{G}\mathbf{Y}\mathbf{M} - \mathbf{J}\|_F^2 \\ & + \|\mathbf{W} - \mathbf{Y}\mathbf{M} - \mathbf{H}\|_F^2), \end{aligned} \tag{25}$$

where $\mathbf{J} = \mathbf{G}\mathbf{H} + \mathbf{E}$, the matrices $\boldsymbol{\lambda}_1$, $\boldsymbol{\lambda}_2$ are Lagrange multipliers, $\|\cdot\|_F$ is the Frobenius norm, $\bullet$ denotes the inner product between the vectorization of two matrices, and $\mu > 0$ is a penalty parameter.

Solver. ADMM iteratively updates the variables and multipliers until convergence. Let $\mathbf{W}^{(k)}, \mathbf{X}^{(k)}, \mathbf{Y}^{(k)}, \boldsymbol{\lambda}_1^{(k)}, \boldsymbol{\lambda}_2^{(k)}$ be their values at iteration k. Then each iteration consists of the following steps:

1. Update $\mathbf{W}, \mathbf{X}$:

$$(\mathbf{W}^{(k+1)}, \mathbf{X}^{(k+1)}) = \operatorname{argmin}_{\mathbf{W},\mathbf{X}} \ L(\mathbf{W}, \mathbf{X}, \mathbf{Y}^{(k)}, \boldsymbol{\lambda}_1^{(k)}, \boldsymbol{\lambda}_2^{(k)}).$$

2. Update $\mathbf{Y}$:

$$\mathbf{Y}^{(k+1)} = \operatorname{argmin}_{\mathbf{Y}} \ L(\widetilde{\mathbf{W}}^{(k+1)}, \widetilde{\mathbf{X}}^{(k+1)}, \mathbf{Y}, \boldsymbol{\lambda}_1^{(k)}, \boldsymbol{\lambda}_2^{(k)}),$$

where $\widetilde{\mathbf{W}}^{(k+1)} = \alpha \mathbf{W}^{(k+1)} - (\alpha - 1)(\mathbf{Y}^{(k)}\mathbf{M} + \mathbf{H})$, $\widetilde{\mathbf{X}}^{(k+1)} = \alpha \mathbf{X}^{(k+1)} - (\alpha - 1)(\mathbf{G}\mathbf{Y}^{(k)}\mathbf{M} + \mathbf{J})$, and $\alpha \in [1.5, 1.8]$ is an over-relaxation parameter [Boyd et al. 2011].

3. Update $\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2$:

$$\begin{aligned} \boldsymbol{\lambda}_1^{(k+1)} &= \boldsymbol{\lambda}_1^{(k)} + \mu(\widetilde{\mathbf{X}}^{(k+1)} - \mathbf{G}\mathbf{Y}^{(k+1)}\mathbf{M} - \mathbf{J}), \\ \boldsymbol{\lambda}_2^{(k+1)} &= \boldsymbol{\lambda}_2^{(k)} + \mu(\widetilde{\mathbf{W}}^{(k+1)} - \mathbf{Y}^{(k+1)}\mathbf{M} - \mathbf{H}). \end{aligned}$$

Thanks to the convexity of the problem, this solver is guaranteed to converge to the solution [Boyd et al. 2011]. Convergence is indicated by the primal residual $\mathbf{r}_p$ and the dual residual $\mathbf{r}_d$ that are small enough, where

$$\mathbf{r}_p = \begin{bmatrix} \mathbf{X}^{(k+1)} - \mathbf{G}\mathbf{Y}^{(k+1)}\mathbf{M} - \mathbf{J} \\ \mathbf{W}^{(k+1)} - \mathbf{Y}^{(k+1)}\mathbf{M} - \mathbf{H} \end{bmatrix}, \tag{26}$$

$$\mathbf{r}_d = \mu \begin{bmatrix} \mathbf{G}(\mathbf{Y}^{(k+1)} - \mathbf{Y}^{(k)})\mathbf{M} \\ (\mathbf{Y}^{(k+1)} - \mathbf{Y}^{(k)})\mathbf{M} \end{bmatrix}. \tag{27}$$

Thus, the solver terminates when $\|\mathbf{r}_p\|_F \leq \epsilon_p$ and $\|\mathbf{r}_p\|_F \leq \epsilon_d$, where ϵ_p, ϵ_d are convergence thresholds given by the user.

