Local Deformation for Interactive Shape Editing

Honglin Chen  
Columbia University  
New York City, NY, USA  
honglin.chen@columbia.edu

Changxi Zheng  
Columbia University  
New York City, NY, USA  
cxz@cs.columbia.edu

Kevin Wampler  
Adobe Research  
Seattle, WA, USA  
kwampler@adobe.com

Figure 1: Our method enables the user to edit shapes in an interactive and physically plausible way. The edit is local, meaning that the user can focus on one region of the complex scene without worrying about inadvertent changes elsewhere. To visualize the locality, in the rightmost figure we highlight the regions where the vertex displacement is larger than $10^{-3}$ in red. (Undeformed scene shapes thanks to [Zhang et al. 2022])

ABSTRACT
We introduce a novel regularization for localizing an elastic-energy-driven deformation to only those regions being manipulated by the user. Our local deformation features a natural region of influence, which is automatically adaptive to the geometry of the shape, the size of the deformation and the elastic energy in use. We further propose a three-block ADMM-based optimization to efficiently minimize the energy and achieve interactive frame rates. Our approach avoids the artifacts of other alternative methods, is simple and easy to implement, does not require tedious control primitive setup and generalizes across different dimensions and elastic energies. We demonstrate the effectiveness and efficiency of our localized deformation tool through a variety of local editing scenarios, including 1D, 2D, 3D elasticity and cloth deformation.

CCS CONCEPTS
• Computing methodologies → Mesh geometry models.

KEYWORDS
Local control, shape deformation, elasticity, sparsity, ADMM.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SIGGRAPH ’23 Conference Proceedings, August 6–10, 2023, Los Angeles, CA, USA  
© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.  
https://doi.org/10.1145/358432.3591501

1 INTRODUCTION
Local deformation is a core component in modeling and animation. In a localized deformation, only the parts of the shape near where the user is currently manipulating move—everything else stays still, ensuring that the user can focus entirely on one region of the shape without worrying about inadvertent changes elsewhere. However, existing localized deformation tools tend to have practical impediments for interactive design: they are either too slow to run, unaware of the geometry, introduce artifacts, or require a careful control point setup.

When global deformation is acceptable, a widely useful approach is to solve for the deformation by minimizing an elastic energy defined over the shape, subject to positional constraints derived from the user’s input. This paradigm has many advantages. The deformation accounts for the geometry of the shape, generalizes well to 2D, 3D, and cloth, and the elastic energy can be used to model a wide range of both real-world and stylized materials. Unfortunately elastic energy minimization is by its nature global, and jointly solves for all the degrees of freedom in the shape. This necessitates a rigging step to “pin down” certain aspects of the deformation lest the optimizer move them. This limits the applicability of such methods to situations where a suitable rig is available, or the region of influence for a deformation is known in advance.
We seek to combine the advantages of elastic energy minimization with the locality of sculpting-style tools. In doing so, we also want the locality of a deformation to be automatic, natural and efficient. The locality of an edit should be automatic in that the region of influence (ROI) of the deformation should scale automatically depending on the size of the desired deformation. In addition, although we do not require a rig, in cases where a rig or other constraints have been placed on the shape, the ROI needs to automatically adapt to them. We further want the notion of locality to be natural in that it adjusts to both the geometry of the shape and the elastic energy driving its deformation, where changes to either will lead to a fitting change in the ROI. Finally, we require that the method be fast enough to run in real time.

We achieve this with the following contributions:

- We introduce a novel deformation regularizer, called a smoothly clamped $t_1$ (SC-L1) loss which augments an elastic energy with a notion of locality. SC-L1 regularization is simple to implement, and avoids artifacts of previous methods.
- We enable real-time localized deformation with an ADMM-based optimization algorithm for SC-L1-regularized deformation which is significantly faster than prior work using a group lasso regularizer.

We illustrate the utility of SC-L1 regularization on a wide range of examples, including multiple different elastic energies, 1D curves, 2D and 3D meshes, and cloth. This provides a localized deformation tool which avoids artifacts of other regularizers, is easy to implement, generalizes across different dimensions and material models, and performs fast enough to run in real time.

2 RELATED WORK

Shape deformation algorithms in computer graphics have typically fallen into one of two categories, which we will call the direct approach and the optimization approach. In the direct approach, the shape’s deformation is an explicit function of the user’s input, often either by modifying some high-level parameterization or by applying a pre-specified deformation field. In the optimization approach, the shape deformation is an indirect product of the user’s input combined with an elastic energy, and the deformation itself is only known after an optimization process has converged to minimize this energy. These two approaches differ in how (and often whether) they enforce the locality of a deformation.

2.1 Locality in High-Level Parameterizations

A time-honored and common approach for localized deformation is the direct manipulation of high-level parameters. From the onset, a notion of locality is baked into these parameters, which can directly encode the shape itself, often using splines [Hoppe et al. 1993] or a “rig” to control a shape’s deformation, as with cage-based generalized barycentric coordinates [Joshi et al. 2007; Lipman et al. 2008], linear-blend skinning [Jacobson et al. 2011; Magnenat-Thalmann et al. 1989; Wang et al. 2015], lattice deformers [Cootes et al. 1992; Sederberg and Parry 1996], wire curves [Singh and Fiume 1998], or learned skinning weights [Genova et al. 2020], to list a few.

Approaches of this nature, although popular in computer graphics, have a few disadvantages. Firstly, since the locality is baked into the parameterization, it cannot easily adapt based on changes in the deformation. Secondly, without an optimization step, even in cases where a capable elastic energy is within reach there is no way to incorporate it into the deformation.

2.2 Locality in Deformation Fields

Instead of providing localized deformation via pre-chosen parameters, it is also possible to define a localized deformation field, then apply it to a shape. These methods often follow a “sculpting” metaphor, and include simple move, scale, pinch, and twist edits as well as more sophisticated operations [Cani and Angelidis 2006]. Recently, De Goes and James [2017] introduced regularized Kelvinlets, which provides real-time localized volumetric control based on the regularized closed-form solutions of linear elasticity. These closed-form solutions were later extended to handle dynamic secondary motions [De Goes and James 2018], sharp deformation [de Goes and James 2019] and anisotropic elasticity [Chen and Desbrun 2022].

These approaches allow for local deformation with real-time feedback. However, as they are designed for digital sculpting, these methods usually require the user to explicitly pick the falloff of the brushes. Furthermore, these methods are usually based on Euclidean distance, unaware of the shape’s geometry. In contrast, our method is shape-aware and enables automatic dynamic region of influence with interactive feedback.

2.3 Localized Optimization via an ROI

The most natural formulation of optimization-based deformation editing is by solving globally for the entire shape’s deformation at once [Shtengel et al. 2017; Smith et al. 2019; Zhu et al. 2018]. Nevertheless, there are methods attempting to enforce locality in the shape optimization process. Previous methods of this sort have often computed the ROI of a manipulation as a preprocessing step, then restricting the optimization to only move parts of the shape within this ROI. The ROI can be taken as an input [Alexa 2006] or based on a small amount of user markup [Kho and Garland 2005; Luo et al. 2007; Zimmermann et al. 2007]. Other methods combine deformation energies with handle-based systems, including skeleton rigs [Hahn et al. 2012; Jacobson et al. 2012; Kavan and Sorkine 2012] and cages [Ben-Chen et al. 2009].

