

Simple Approximations of Planar Deformation Operators

Roi Poranne¹ Yaron Lipman²
¹ETH Zurich ²Weizmann Institute

Abstract

We consider the problem of shape deformation in two dimensions, using point handles. The challenge these types of deformation algorithms face is, given the user handles' positions, to quickly infer the deformation of the rest of the shape. Current algorithms can be split into two groups: variational algorithms that minimize or approximate the minimizer of some energy every time new handles' positions are provided (*e.g.* As-Rigid-As-Possible), and closed-form expressions that relate the handles' positions and the deformation using known weights or formulas (*e.g.* Linear Blend Skinning or Generalized Barycentric Coordinates). Variational algorithms are time consuming, but usually produce high quality deformations while closed-form expressions are very fast to evaluate, but sometimes fail to produce satisfying deformations.

In this paper we bridge this gap for planar deformations by designing closed-form expressions that well approximate the deformations computed by variational algorithms and consequently achieves both real-time performance and high-quality deformations.

1 Introduction

Deformations are central in computer graphics and image processing. Their applications are wide-ranging, including image warping, registration and character animation.

Our main focus in this paper are deformations controlled by *point handles*. In this case, the user drives the deformation of a planar shape $\Omega \subset \mathbb{R}^2$ by displacing a set of points located on the shape $\mathbf{P} = (\mathbf{p}_1, \dots, \mathbf{p}_n) \subset \Omega^n$ to new positions $\mathbf{Q} = (\mathbf{q}_1, \dots, \mathbf{q}_n) \in \mathbb{R}^{2 \times n}$ and the algorithm computes a map deforming the rest of the shape $\mathbf{f}(\mathbf{x}) : \Omega \rightarrow \mathbb{R}^2$.

State-of-the-art deformations are usually achieved using a variational deformation framework. In this framework, the deformation \mathbf{f} is computed via a minimization of an energy functional. This is done by first specifying a set of spatial basis functions $\Phi = \{\varphi_1(\mathbf{x}), \dots, \varphi_{|\Phi|}(\mathbf{x})\}$ (*e.g.*, piecewise-linear basis functions over a triangulation of Ω or some reduced basis), and expressing the family of candidate deformations in this basis,

$$\mathbf{f}_\Phi(\mathbf{x}) := \sum_{i=1}^{|\Phi|} \mathbf{C}_i \varphi_i(\mathbf{x}), \quad (1)$$

where $\mathbf{C}_i = [C_i^x, C_i^y]^T \in \mathbb{R}^{2 \times 1}$. Then, the unknowns \mathbf{C}_i are found by minimizing a *deformation energy*. This usually leads to a non-linear optimization problem that is computationally demanding, and special efforts were dedicated in the past to design efficient solvers and acceleration techniques [Sheffer and Kraevoy 2004; Botsch et al. 2006; McAdams et al. 2011; Hildebrandt et al. 2011; Manson and Schaefer 2011].

Conceptually, the deformations resulting from the variational framework can be described for *all* possible handle positions *a-priori*. This can be done by regarding the vector coefficients \mathbf{C}_i as functions of the handles' positions, namely $\mathbf{C}_i(\mathbf{Q})$. In this sense, the handles' positions \mathbf{Q} can be regarded as part of the deformation problem and