Solutions. Step 1 consists of the following subproblems

$$\min_{\mathbf{x}_i^s} \ \frac{\mu}{2} \|\mathbf{x}_i^s - \mathbf{a}_{s,i}^{(k)}\|_2^2 + \phi_s A_s \|\mathbf{x}_i^s\|_2, \tag{28}$$

$$\min_{\mathbf{W}} \ \frac{\mu}{2} \|\mathbf{W} - \mathbf{Q}^{(k)}\|_F^2 + \sigma(\mathbf{W}), \tag{29}$$

which can be solved in parallel, where $\mathbf{a}_{s,i}^{(k)}$ are the components of matrix $\mathbf{G}\mathbf{Y}^{(k)}\mathbf{M} + \mathbf{J} - \boldsymbol{\lambda}_1^{(k)}/\mu$ that correspond to $\mathbf{x}_i^s$, and $\mathbf{Q}^{(k)} = \mathbf{Y}^{(k)}\mathbf{M} + \mathbf{H} - \boldsymbol{\lambda}_2^{(k)}/\mu$. These problems have closed-form solutions

$$\mathbf{x}_i^s = \begin{cases} \mathbf{0}, & \text{if } \|\mathbf{a}_{s,i}^{(k)}\| \leq \frac{\phi_s A_s}{\mu}, \\ \left(1 - \frac{\phi_s A_s}{\mu \|\mathbf{a}_{s,i}^{(k)}\|}\right) \mathbf{a}_{s,i}^{(k)}, & \text{otherwise}, \end{cases} \tag{30}$$

$$\mathbf{W}_{i,j} = \max(0, \mathbf{Q}_{i,j}^{(k)}), \tag{31}$$

where $\mathbf{W}_{i,j}, \mathbf{Q}_{i,j}^{(k)}$ are the (i,j)-th elements of $\mathbf{W}$, $\mathbf{Q}^{(k)}$, respectively. In Step 2, $\mathbf{Y}^{(k+1)}$ is the solution to a set of symmetric positive definite linear systems with matrix $\mathbf{G}^T\mathbf{G} + \mathbf{I}$ and right-hand sides $[\mathbf{G}^T(\widetilde{\mathbf{X}}^{(k+1)} - \mathbf{J} + \boldsymbol{\lambda}_1^{(k)}/\mu) + \widetilde{\mathbf{W}}^{(k+1)} - \mathbf{H} + \boldsymbol{\lambda}_2^{(k)}/\mu]\mathbf{M}^T$. Note that the linear system matrix remains fixed in all iterations. So we precompute the sparse Cholesky factors of the system matrix once, and use them to solve the system in subsequent iterations. In all iterations, different right-hand sides are solved in parallel.

Parameters. The solver requires user-specified values of the penalty weight μ, the over-relaxation parameter α, the convergence thresholds ϵ_p, ϵ_d, as well as initial values for the variables and Lagrange multipliers. For all examples in this paper, we rescale the models to have bounding box diameter 1, and set $\mu = 10$. The other parameters are chosen as $\alpha = 1.65$, $\epsilon_p = N_p \times 10^{-6}$, $\epsilon_d = N_d \times 10^{-5}$, where N_p and N_d are the dimension of $\mathbf{r}_p$ and $\mathbf{r}_d$, respectively. The initial values of variables and multipliers are set to $\mathbf{Y}^{(0)} = \mathbf{0}$, $\boldsymbol{\lambda}_1^{(0)} = \boldsymbol{\lambda}_2^{(0)} = \mathbf{0}$.