Unfortunately, in many contexts, the ROI is hard or impossible to know in advance. This is particularly the case when constraints are involved, or where it is not known in advance if a deformation will be small (best fitting a small ROI) or large (best fitting a large ROI). In addition, the correct ROI may also depend on the elastic energy driving the deformation, thus difficult to account for when the ROI calculation is decoupled in a separate step.
Another class of methods adds a sparsity-induced regularization term to produce local deformation which is smooth, shape-aware and has automatically adaptive region of influence without the need of additional handles. Here we highlight all the handles that have been moved in yellow. For the other alternatives, $l_1$-norm-based method (b) contains artifacts due to the use of $l_1$ norm (see the green highlighted regions); regularized Kelvinlets technique (c) is based on Euclidean distance and unaware of the geometry (see the blue highlighted regions); and biharmonic coordinates approach (d) requires a careful placement of additional handles (green) to explicitly control the region of influence. Note that the use of $l_1$-norm (b) leads to either small global motions and minor artifacts (b2, smaller $l_1$-norm) or no global motions but significant artifacts (b3, larger $l_1$-norm). Thus [Chen et al. 2017] mitigates it with a smoothing regularizer (b1), but this changes the elastic energy and it no longer deforms like a localized ARAP.

2.4 Localized Optimization via Sparsity Norms

Sparsity-inducing norms, such as the smooth $l_0$ norm, have been widely applied to many domains and problems including medical image reconstruction [Xiang et al. 2022], sparse component analysis [Mohimani et al. 2007] and UV mapping [Poranne et al. 2017]. To allow an adaptive ROI while preserving the benefits of optimization-based deformation, a few works have adopted a sparsity-inducing norm, typically a $\|x\|_1$ or $\|x\|_2$ norm, in their energy, which is then minimized by Alternating Direction Method of Multipliers (ADMM) [Boyd et al. 2011; Peng et al. 2018; Zhang et al. 2019] or Augmented Lagrangian Method (ALM) [Bertsekas 1996]. We refer the readers to [Xu et al. 2015] for a survey on sparsity in geometry modelling and processing.

Several methods of this variety rely on sparsity-inducing norm formulated as a sum of $\|x\|_2$ norms, referred to as $f_{2,1}$ or $l_1/l_2$ norms, or a group lasso penalty. They have been applied in a pre-processing phase to compute sparse deformation modes for interactive local control [Brandt and Hildebrandt 2017; Deng et al. 2013; Neumann et al. 2013]. However, the deformation is limited by the linear deformation modes and thus struggles with large deformation.

Another class of methods adds a sparsity-induced regularization to an elastic energy optimization to achieve local deformation. Gao et al. [2012] applied different $l_p$ sparsity norms to the as-rigid-as-possible energy [Sorkine and Alexa 2007] to create various deformation styles. Recently, Chen et al. [2017] used $f_{2,1}$ regularization on vertex positions to locally control the deformation. However, their direct use of the $f_{2,1}$ norm will create artifacts when the control point is not on the boundary of the shape (see Fig. 3(b) and Fig. 13(b)). Moreover, their method requires one ADMM solve in each global iteration, which renders the optimization less efficient and slow in runtime. Also, their framework is limited to 2D deformation with ARAP energy only, while our framework generalizes across dimensions and a variety of energy models.

Our algorithm, inspired by sparsity-seeking regularizers such as that used by [Chen et al. 2017; Fan and Li 2001], addresses these shortcomings. In particular, we propose a simple and novel sparsity-inducing norm that eliminates artifacts arising from the $f_{2,1}$ norm, and our efficient optimization scheme leads to interactive performance.

3 OVERVIEW

The main idea of our method is to use a novel regularization term to produce local deformation with a dynamic region of influence (ROI). Our method takes a triangle/tetrahedral mesh (or a 1D polyline) and a set of selected vertices as control handles as input. The output of our method is a deformed shape where the deformation is both local and natural and the ROI is automatically adaptive to the deformation. Here the “locality” implies that a handle only dominates its nearby areas without affecting the regions far away.

3.1 SC-L1 Regularization

We suggest a sparsity-inducing regularization term to produce natural local deformation. This regularizer is applied per-vertex to $\hat{V}_i - \bar{V}_i$ to bias each vertex deformed position $\hat{V}_i$ to exactly match its initial rest position $\bar{V}_i$ except in isolated regions of the shape. The most obvious choice for this regularization would be to enforce sparsity either with an $l_1$-norm, or with a group lasso / $f_{2,1}$ regularization defined as $\sum_i \|\hat{V}_i\|_2$ as in [Chen et al. 2017].
However, direct use of a $\ell_{2,1}$ regularization term leads to artifacts, due to the fact that the $\ell_{2,1}$-norm competes with the elastic energy by dragging all the vertices towards their original positions. This results in undesired distortion near the deformation handles (see Fig. 3(b) and Fig. 13(b)), making the deformed region look unnatural. Previous $\ell_{2,1}$-based methods [Chen et al. 2017] focus specifically on ARAP-like deformation, and attempt to alleviate this artifact by adding a Laplacian smoothness term and a weighting term based on biharmonic distance. Unfortunately, as seen in Fig. 3 at the ends of the octocat’s tentacles, artifacts can arise in regions quite close to the deformation handles, and there is not necessarily any setting of these parameters which alleviates these artifacts without oversmoothing the entire deformation.

Inspired by “folded concave” losses in statistical regression [Fan and Li 2001; Zhang 2010] and the use of the $\ell_{2,1}$ loss in deformation [Chen et al. 2017], we propose using a smoothly clipped group $\ell_1$-norm as our locality-inducing regularization. We call it a SC-L1 loss for “smoothly clamped $\ell_1$” loss (see the inset).

We adopt a simple implementation for our SC-L1 loss:

$$\|x\|_{SC-L1} = \begin{cases} \|x\|_2 - \frac{s^2}{2s} \|x\|_2 & \text{if } \|x\|_2 < s \\ \|x\|_2 & \text{if } \|x\|_2 \geq s \end{cases}$$

where $s$ is the threshold distance beyond which the regularizer is disabled (this is equivalent to a group-variant of the MCP loss in [Zhang 2010], but we use the term “SC-L1” to emphasize that the minimax concave property is not critical for localized deformation). This function is continuously differentiable and piecewise smooth, and admits a proximal shrinkage operator free of local minima. Near the origin the SC-L1 loss function acts like the group-variant of the MCP loss in [Brown and Narain 2021; Zhang 2010] and the use of the $\ell_1$-norm competes with the elastic energy

<ALGORITHM 1: Three-block ADMM Overview>

Input: A triangle or tetrahedral mesh $V$, $T$
Output: Deformed vertex positions $V$

$V \leftarrow \tilde{V}$

while not converged do
$X_i \leftarrow \text{local step}_1(V, \tilde{V})$ (local step 1)
$Z_i \leftarrow \text{local step}_2(V, \tilde{V})$ (local step 2)
$V \leftarrow \text{global step}_1(X_i, Z_i, \tilde{V})$ (global step)
$U_i \leftarrow \text{local step}_2(X_i, V_i, \tilde{V}_i)$ (local step 2)
$W_i \leftarrow \text{local step}_1(X_i, V_i, \tilde{V}_i)$ (local step 1)
end

In Eq. 2 we can be rewritten as:

$$\begin{align*}
\text{minimize } & \sum_j E(X_j) + \sum_{i \in V} w_i \|V_i - \tilde{V}_i\|_{SC-L1} \\
\text{s.t. } & X_j = \text{sym}(D_j, V), V_j
\end{align*}$$

where $D_j$ is the selection matrix for edges of the $j$-th vertex or element. sym($F$) denotes the symmetric factor $S$ computed using the polar decomposition $F = RS$, where $R$ is the deformation gradient. Thus $X_j$ is the symmetric factor of deformation gradient of the $j$-th vertex or element. (Note that in general sym($F$) $\neq \frac{1}{2}(F + F^T)$.) The goal of using sym($\cdot$) here is to ensure the local coordinates $X_j$ are invariant to rotations as well as translations. For details, please see [Brown and Narain 2021].