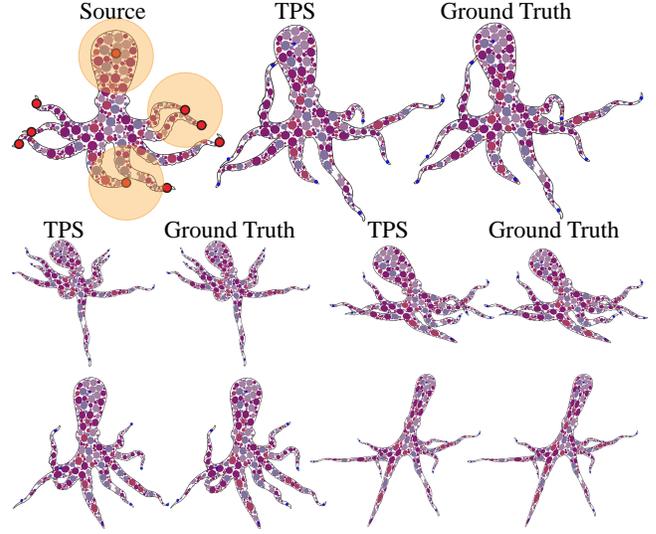


Figure 1: Deformation of the Octo model, using an approximate deformation operator. Once the approximate DefOp has been trained, it can be evaluated in a fraction of the time it takes to compute the variational DefOp. In this example, we used 1000 samples and RBFs. The sample were taken with equal probability from discs around the samples (shown in orange), and the samples' voronoi cells.

it becomes useful to encode the relations between handles' positions \mathbf{Q} and deformation of points $\mathbf{x} \in \Omega$ in a functional form. This leads to the *deformation operator*, or DefOp for short, denoted by

$$\mathcal{D}(\mathbf{x}, \mathbf{Q}) : \mathbb{R}^2 \times \mathbb{R}^{2 \times n} \rightarrow \mathbb{R}^2. \quad (2)$$

Analogously to the treatment of spatial dimensions in variational deformations, we can choose a *handle basis* $\Psi = \{\psi_1(\mathbf{Q}), \dots, \psi_{|\Psi|}(\mathbf{Q})\}$ for the handles space $\mathbb{R}^{2 \times n}$ and approximate any DefOp with the finite dimensional, (Φ, Ψ) -DefOp:

$$\mathcal{D}_{\Phi, \Psi}(\mathbf{x}, \mathbf{Q}) := \sum_{i=1}^{|\Phi|} \sum_{j=1}^{|\Psi|} \mathbf{C}_{ij} \varphi_i(\mathbf{x}) \psi_j(\mathbf{Q}), \quad (3)$$

where $\mathbf{C}_{ij} = [C_{ij}^x, C_{ij}^y]^T \in \mathbb{R}^{2 \times 1}$ are the unknown variables of the approximation. The main advantage of this description is that once the coefficients \mathbf{C}_{ij} are fixed, computing the deformation for any $\mathbf{x} \in \Omega$ and every \mathbf{Q} amounts to the simple closed-form expression in Eq. (3), where no optimization is required.

Somewhat surprisingly, formula (3) can be used to approximate variational deformation models such as as-rigid-as-possible (ARAP) [Sorkine and Alexa 2007; Chao et al. 2010]. Constructing such an approximation $\mathcal{D}_{\Phi, \Psi}$ is rather simple and follows the following steps: First, choose any spatial basis Φ , and let handle basis Ψ to be a generic approximation function bases (*e.g.*, polynomials and radial-basis functions). Second, sample the desired deformation operator at a set of sample handle points $\{\mathbf{Q}_k\}$. Lastly, fit the model