3.2 Local Deformation Energy

We denote $V$ as a $|V| \times d$ matrix of vertex positions at the deformed state, and $\tilde{V}$ as a $|V| \times d$ matrix containing rest state vertex positions.

The total energy for our local deformation is as follows:

$$\begin{align*}
\text{minimize } & E(V) + \sum_{i \in V} w_i \|V_i - \tilde{V}_i\|_{SC-L1} \\
\text{s.t. } & V_s = p_s, \quad (\text{position constraint}) \\
& A_k V_l = b_k, \forall t \in S_k \quad (\text{affine constraint})
\end{align*}$$

The first term is an elasticity energy of choice, and can be selected independently of the locality regularization. The second term is the novel “SC-L1 loss” term on the vertex position changes, which measures the locality of the deformation. $a_i$ is the barycentric vertex area of the $i$-th vertex, which ensures the consistency of the result across different mesh resolutions for the same constant $w$. To enable more user control, position constraints and optional affine constraints can be added on selected vertices to achieve different deformation effects. $s$ denotes the indices of the vertices with the position constraint, and we call these vertices “handles”. $S_k$ is the $k$-th set of vertex indices where an affine constraint is added. For simplicity, we omit the position constraints and affine constraints in the discussion below, as they can be easily intergraded to the system by removing the corresponding degrees of freedom and using Lagrange multiplier method (see Eq(1) in [Wang et al. 2015]).

Inspired by the local-global strategy in [Brown and Narain 2021], our local deformation energy (Eq. 2) can be rewritten as:

4 OPTIMIZING WITH ADMM

A natural way to minimize this energy is to use the alternating direction method of multipliers [Boyd et al. 2011] for the sparsity term, and to use a local-global update strategy for the elasticity term. However, previous $\ell_{2,1}$-based methods [Chen et al. 2017] apply these two strategies separately in their local and global steps, resulting in an inefficient optimization scheme. As they only support 2D ARAP energy, we discuss further in Sec. 4.1.

We propose a new way to efficiently minimize such energies in Eq. 3 by combining the sparsity-targeted ADMM with the elasticity-focused local-global strategy. We minimize our energy (Eq. 3) using a three-block alternating direction method of multipliers scheme [Boyd et al. 2011] following the local-global update strategy. Our first local step, finding the optimal symmetric factor $X_j$ of the deformation gradient, can be formulated as a minimization problem on its singular values. Our second local step, minimizing the SC-L1 loss term for each $Z_i$, can be solved using a shrinkage step. Our global step, updating vertex positions $V$, is achieved by solving a linear system. We provide an overview of our three-block ADMM scheme in Alg. 1.
4.1 Example I: Local ARAP Energy

We begin by considering how to maximize an as-rigid-as-possible (ARAP) energy [Sorkine and Alexa 2007] when combined with a SC-L1 loss regularizer. With an ARAP elastic energy, the total energy (Eq. 3) for our local deformation is as follows:

\[
\text{minimize } \sum_{i \in V} \frac{1}{2} \| R_i D_i - \tilde{D}_i \|^2_{W_i} + w_{ai} \| Z_i \|_{\text{SC-L1}},
\]

where \( R_i \) is a \( d \times d \) rotation matrix, \( W_i \) is a \( |N(i)| \times |N(i)| \) diagonal matrix of cotangent weights, \( D_i \) and \( \tilde{D}_i \) are \( 3 \times |N(i)| \) matrices of "spokes and rims" edge vectors of the \( i \)-th vertex at the rest and deformed states respectively. \( \| X \|^2_{W_i} \) denotes \( \text{Tr}(X^T W_i X) \). Here we use \( R_i \) to denote \( X_i \), since we drive the deformation gradient towards a rotation matrix in ARAP energy.

Previous method [Chen et al. 2017] optimizes the \( \ell_{2,1} \) version of Eq. 4 in a less efficient way. Their local step optimizes over per-vertex rotation \( R_i \) and their global step minimizes over vertex positions \( V \) using a two-block ADMM scheme. This leads to an expensive optimization with a full ADMM optimization in each global step, making their method too slow for interactive usage. In contrast, applying our new three-block ADMM scheme to the local ARAP energy results in a much more efficient solver, which is one ADMM optimization itself (see the inset, where the blue regions denote an ADMM optimization). We further show the pseudocode of our three-block ADMM for local ARAP energy in Suppl. Alg. 1.

More concretely, by setting \( Z_i = V_i - \tilde{V}_i \), we can further rewrite Eq. 4 as

\[
\text{minimize } \sum_{i \in V} \frac{1}{2} \| R_i D_i - \tilde{D}_i \|^2_{W_i} + w_{ai} \| Z_i \|_{\text{SC-L1}},
\]

\[
\text{s.t. } Z_i = V_i - \tilde{V}_i, \forall i.
\]

(5a)

(5b)

The above minimization problem can be solved efficiently using the following ADMM update steps:

\[
R_i^{k+1} \leftarrow \arg\min_{R_i \in SO(3)} \frac{1}{2} \| R_i D_i - \tilde{D}_i \|^2_{W_i}
\]

\[
Z_i^{k+1} \leftarrow \arg\min_{Z_i} \| Z_i \|^2_{\text{SC-L1} + \frac{\rho}{2} Z_i \| X_i \|_{\text{SC-L1}} + \frac{\rho}{2} W_i \| V_i \|_{\text{SC-L1}} + \frac{2}{\rho} \| V_i - \tilde{V}_i - Z_i + U_i \|^2_{2}
\]

\[
V_i^{k+1} \leftarrow \arg\min_{V_i} \sum_{j \in T} \frac{1}{2} \| V_i - \tilde{V}_i \|_{\text{SC-L1}} + \frac{\rho}{2} \| V_i - \tilde{V}_i - Z_i + U_i \|^2_{2}
\]

\[
U_i^{k+1} \leftarrow U_i^k + V_i^{k+1} - \tilde{V}_i - Z_i^{k+1}
\]

(6a)

(6b)

(6c)

(6d)

Here \( \rho \) is a fixed penalty parameter. For a detailed derivation of the ADMM update, please see Sec. 2 of the supplementary material.

The various steps in this ADMM-based algorithm are computed as follows:

For updating \( R_i \), local step 1 (Eq. 6a) is an instance of the Orthogonal Procrustes problem, which can be solved in the same way as the rotation fitting step in [Sorkine and Alexa 2007]. The optimal \( R_i \) can be computed as \( R_i^{k+1} = \text{arg min}_{R_i \in SO(3)} \| R_i D_i - \tilde{D}_i \|^2_{W_i} \) from the singular value decomposition of \( M_i = U_i \Sigma_i U_i^T \), where \( M_i = D_i \tilde{D}_i^T \).

For updating \( Z_i \), following the derivation of the proximal operator of SC-L1 loss in Sec. 1 of the supplementary material, our local step 2 (Eq. 6b) is solved using a SC-L1 loss-specific shrinkage step:

\[
Z_i^{k+1} \leftarrow \text{SC-L1} (V_i - \tilde{V}_i + U_i)
\]

\[
S_{\omega_{ai}}(x) = \begin{cases} \left( \frac{\rho - \omega_{ai}}{\rho} \right) x, & \text{if } \| x \| \leq s \\ x, & \text{otherwise} \end{cases}
\]

(7)

(8)

To avoid local minima in the shrinkage step, this assumes \( \rho \) is set to satisfy \( \rho > \frac{\max(\omega_{ai})}{2} \) (see Sec. 1 of the supplementary material).