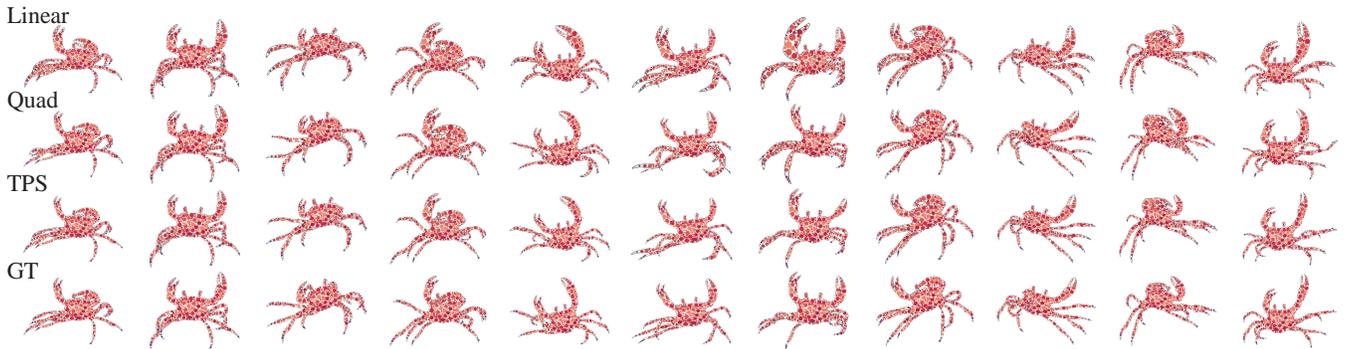


Figure 2: Approximate ARAP deformations of the Crab model, using linear, quadratic and TPS basis functions, compared to the ground truth (bottom). This model has 11 handles, and was trained using 500 samples. We picked several handle positions at random. Note how generally the approximation improves when the handle basis becomes richer.

(3) using a least-squares error functional to the DefOp at the sample points.

The sampling of the DefOp at the handle points \mathbf{Q}_k , that is, computing the deformations for a set of handle positions \mathbf{Q}_k , is the most computationally demanding part of our algorithm. Nevertheless, it can be done efficiently in our case using the following bootstrapping procedure: We first use a linear basis Ψ on a small sampling, to train a simple DefOp, providing a rough approximation to the variational one. Next, we use this approximation to initialize the optimization for a desired number of sample points. Finally, we use the new sample points to train the final approximation $\mathcal{D}_{\Phi, \Psi}$.

We demonstrate that DefOp approximations of ARAP can provide comparable deformations using a simple closed-form expression (3) at reasonable preprocessing times. For example, Figure 1 shows an octopus model with 9 handle points. The deformation operator in this case was trained on 1000 samples in 20 minutes. We used at most 1019 handle basis functions, and achieved deformations computable at rate of more than 1,800 fps.

2 Background and previous work

Linear handle basis. Possibly the simplest example of a (Φ, Ψ) -DefOp used in practice is,

$$\mathcal{D}_w(\mathbf{x}, \mathbf{Q}) = \sum_i w_i(\mathbf{x}) \mathbf{q}_i, \quad (4)$$

where $w_i(\mathbf{x})$ are scalar weight functions defined over the shape Ω . Previous methods that used this model differ mostly in how the weight functions are defined. One early case is [Sederberg and Parry 1986] which uses tensor product B-splines. Currently, this model is perhaps most associated with Generalized Barycentric Coordinates (GBC) [Floater 2003; Ju et al. 2005; Hormann and Floater 2006; Joshi et al. 2007] where \mathbf{q}_i are vertices of an enclosing polygon (“cage”) used to control the deformation. It is important to note again that the weight functions in (4) are predefined and that the dependence of \mathcal{D}_w in \mathbf{q}_i is linear. This is in contrast to methods that use basis functions to *interpolate* the handles’ positions by solving a linear system, such as [Bookstein 1989; Botsch and Kobbelt 2005].

Eq. (4) can be seen as a (Φ, Ψ) -DefOp by considering $\varphi_i(\mathbf{x}) = w_i(\mathbf{x})$, and $\psi_j(\mathbf{Q})$ to consist of the linear coordinate functions q_j^x, q_j^y . This specific model is quite limited, since the different coordinate functions do not blend (*i.e.*, x coordinate of \mathbf{q}_i only affects the x coordinate of the map) and each \mathbf{q}_i only multiplies a single matching w_i . While the weight functions have been improved in terms of, *e.g.*, smoothness [Joshi et al. 2007] and locality [Zhang et al. 2014],