For updating \( V \), the global step (Eq. 6c) can be achieved by solving a linear system:

\[
(L + \rho I) V = B + \rho (V - Z^k - U^k),
\]

where the Laplacian \( L \) and \( B \) are defined in the same way as the global step (Eq. 9) in [Sorkine and Alexa 2007]. For fixed \( \rho \) an efficient implementation is obtained by precomputing and storing the Cholesky factorization of \( L + \rho I \).

4.2 Example II: Local Neo-Hookean Energy

Our local deformation scheme can be further extended to physics-based elasticity energies, e.g., Neo-Hookean energy. Using the Neo-Hookean energy as our elasticity energy and following the framework of [Brown and Narain 2021], the optimization problem in Eq. 3 can be written as follows:

\[
\text{minimize } \sum_{j \in T} E_{\text{nh}}(X_j) + \sum_{i \in V} w_{ai} \| V_i - \tilde{V}_i \|^2_{\text{SC-L1}},
\]

\[
\text{s.t. } X_j = \text{sym}(D_j V_i), \forall j.
\]

(10a)

(10b)

where \( T \) denotes all the elements.

Similarly, by introducing \( Z_i = V_i - \tilde{V}_i \), we can minimize our local Neo-Hookean energy using ADMM. For a detailed derivation, please see Sec. 3 of the supplementary material.

The ADMM update (Alg. 1) for the above minimization problem is as follows:

\[
X_{j+1} \leftarrow \text{arg min}_{X_j} E_{\text{nh}}(X_j) + \frac{\gamma}{2} \| \text{sym}(D_j V_i) - X_j + W_j \|_2^2
\]

\[
Z_i^{k+1} \leftarrow \text{arg min}_{Z_i} w_{ai} \| Z_i \|^2_{\text{SC-L1}} + \frac{\rho}{2} \| V_i^{k+1} - \tilde{V}_i - Z_i + U_i^k \|^2_{2}
\]

\[
V_i^{k+1} \leftarrow \text{arg min}_{V_i} \sum_{j \in T} \frac{1}{2} \| V_i - \tilde{V}_i \|_{\text{SC-L1}} + \frac{\rho}{2} \| V_i - \tilde{V}_i - Z_i + U_i^k \|^2_{2}
\]

\[
U_i^{k+1} \leftarrow U_i^k + V_i^{k+1} - \tilde{V}_i - Z_i^{k+1}
\]

(11a)

(11b)

(11c)

(11d)

(11e)

Here \( \rho \) and \( \gamma \) are fixed penalty parameters.

The local step 2 (updating \( Z_i \)) and the global step (updating \( V \)) can be solved in the same way as the local ARAP energy (see Sec. 4.1).

For updating \( X_i \), local step 1 (Eq. 11a) can be solved by performing the energy minimization on the singular values of \( \text{sym}(D_j V_i) + W_j \). This is a proximal operator of \( E_{\text{nh}} \) at the rotation-invariant \( \text{sym}(D_j V_i) + W_j \).
Figure 5: Given the same handle offset magnitude, a "natural" ROI size also depends on the way the handles are moved, which is more complex than simply growing the ROI proportional to the handle displacement. Here we moved the handles (yellow) with the same offset magnitude 1.0 towards the bottom, left and right respectively, resulting different ROIs for the same handle offset magnitude.

4.3.1 As-Conformal-As-Possible Energy. For editing tasks where users intend to locally scale the geometry while preserving the texture, it’s desirable to constrain the angle preservation, or conformality (see Fig. 4 and Fig. 10). We can adapt the ARAP energy to as-conformal-as-possible (ACAP) energy [Bouaziz et al. 2012] by allowing local scaling:

\[ E_{\text{ACAP}}(\mathbf{V}) = \sum_{k \in T} \sum_{i,j \in N(k)} \frac{w_{ij}}{2} \|s_k \mathbf{R}_k \mathbf{d}_{ij} - \mathbf{d}_{ij}\|^2 \]  

(16)

where \(s_k\) is a scalar controlling the scaling of the local patch and can be computed analytically (see Sec.4 of the supplementary material).

4.3.2 Cloth. Our method also generalizes to higher co-dimensional settings, such as deformable thin sheets and cloth in \(\mathbb{R}^3\). We model the cloth deformation using ARAP elasticity (Eq. 4), hard strain limiting, and quadratic bending resistance [Bergou et al. 2006].

4.3.3 1D Polyline. Our algorithm can be also extended to the local editing of 1D polyline in vector graphics. The deformation of a polyline can be modeled using the ARAP energy (Eq. 4) with uniform weights.

5 RESULTS

We evaluate our method by comparing it against existing local deformation tools and showcasing its extension to various elastic energies. All the colormaps in our figures visualize the vertex displacement with respect to the rest shape. The accompanying video also includes several animation examples generated using our local deformation tool.

We implement a 2D version of our method in MATLAB with gptoolbox [Jacobson et al. 2018a], and a 3D version in C++ with libigl [Jacobson et al. 2018b] based on the WRAPD framework [Brown and Narain 2021]. We also implement the 2D version of our method in C++ for runtime evaluation and comparison. Benchmarks are performed using a MacBook Pro with an Apple M2 processor and 24GB of RAM for 3D and a Windows desktop with an i9-9900K 3.60 GHz CPU for 2D. Table 2 in the supplementary material shows the performance statistics and relevant parameters of all our examples.

Quality. We compare our methods against other local editing tools, including i) \(L_1\)-based deformation [Chen et al. 2017], ii) regularized Kelvinlets [De Goes and James 2017] and iii) biharmonic...
coordinates [Wang et al. 2015]. Among them, the use of \( \ell_2,1 \)-norm regularization [Chen et al. 2017] causes artifacts (see Fig. 3-b and Fig. 13-b). Regularized Kelvinlets technique [De Goes and James 2017] deforms a shape based on Euclidean distances, thus not shape-aware, creating artifacts when two disjoint parts are close in Euclidean space but far away geodesically (see the blue region in Fig. 3-c and the teeth area in Fig. 13-c). Methods based on biharmonic coordinates, such as [Wang et al. 2015], usually require careful placement of additional fixed control points to pre-determine the ROI. The latter two methods do not minimize any elastic energy in the deformation process, and thus their deformations are more susceptible to shape distortion (see Fig. 13). We additionally compare our method against the sparse deformation method [Gao et al. 2012], which directly introduces a sparsity-induced norm in ARAP energy. As shown in Fig. 2, the resulting deformation is sparse but not local, thus requiring the setup of additional fixed constraints.

In contrast, our method produces the deformation which is local, natural, and shape-aware; it automatically adapts the ROI without the need for careful control primitive setup.

Efficacy. We illustrate the ROI adaptation of our method in different situations: It adapts to different energy models—for example, the local ACAP has a smaller ROI than the local ARAP energy as the former allows for local scaling (see Fig. 4). It also adapts to different extents of deformation. As Fig. 8 demonstrates, the ROI gradually increases as the deformation of the bar becomes larger.

One can configure the local deformation style by choosing various elastic energy models. For example, the local Neo-Hookean energy leads to deformation that preserves volume, while the local ARAP energy is volume agnostic (see Fig. 11). The deformation can be further tuned by introducing additional affine constraints—for instance, to enable the character to wave hands (Fig. 12) or the crocodile to open its mouth (Fig. 13)—in a natural way.