the above limitation hinders GBC from handling rotations well in deformations, and GBC has been noted in the past to cause excessive shearing. A partial remedy was suggested in [Weber et al. 2011] where the coordinate functions of \mathbf{q}_i do blend in a certain way due to the representation of \mathbf{q}_i as complex numbers, and w_i as complex valued functions. This model can be formulated using matrix-valued functions as follows,

$$\mathcal{D}_W(\mathbf{x}, \mathbf{Q}) = \sum W_i(\mathbf{x}) \mathbf{q}_i, \quad (5)$$

where $W_i(\mathbf{x})$ are 2×2 similarity matrices. These matrix weights can generally handle rotations better, and in 2D can even generate conformal maps [Lipman et al. 2008; Weber et al. 2009], though interpolation capabilities are then lost. Nevertheless, this model is still not the most general linear model since the matrices $W_i(\mathbf{x})$ can be taken to include non-similarities, and coupling of different \mathbf{q}_i can be introduced.

This hints that there is something to be gained by extending the handle basis to the full linear basis in the handle space $\mathbb{R}^{2 \times n}$, and probably even beyond linear. This is exactly what we pursue in this paper.

Variational deformation. Currently, variational deformation techniques produce the highest quality deformations with the drawback of being computationally intensive. In variational techniques, an optimization problem is solved repeatedly during user interaction, for every set of new handle positions \mathbf{Q} specified by the user. The fundamental model for these problems is,

$$\begin{aligned} \min_{\{\mathbf{C}_i\}} \quad & \mathbf{E}(\mathbf{f}_{\Phi}) \\ \text{s.t.} \quad & \mathbf{f}_{\Phi}(\mathbf{p}_{\ell}) = \mathbf{q}_{\ell}, \quad \ell = 1, \dots, n \\ & \mathbf{f}_{\Phi}(\cdot) = \sum \mathbf{C}_i \varphi_i(\cdot), \end{aligned} \quad (6)$$

where \mathbf{f}_{Φ} is a representation of a map in some spatial basis Φ . Each such problem defines a *variational* DefOp $\mathcal{D}_{\Phi}^{\mathbf{E}}$, which, given \mathbf{Q} , can be solved to obtain $\mathcal{D}_{\Phi}^{\mathbf{E}}(\mathbf{x}, \mathbf{Q})$ as a function of \mathbf{x} . Different methods vary mostly in terms of the deformation energy \mathbf{E} they use, the minimization algorithm used to solve (6), and the discretization of the problem, namely the choice of the basis Φ .

The vast majority of variational methods use linear finite elements as the basis for the optimization. Early methods minimized some form of quadratic smoothness energy, *e.g.*, Dirichlet, Poisson or biharmonic energy [Botsch and Kobbelt 2004; Yu et al. 2004; Sorkine et al. 2004], or conformal energy [Lévy et al. 2002; Liu et al. 2008; Igarashi et al. 2005], by solving a linear system. These methods perform well for small rotations, but suffer from lack of rotation

invariance and are sub-optimal in handling larger rotations. Linear rotation-invariant methods do not tackle the point handle case [Lipman et al. 2005; Kircher and Garland 2008]. Non-linear variational methods have formulated rotational invariant energies [Sheffer and Kraevoy 2004] but the focus then moved to designing an efficient optimization which turned out to be a challenge [Botsch et al. 2006; Huang et al. 2006; Sorkine and Alexa 2007; Chao et al. 2010].

Skeletal Deformation. A different deformation technique other than the handle-based model is based on skeletons. While we do not treat this kind of deformation in this paper, some of the previous work is still relevant. We give a brief review here, and refer the reader to [Jacobson et al. 2014] for a more comprehensive introduction.