Performance. In terms of performance, our solver is able to efficiently minimize the energy at interactive rate, while the method of [Chen et al. 2017] is too slow to run in realtime. Because their method only supports 2D ARAP energy, in Table 1 of the supplementary material, we evaluate the runtime of our method (using both the SC-L1 loss and \( \ell_2,1 \) loss) and [Chen et al. 2017] on a 2D ARAP local energy and across different mesh resolutions and deformations. Measured with the same convergence threshold, our method runs orders of magnitude faster than [Chen et al. 2017], achieving roughly 1000x speedup for small deformation and 100x speedup for large deformation.

Extensibility. Our method can easily generalize to other dimensions and material models, such as the ACAP deformation, cloth deformation, and 1D polyline deformation (see Fig. 10 and Fig. 4). The local ARAP energy enables local scaling and better preserves the texture around the deformed region. In Fig. 7, the user can interactively edit a polyline and naturally recovers its rest shape, which is a desirable feature by the users. In Fig. 6, to deform a cloth in a physically plausible way, the deformation locality is particularly useful, as otherwise a local edit of the cloth may cause a global change leading to unexpected intersections with other objects. Lastly, to demonstrate our method in a more complex scenario, an editing session involving multiple objects and clothes is shown in Fig. 1.

6 CONCLUSION & FUTURE WORK

We describe a regularization based on an “SC-L1 loss” which provides an effective and simple to implement tool for localizing an elastic energy driven deformation to only those regions of a shape being manipulated by a user. The region of influence induced by our method naturally adapts to the geometry of the shape, the size of the deformation, and the elastic energy being used. Furthermore SC-L1 regularization is generic enough to be applied to a wide range of shapes and elastic energies, including 1D, 2D, 3D and cloth finite element deformation, and is fast enough to be used in real-time. Our proposed approach offers several benefits for shape manipulation: It avoids undesired movement in far-off regions of a shape when only one part is being moved by the user, it allows parts of a shape to be deformed with direct manipulation without a pre-rigging step, and avoids the visual artifacts of previous work.

There remain several issues related to localized shape deformation not addressed by our method. Firstly our regularization is applied independently per-vertex, which makes it difficult to apply to splines, NURBS, or even meshes with highly irregular element sizes, which we mark an important direction for future work. In addition, since we use an ADMM method in the optimization, our approach suffers from the common shortcomings of applying ADMM to non-convex energies, including lack of convergence guarantees and slow convergence when high precision is required. Exploration of other optimization algorithms alleviating these issues is another useful future direction. Finally, although it is out of scope for our work here, we note in particular the usefulness of incorporating localized elastic energy deformation into sculpting workflows for artists. This involves a number of facets: Choosing the correct elastic energy to achieve an artistic effect, providing an intuitive UI to adjust the scale of the ROI (for instance by adjusting \( w \) and \( s \), see Fig. 9 and the supplementary video), and in ensuring that our tool integrates well with other sculpting tools. This is particularly useful when handling large “freeform” deformations, as the elastic energy will tend to fight against such deformations, making other tools more suitable. One simple idea for this is to simply reset the rest shape after each click-and-drag, since each deformation step is then independent of the others, and one could switch between our method and others at each step. We have found this mode of interaction to be useful even when only using our method, as it leads to a simple sculpting-style interface, and we include some examples in the supplementary video.

ACKNOWLEDGMENTS

This work is funded in part by National Science Foundation (1910839). We especially thank George E. Brown for sharing the WRAPD implementation and the help with setting up experiments. We thank Jiayi Eris Zhang and Danny Kaufman for sharing the undeformed scene geometry; Lillie Kittredge and Huy Ha for proofreading; Rundi Wu for the help with rendering; all the artists for sharing the 2D and 3D models and anonymous reviewers for their helpful comments and suggestions.
Figure 6: Our method enables the user to locally edit a simulated cloth in an interactive fashion, without the need of rerunning the simulation. In contrast, quasi-static deformations generated by directly moving the control points [Brown and Narain 2021] have global effects, easily deviating the edit from the initial shape in a distinct way. Bottom row: To show the extent to which the handles affect cloth vertex positions, we colormap the vertex displacement from the input cloth. The handles moved by the user are in yellow. (Undeformed geometry thanks to [Zhang et al. 2022])

Figure 7: Our method also generalizes to 1D polyline editing. Our method enables the deformed shape to naturally return to the rest shape when they are close enough.

Figure 8: Our algorithm naturally enables an adaptive region of influence under different deformations. Only the vertices (yellow) on the rightmost end are selected as handles. From left to right, the offset $\Delta x$ of the rightmost end is 0.0, 0.5, 1.0, 2.0, 4.0, 6.0 and 8.0 respectively.

Figure 9: We show how the ROI (yellow) changes when adjusting $s$ or $w$, while keeping the rest the same. Here the handles are highlighted in red. In general, we use $w$ to control the scale of the ROI, and set $s$ to a small factor of the size of the shape and then leave it alone.
Figure 10: As-conformal-as-possible (ACAP) energy encourages conformality, thus better preserving the color texture by allowing local scaling, while the as-rigid-as-possible (ARAP) energy favors preserving the rigidity. Here the handle (in blue) is placed around the eye and an affine constraint is added to the red region.

Figure 11: ARAP v.s. Neo-Hookean energy with SC-L1 loss. Only the vertices (yellow) on the nose of the pig are selected as handles. Note that Neo-Hookean version exhibits the volume-preservation property.

Figure 12: Our method supports adding both positional constraints (green) and affine constraints (red) on vertices. The region of influence in the deformed shape naturally adapts to the resulting deformation.

Figure 13: Our local deformation (a) is natural and shape-aware and supports an adaptive ROI without the need of additional handles. Here we highlight all the handles that have been moved in yellow, all the additional fixed handles in green and the affine constraint regions in orange. Other local editing alternatives either introduce artifacts (see the flame on the tail in b and d), is unaware of the geometry (see the teeth region in c) or requires additional handles setup (d). More specifically, [Chen et al. 2017] uses a group lasso penalty, which yields hard to avoid artifacts (b). They therefore mitigate it with a smoothing regularizer, at the expense of fidelity in rotations (b1). Removing the smoothing leads to other artifacts and still has global motion (b2). Further increasing the group lasso penalty removes the global motion but amplifies the artifacts (b3).
Supplementary Material for Local Deformation for Interactive Shape Editing

Honglin Chen
Columbia University
New York City, NY, USA
honglin.chen@columbia.edu

Changxi Zheng
Columbia University
New York City, NY, USA
cxz@cs.columbia.edu

Kevin Wampler
Adobe Research
Seattle, WA, USA
kwampler@adobe.com

ACM Reference Format:

1 ADDITIONAL RESULTS

We report the runtime of our method and [Chen et al. 2017] on a 2D ARAP local energy and across different mesh resolutions and deformations in Table 1. We also provide the timings and experiment setup of all our 3D examples in Table 2 and the convergence plot in Fig. 1. In addition to the results in Sec. 5, we include an additional example with one or multiple control points in Fig. 2.

Table 1: Timings comparing our ARAP ADMM formulation in Sec 4.1 with standard local-global ARAP and [Chen et al. 2017] for moving the top-right arm of the octocat in Fig.3 of the main text, meshed at three different resolutions. Wall-clock time is measured by averaging over 100 random deformations and uses the (relatively tight) convergence criteria from [Chen et al. 2017]. All timings are generated by single-threaded C++ implementations, written as equivalently as possible between the methods. For an apples-to-apples comparison these timings include the $E_{smooth}$ energy from [Chen et al. 2017] in the SC-L1 loss and ARAP elastic energies.

<table>
<thead>
<tr>
<th># triangles</th>
<th>ours (w. SC-L1 loss)</th>
<th>ARAP</th>
<th>[Chen et al. 2017]</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>0.00087s</td>
<td>0.00015s</td>
<td>0.11s</td>
</tr>
<tr>
<td>1000</td>
<td>0.0055s</td>
<td>0.0010s</td>
<td>0.59s</td>
</tr>
<tr>
<td>10000</td>
<td>0.058s</td>
<td>0.017s</td>
<td>2.05s</td>
</tr>
<tr>
<td># triangles</td>
<td>large deformation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>0.0053s</td>
<td>0.00032s</td>
<td>0.022s</td>
</tr>
<tr>
<td>1000</td>
<td>0.0078s</td>
<td>0.0018s</td>
<td>0.175</td>
</tr>
<tr>
<td>10000</td>
<td>0.095s</td>
<td>0.027s</td>
<td>2.03s</td>
</tr>
</tbody>
</table>

2 DERIVATION OF PROXIMAL OPERATOR FOR SC-L1 LOSS

For simplicity, we start the derivation with scalar function $f(v) = |v|_SC-L1$. However, the following derivation can be easily extended to vector function $f(x) = \|x\|_SC-L1$ by replacing the $| \cdot |$ with $\| \cdot \|$.

For scalar $v$,

$$f(v) = \begin{cases} |v| - \frac{1}{2\rho} v^2 & |v| < \rho \\frac{1}{\rho} v & |v| \geq \rho \end{cases}$$ (1)

The gradient of $f(v)$ is

$$f'(v) = \begin{cases} \text{sgn}(v) - \frac{1}{\rho} v & |v| < \rho \\frac{1}{\rho} v & |v| \geq \rho \end{cases}$$ (2)

The proximal operator of $f(v)$ can be evaluated as

$$\argmin_v f(v) + \frac{\rho}{2} (v - x)^2$$ (3)

The minimum can be found by setting the derivative to be 0, thus we have

$$0 = f'(v) + \rho (v - x)$$ (4)

$$0 = \rho (v - x) + \begin{cases} \text{sgn}(v) - \frac{1}{\rho} v & |v| < \rho \\frac{1}{\rho} v & |v| \geq \rho \end{cases}$$ (5)

$$0 = \begin{cases} \rho - \frac{1}{\rho} v + \text{sgn}(v) - \rho x & |v| < \rho \\rho (v - x) & |v| \geq \rho \end{cases}$$ (6)

The location of the minimum $v$ depends on whether or not $\rho > \frac{1}{\rho}$.

Case 1: $\rho \leq \frac{1}{\rho}$

When $\rho \leq \frac{1}{\rho}$, then the proximal operator may have up to two local minima—one at the origin and one in the flat region where $|v| \geq \rho$. We arbitrarily prefer the minima at the origin if it exists, giving:

$$S(x) = \begin{cases} 0 & |x| < \rho \\frac{1}{\rho} |x| & |x| \geq \rho \end{cases}$$ (7)

Case 2: $\rho \geq \frac{1}{\rho}$

When $\rho \geq \frac{1}{\rho}$, then there is only one (global) minimum, so the optimal $v$ is:

$$S(x) = \begin{cases} 0 & |x| \leq \frac{1}{\rho} \\frac{\rho x - \text{sgn}(x)}{\rho x - 1} & \frac{1}{\rho} \leq |x| \leq \rho \\frac{1}{\rho} |x| & |x| \geq \rho \end{cases}$$ (8)

for a vector $x$ this can be further simplified to

$$S(x) = \begin{cases} \frac{\rho x - \text{sgn}(x)}{\rho x - 1} x & |x| \leq \frac{1}{\rho} \\frac{\rho x - \text{sgn}(x)}{\rho x - 1} x, \text{ if } |x|_2 \leq s \\text{otherwise} \end{cases}$$ (9)
Table 2: Performance of and parameters for our algorithm on all the 3D examples. All wall-clock timings are reported in seconds, physical parameters are reported with appropriate units. Timing experiments are measured with 8-thread parallelization and PARDISO support [Alappat et al. 2020; Bollhöfer et al. 2019, 2020]. \( |V| \) is the number of vertices, \( |T| \) is the number of elements. \( \rho \) is the ADMM penalty parameter and \( s \) is the SC-L1 loss threshold. Local 1, Local 2 and Global is the total runtime for the local step 1, the local step 2 and the global step respectively. Total is the total runtime for our ADMM algorithm. Residual is the final residuals of our energy, defined as \( |E_{\text{curr}} - E_{\text{prev}}|/(E_{\text{curr}} + 1.0) \), where \( E \) is the total energy and adding 1.0 is to avoid dividing by 0. We use \( w = 1 \times 10^6 \) as the weight of the SC-L1 loss for all the 3D examples. The Young’s modulus is \( 1 \times 10^7 \) Pa and the poisson’s ratio is 0.45 for all our Neo-Hookean examples. We use different \( \rho \) for ARAP and Neo-Hookean energy to handle the different magnitude of the energies due to the physical parameters in Neo-Hookean energy. In general we set \( s \) to be 0.01 times the max extent of the mesh’s bounding box, but other values for \( s \) also work well. For interactive frame rates, we generate all the results using a fixed 500 ADMM iterations or when its objective residual reaches \( 1 \times 10^{-11} \). As shown by Fig. 1, all of examples converged within 500 iterations. For the bar example in Fig.7 and Fig.13 of the main text, we report its average runtime.

| Example  | \( |V| \) | \( |T| \) | Material | \( \rho \) | \( s \) | Residual | Local 1 (s) | Local 2 (s) | Global (s) | Total (s) |
|----------|---------|---------|-----------|---------|-------|---------|-----------|-----------|-----------|-----------|
| Cloth 1(Fig. 6) | 14,641 28,800 | Cloth | 5e3 | 0.4 | 3.93e-5 | 1.58 | 0.04 | 1.76 | 3.38 |
| Cloth 2(Fig. 6) | 14,641 28,800 | Cloth | 5e3 | 0.4 | 2.09e-5 | 1.52 | 0.03 | 1.74 | 3.29 |
| Cloth 1(Suppl Fig.3) | 14,641 28,800 | Cloth | 3e6 | 0.01 | 5.04e-5 | 1.95 | 0.05 | 1.88 | 3.89 |
| Cloth 2(Suppl Fig.3) | 14,641 28,800 | Cloth | 3e6 | 0.01 | 3.54e-7 | 1.60 | 0.06 | 1.71 | 3.37 |
| Cloth(Fig. 1) | 46,977 92,928 | Cloth | 3e6 | 0.01 | 3.26e-4 | 6.06 | 0.10 | 5.41 | 11.57 |
| Pig(Fig. 1) | 10,416 43,749 | NH | 5e5 | 0.01 | 3.58e-3 | 5.85 | 0.03 | 1.61 | 7.49 |
| Pig 1(Fig. 11) | 10,416 43,749 | ARAP | 1e-1 | 0.01 | 3.84e-7 | 2.40 | 0.03 | 1.59 | 4.03 |
| Pig 2(Fig. 11) | 10,416 43,749 | ARAP | 1e-1 | 0.01 | 3.94e-5 | 5.56 | 0.03 | 1.58 | 7.18 |
| Cat(Fig. 1) | 9,078 36,115 | NH | 5e3 | 0.01 | 7.81e-6 | 4.24 | 0.02 | 1.30 | 5.56 |
| Robot(Fig. 1) | 6,601 26,024 | NH | 1e3 | 0.04 | 6.88e-4 | 3.91 | 0.04 | 0.86 | 4.81 |
| Cube(Fig. 1) | 4,913 20,480 | NH | 1e5 | 1.0 | 1.14e-5 | 2.77 | 0.01 | 2.17 | 4.95 |
| Apple(Fig. 1) | 1,088 4,072 | NH | 1e5 | 0.01 | 1.69e-3 | 0.60 | 0.01 | 0.52 | 1.13 |
| Table(Fig. 1) | 6,483 21,864 | ARAP | 2e-4 | 0.01 | 3.41e-11 | 1.46 | 0.04 | 1.03 | 2.53 |
| Bar(Fig. 8) | 7,011 34,557 | ARAP | 1e-1 | 0.1 | 1.28e-4 | 2.89 | 0.08 | 2.69 | 5.65 |
| Bar(Fig. 5) | 7,011 34,557 | NH | 5e3 | 0.1 | 6.66e-5 | 5.74 | 0.03 | 1.63 | 7.41 |