Skeletal deformation is usually associated with Linear Blend Skinning, which has the following form

$$\mathcal{D}_{LBS}(\mathbf{x}) = \sum w_i(\mathbf{x})(T_i\mathbf{x} + \mathbf{q}_i), \quad (7)$$

where each T_i is a transformation corresponding to a certain bone in the skeleton. [Jacobson et al. 2014] noted that Eq. (7), similarly to (4), is not the most general linear model. [Wang and Phillips 2002; Merry et al. 2006] have noted this as well, and have suggested more general models, but still not the most general model possible. Several authors have suggested different extensions to the linear basis [Jacobson and Sorkine 2011; Kavan and Sorkine 2012]. Example-based skinning assumes that several examples of deformed poses are given and solves an interpolation problem [Sloan et al. 2001; Mohr and Gleicher 2003; Lewis et al. 2000; Weber et al. 2007].

Our approach is related, however differ in two key aspects: First, our deformation operators model the relation between position and handles in the arguably most general way allowing non-linear relations approximated with polynomials and radial-basis functions. Second, we approximate the deformation operator for point-based deformations where no input rotations are provided. This is in contrast to skinning where transformations of bones are known (*i.e.*, T_i in Eq. (7)). We show that even in this more challenging settings, simple closed-form formula such as (3) can handle rotations gracefully.

3 Method

In this section we present our approach for computing a finite dimensional (Φ, Ψ) -DefOp $\mathcal{D}_{\Phi, \Psi}$, which approximates a given variational DefOp, \mathcal{D}_{Φ}^E .

First, we discuss the choice of approximation bases Φ, Ψ and prove an interpolation result for DefOps. Second, we describe the least-squares optimization to calculate the (Φ, Ψ) -DefOp approximation by fitting at a set of handle point samples $\{\mathbf{Q}_k\}$. Lastly, we discuss how to calculate the deformations at the samples \mathbf{Q}_k efficiently with a bootstrapping procedure taking advantage of lower degree DefOp approximation.

3.1 Choice of bases Φ, Ψ

In general, our method can work with any choice of the spatial basis Φ and the handle basis Ψ . However, in order to produce useful approximations to DefOps, the bases should possess some approximation power. Therefore, we first require Φ and Ψ to contain the linear functions: for Φ , it means just the constant function together with the x and y coordinate functions in \mathbb{R}^2 , and for Ψ , the constant functions and all the coordinate functions in $\mathbb{R}^{2 \times n}$, in total $2n + 1$ functions.

To understand how to extend Φ, Ψ over the linear bases, we look again at Eq. (3). Fixing \mathbf{Q} and observing $\mathcal{D}_{\Phi, \Psi}(\mathbf{x}, \mathbf{Q})$ as a function