3 DERIVATION OF LOCAL ARAP ENERGY

With as-rigid-as-possible (ARAP) energy [Sorkine and Alexa 2007] as our elastic energy, the total energy for our local deformation is as follows:

minimize \( \sum_{i \in V} \sum_{j \in N(i)} \frac{w_{ij}}{2} \| \| \tilde{d}_{ij} - d_{ij} \|_2^2 + w_{ij} \| \| \tilde{V}_i - V_i \|_2 \|_\text{SC-L1} \) ARAP Localness

(10)

where \( R_i \) is a \( d \times d \) rotation matrix, \( w_{ij} \) is the cotangent weight, \( d_{ij} = [V_j - V_i]^\top \) and \( \tilde{d}_{ij} = [\tilde{V}_j - \tilde{V}_i]^\top \) are the edge vertices between \( i, j \) at the deformed and rest states respectively. \( N(i) \) denotes the neighboring edges of the \( i \)-th vertex in the style of ‘spokes-and-rims’. Here we use \( R_i \) to denote \( X_i \), since we drive the deformation gradient towards a rotation matrix in ARAP energy.

Our local ARAP deformation energy (Eq. 10) can be rewritten as

minimize \( \sum_{i \in V} \frac{1}{2} \| \| R_i D_i - \tilde{D}_i \|_2 \| + w_{ij} \| \| V_i - \tilde{V}_i \|_2 \|_\text{SC-L1} \) \n
(11)

where \( W_i \) is a \( |N(i)| \times |N(i)| \) diagonal matrix of cotangent weights, \( \tilde{D}_i \) and \( D_i \) are \( 3 \times |N(i)| \) matrices of “spokes and rims” edge vectors of the \( i \)-th vertex at the rest and deformed states respectively. \( \| X \|_{W_i}^2 \) denotes \( \text{Tr}(X^\top W_i X) \).

By setting \( Z_i = V_i - \tilde{V}_i \), we can further rewrite Eq. 11 as

\[
\begin{align*}
\text{minimize} & \quad \sum_{i \in V} \frac{1}{2} \| R_i D_i - \tilde{D}_i \|_2^2 + w_{ij} \| V_i - \tilde{V}_i \|_2^2 \\
\text{s.t.} & \quad Z_i = V_i - \tilde{V}_i, \quad \forall i
\end{align*}
\]

(12a)

(12b)

The ADMM update for the above minimization problem is as follows:

\[
\begin{align*}
R_i^{k+1} & \leftarrow \arg \min_{R_i \in SO(3)} \frac{1}{2} \| R_i D_i - \tilde{D}_i \|_2^2 \\
Z_i^{k+1} & \leftarrow \arg \min_{Z_i} w_{ij} \| Z_i \|_2^2 + \frac{\rho}{2} \| V_i^{k+1} - \tilde{V}_i - Z_i + U_i^{k+1} \|_2^2 \\
V_i^{k+1} & \leftarrow \arg \min_{V_i} W_i (V_i^\top L V_i - B_i V_i) + \frac{\rho}{2} \| V_i - \tilde{V}_i - Z_i^{k+1} + U_i^{k+1} \|_2^2 \\
U_i^{k+1} & \leftarrow U_i^k + V_i^{k+1} - \tilde{V}_i - Z_i^{k+1}
\end{align*}
\]

(13a)

(13b)

(13c)

(13d)

Here \( \rho \) is a fixed penalty parameter.

**Updating R**

Local step 1 (Eq. 13a) is an instance of the Orthogonal Procrustes problem, which can be solved in the same way as the rotation fitting
Supplementary Material for Local Deformation for Interactive Shape Editing

Conference’17, July 2017, Washington, DC, USA

ALGORITHM 1: Three-block ADMM for ARAP energy

Input: A triangle or tetrahedral mesh \( \bar{V}, T \)
Output: Deformed vertex positions \( V \)

\[ V \leftarrow \bar{V} \]

while not converged do

\[ R_i \leftarrow \text{local-step}_i(X(V, \bar{V})) \quad \Rightarrow \text{local step 1} \]
\[ Z_i \leftarrow \text{local-step}_i(Z(V, \bar{V})) \quad \Rightarrow \text{local step 2} \]
\[ U_i \leftarrow \text{dual-update}(Z_i, V_i, \bar{V}) \quad \Rightarrow \text{global step} \]

end

step in [Sorkine and Alexa 2007]:

\[
R_{k+1}^i \leftarrow \arg \max_{R_i \in \text{SO}(3)} R_i \cdot M_{ki} \quad (14)
\]

\[
M_i = D_i \tilde{D}_i^T \quad (15)
\]

One can derive the optimal \( R_i \) from the singular value decomposition of \( M_i = U_i \Sigma_i V_i^T \):

\[
R_{k+1}^i \leftarrow V_i U_i^T \quad (16)
\]

Updating \( Z \)

Local step 2 (Eq. 13b) can be solved with a shrinkage step (see derivation in App. 2):

\[
Z_i^{k+1} \leftarrow \text{shrink} \left( S_{\text{shrink}}(x) = \begin{cases} (x - \lambda \text{sign}(x)) / \lambda, & \text{if} \|x\| \leq \lambda \\ x, & \text{otherwise} \end{cases} \right) \quad (17)
\]

where \( \lambda \) is set to satisfy \( \lambda > \frac{\text{max} \|x\|}{2} \) (see App. 2).

Updating \( V \)

The global step (Eq. 13c) can be achieved by solving a linear system:

\[
(L + \rho I) V = B + \rho (\bar{V} + Z^k - U^k), \quad (19)
\]

which can be precomputed for fixed \( \rho \).

4 DERIVATION OF LOCAL NEO-HOOKEAN ENERGY

Our local deformation scheme can be further extended to physics-based elasticity energies, e.g., Neo-Hookean energy. Using the Neo-Hookean energy as our elasticity energy, the optimization problem can be written as follows:

\[
\text{minimize}_{V_i(X_i), \{Z_i\}} \sum_{j \in T} E_{nh}(X_j) + \sum_{i \in V} w_{ai} \|Z_i - \bar{V}_i\|_{\text{SC-L1}}, \quad (20a)
\]

s.t. \( X_j = \text{sym}(D_j V), \forall j \),

where \( T \) denotes all the elements.

By introducing \( Z_i = V_i - \bar{V}_i \), Eq. 20 can be further rewritten into:

\[
\text{minimize}_{V_i(X_i), \{Z_i\}} \sum_{j \in T} E_{nh}(X_j) + \sum_{i \in V} w_{ai} \|Z_i\|_{\text{SC-L1}}, \quad (21a)
\]

s.t. \( X_j = \text{sym}(D_j V), \forall j \), \( Z_i = V_i - \bar{V}_i, \forall i \) \quad (21c)

The augmented Lagrangian for this problem (Eq. 21) is:

\[
L(V, \{X_i\}, \{W_i\}, \{Z_i\}, \{U_i\}) = \sum_{j \in T} E_{nh}(X_j) + \sum_{i \in V} w_{ai} \|Z_i\|_{\text{SC-L1}} + \sum_{i \in V} \frac{\gamma}{2} \|V_i - \bar{V}_i\|_2^2 + \sum_{i \in V} \frac{\rho}{2} \|Z_i - \bar{V}_i\|_2^2 + \sum_{i \in V} \frac{\rho}{2} \|Z_i - \bar{V}_i - \tilde{V}_i\|_2^2 \quad (22)
\]

Therefore, the ADMM update for the above minimization problem is as follows:

\[
X_j^{k+1} \leftarrow \arg \min_{X_j} E_{nh}(X_j) + \frac{\gamma}{2} \|\text{sym}(D_j V) - X_j + W_j\|_2^2 \quad (23a)
\]

\[
Z_i^{k+1} \leftarrow \arg \min_{Z_i} w_{ai} \|Z_i\|_{\text{SC-L1}} + \frac{\rho}{2} \|V_i^{k+1} - \bar{V}_i - Z_i + U_i^k\|_2^2 \quad (23b)
\]

\[
V_i^{k+1} \leftarrow \arg \min_{V_i} \sum_{j \in T} \frac{\gamma}{2} \|\text{sym}(D_j V) - X_j + W_j\|_2^2 + \sum_{i \in V} \frac{\rho}{2} \|V_i - Z_i\|_2^2 + \sum_{i \in V} \frac{\rho}{2} \|Z_i - \bar{V}_i\|_2^2 \quad (23c)
\]

\[
W_j^{k+1} \leftarrow \text{sym}(D_j V) - X_j \quad (23d)
\]

\[
U_i^{k+1} \leftarrow U_i^k + V_i^{k+1} - \bar{V}_i - Z_i^{k+1} \quad (23e)
\]

Here \( \rho \) and \( \gamma \) are fixed penalty parameters.

Compared with the ADMM steps for the local ARAP energy (Eq. 13), the local step 2 (arg min_{Z_i}) and the global step (arg min_{V})
for the local Neo-Hookean energy can be solved in the same way as the local ARAP energy (see App. 3).

Updating X

Local step 1 (Eq. 23a) can be solved by performing the energy minimization on the singular values of sym(DjV) + Wj. This is a proximal operator of E\text{nh} at rotation-invariant sym(DjV) + Wj.

Let us denote the proximal operator of E\text{nh} and the singular value decomposition of sym(DjV) + Wj as:

\[
\text{prox}_{E_{\text{nh}}}(X_j) = E_{\text{nh}}(X_j) + \frac{\gamma}{2} ||\text{sym}(D_jV) + W_j - X_j||_2^2
\]

sym(DjV) + Wj = UjΣjVjT

As shown by [Brown and Narain 2021], we can compute the optimal Xj as:

\[
\Sigma_j^{k+1} \leftarrow \arg \min_{\Sigma_j} \text{prox}_{E_{\text{nh}}}(Uj\Sigma_j^{k+1}V_j^T)
\]

\[
X_j \leftarrow Uj\Sigma_j^{k+1}V_j^T
\]

Specifically, one can compute the SVD of sym(DjV) + Wj and perform the minimization of \text{prox}_{E_{\text{nh}}} only on its singular values, while keeping singular vectors unchanged. The above optimization of the singular values Σj^{k+1} can be performed using an L-BFGS solver.

5 OPTIMAL SCALING s_k FOR EACAP

Adopting the as-conformal-as-possible energy EACAP as our elastic energy requires solving an instance of isotropic orthogonal Procrustes problem in the local step 1 (updating Rk). Following the derivation in [Schönemann and Carroll 1970], one can compute its analytical solution as follows: The optimal rotation R_k can be computed the same way as the local step 1 (updating R_k) for the ARAP energy, and the optimal scaling s_k can be computed analytically as

\[
s_k = \frac{\text{Tr} \left( W_k^T D_k^T W_k D_k \right)}{\text{Tr} \left( W_k^T D_k^T D_k \right)}
\]

When assembling the matrices B for the global step, one needs to replace R_k with s_kR_k.

6 COMPARISON WITH OTHER SC-L1 LOSS ALTERNATIVES

Our SC-L1 loss is designed with the following desirable properties in mind:

(1) Behaves like a \ell_1 loss near zero.

(2) Is constant far from zero.

(3) Admits an efficient proximal shrinkage operator free of local minima.

We will consider each of these properties in turn, examining the motivation and comparing our loss to other alternatives.

(1) Behaves like a \ell_1 loss near zero. This property is what gives our method the ability to localize deformation. The non-differentiable \ell_1-like “vertex” of the SC-L1 loss at the origin is critical in this, and we refer the reader to [Hastie et al. 2015] Sec. 2.2 for more information as to why. Using a loss which is smooth at the origin will invariably result in small amounts of global motion, with the degree of global motion being roughly related to the magnitude of the function’s Hessian at the origin, so the smoother the loss, the more global motion should be expected. Examples of such smooth losses are the smoothed-\ell_1 or Huber loss [Huber 1964] or smoothed-\ell_0 losses [Mohimani et al. 2007; Xiang et al. 2022], and we avoid these losses due to their inability to eliminate global deformations.

(2) Constant far from zero. This property has the effect of disabling the locality-inducing force for parts of a deformed shape which have moved beyond a certain radius, and serves to remove artifacts resulting from parts of a deformed shape being pulled toward a rest position which is now far away. Fig.3(b) and Fig.10(b) in the main text illustrate the types of artifacts which can be observed using a loss without this property.

(3) Efficient proximal shrinkage operator free of local minima. It is convenient for the loss’ proximal shrinkage operator to be uniquely defined. Recalling that this operator is defined as the minimizer(s) of Eq. 3, the condition that it be uniquely defined is equivalent to requiring that Eq. 3 has a unique minimum. A sufficient condition for this is that \( f'(v) + \frac{\rho}{2} (v - x)^2 \) is strictly convex, or equivalently that \( \rho > -\frac{d^2}{dv^2} f(v) \). We thus prefer a loss function which has a bounded negative second derivative, and the lower this bound the lower the value of \( \rho \) which can be used to convexify it. Within distance \( s \) from the origin (excluding the origin itself), our loss has a constant negative value for \( \frac{d^2}{dv^2} f(v) \), making it in some sense the minimal function satisfying the three desired properties. Other losses are of course possible, but would require a higher value of \( \rho \) for the same radius of effect.

Alternative losses. A natural question is whether there might be other locality-inducing losses which could be used in place of our SC-L1 loss, perhaps even a generic spline-based loss which could perform the minimization of prox_{E_{\text{nh}}} only on its singular values, while keeping singular vectors unchanged. The above optimization of the singular values Σj^{k+1} can be performed using an L-BFGS solver.

REFERENCES


Jianhong Xiang, Hao Xiang, Linyu Wang, and Yu Zhong. 2022. Medical Image Reconstruction Method Based on Smooth L0 Norm. https://doi.org/10.21203/rs.3.rs-1535219/v1