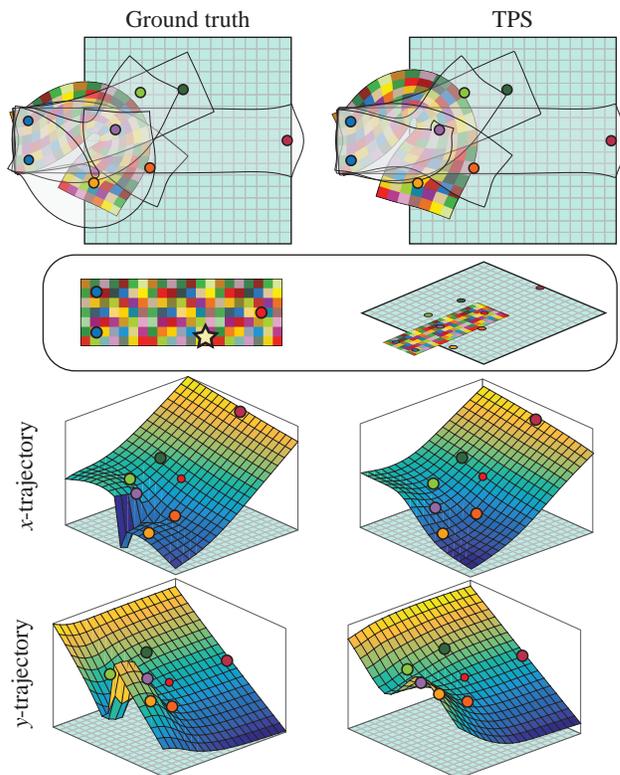


Figure 3: An example showing trajectories and their approximations. The trajectory of a point is its displacement as a function of the displacement of the handles. In this example, only one handle moves, which makes it possible to visualize trajectories. We show the trajectory of the starred point (second row, left). The trajectory is visualized by the x - and y -components of the displacement of the point. In the top row, we show several variational deformations of the bar, with their TPS approximations (see Fig. 7 for more details on this example). The two bottom rows show the trajectories of the true variational DefOp and the approximation. The colored dots correspond to the deformations from the top row (their positions are approximate). Note how the discontinuity in the ground truth trajectory is smoothed out by the approximation, but the overall structure is captured in the approximation.

of \mathbf{x} , we see that Φ should be rich enough to approximate the deformation \mathbf{f}_{Φ} as produced by the variational approach (6). For this reason, we select to include in Φ some of known bases from variational deformation literature: The standard linear finite-elements (FEM) hat functions defined over a mesh triangulating the domain Ω , or, a reduced basis defined over it, e.g. the eigenfunctions of the Laplacian define over the mesh. For the latter we use the linearly precise Laplacian presented in [Wang et al. 2015] that better handles areas near boundaries.

To understand how to set Ψ , we fix a particular point \mathbf{x} and observe how it moves, that is, we look at $\mathcal{D}_{\Phi, \Psi}(\mathbf{x}, \mathbf{Q})$, as a function of \mathbf{Q} . We call this function of the positions as a function of \mathbf{Q} the *trajectory* of \mathbf{x} . Examining these functions for the ARAP energy for a bar shows that they are mostly very smooth. In fact, they are dominated by a linear component, except in area where the deformation is not stable, as can be seen in the left column of Figure 3. This motivates picking generic approximation bases, such as the thin-plate splines (TPS) [Bookstein 1989], a type of Radial Basis Function (RBF), for approximating the trajectories, namely for choosing the handle basis

Ψ . In our approach, we put one RBF centered at each sample \mathbf{Q}_k , $k = 1, \dots, m$. The right column in Figure 3 shows the trajectories of a (Φ, Ψ) -DefOp $\mathcal{D}_{\Phi, \Psi}$ with TPS basis for Ψ as computed using the algorithm presented below. Note that the trajectories are overall well approximated.

Aside from TPS, we also test polynomials of degree up to 3 for Ψ . It is important to note that the number of functions in the Ψ basis rises quickly, due to the high dimension of the handle space ($2n$): there are $1 + 3n + 2n^2$ basis functions in the quadratic polynomials, $1 + \frac{11}{3}n + 4n^2 + \frac{4}{3}n^3$ in cubic polynomials, and $1 + 2n + m$ in the TPS basis. As we show below, the TPS basis provides the best tradeoff of approximation quality and compactness of the basis. An interesting point is that choosing the TPS basis for Ψ allows interpolating arbitrary deformations \mathbf{f}_{Φ} at each and every handle position \mathbf{Q}_k , as the following proposition implies,

Proposition 1. *Let Ψ be the TPS handle basis, and let an arbitrary spatial deformation*

$$\mathbf{f}_{\Phi}^k(\mathbf{x}) = \sum_i \mathbf{C}_i^k \varphi_i(\mathbf{x})$$

be prescribed for every handle sample \mathbf{Q}_k , $k = 1, \dots, m$. Then, there exists an interpolatory (Φ, Ψ) -DefOp, that is

$$\mathcal{D}_{\Phi, \Psi}(\cdot, \mathbf{Q}_k) \equiv \mathbf{f}_{\Phi}^k(\cdot), \quad \forall k = 1, \dots, m$$

Proof. Using the definition of $\mathcal{D}_{\Phi, \Psi}$, the proposition holds if the following equality holds for all \mathbf{x} , and $k = 1, \dots, m$,

$$\sum_i \left(\sum_j \mathbf{C}_{ij} \psi_j(\mathbf{Q}_k) \right) \varphi_i(\mathbf{x}) = \sum_i \mathbf{C}_i^k \varphi_i(\mathbf{x}).$$

Since Φ is a basis this will be true iff

$$\sum_j \mathbf{C}_{ij} \psi_j(\mathbf{Q}_k) = \mathbf{C}_i^k,$$

for all \mathbf{x} , $i = 1, \dots, n$, $k = 1, \dots, m$. And these equalities for each fixed i can be seen as interpolation problem in the TPS basis Ψ . \square

We show examples comparing these bases in approximating a variational deformation operator (denoted as "ground truth") in Fig. 4, and Fig. 7. We note that, as the basis becomes richer with basis functions, the deformation becomes more similar to the ground truth. We investigate the effect of the choice of basis and number of samples on the deformation operator $\mathcal{D}_{\Phi, \Psi}$ in Section 4.

3.2 Least-squares approximation

In this section we present a least-squares approximation of a given ground truth deformation operator \mathcal{D}_{Φ}^E with a (Φ, Ψ) -DefOp $\mathcal{D}_{\Phi, \Psi}$. The idea is to define $\mathcal{D}_{\Phi, \Psi}$ as best L_2 approximation to the ground-truth deformation operator \mathcal{D}_{Φ}^E . Specifically, we consider the (Φ, Ψ) -DefOp that solves the linearly constrained least-squares problem,

$$\min_{\{\mathbf{C}_{ij}\}} \sum_{k=1}^m \left\| \mathcal{D}_{\Phi}^E(\cdot, \mathbf{Q}_k) - \mathcal{D}_{\Phi, \Psi}(\cdot, \mathbf{Q}_k) \right\|^2 \quad (8a)$$

$$\text{s.t. } \mathcal{D}_{\Phi, \Psi}(\mathbf{x}, \mathbf{P}) = \mathbf{x}, \quad \forall \mathbf{x} \in \Omega \quad (8b)$$

$$\mathcal{D}_{\Phi, \Psi}(\mathbf{p}_{\ell}, \mathbf{Q}_k) = \mathbf{q}_{k, \ell}, \quad \forall \ell, k \quad (8c)$$

The energy (8a) simply strives to fit the (Φ, Ψ) -DefOp to the ground truth at a set of sample handle points $\mathbf{Q}_k = (\mathbf{q}_{k, 1}, \dots, \mathbf{q}_{k, n}) \in$

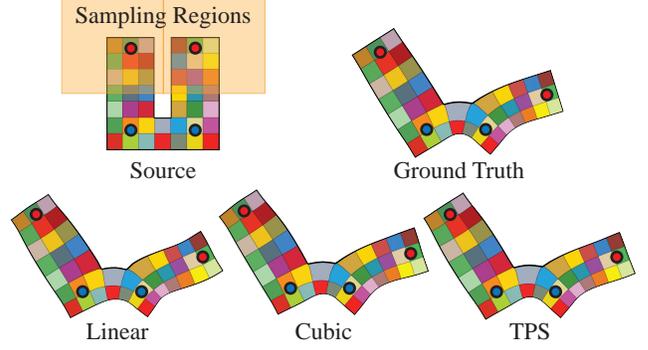


Figure 4: *Deformation of a U shape, with different handle bases. The orange region was sampled uniformly with 50 points for the two red handles, while the blue handles remained fixed. Note that in this case, the linear basis already produces a result very similar to the ground truth, while the cubic and TPS bases improve upon that result.*

$\mathbb{R}^{2 \times n}$, $k = 1, \dots, m$. The constraint (8b) requires that the rest-pose is recovered if the handles are placed in their original positions, *i.e.*, $\mathbf{Q} = \mathbf{P}$. The requirement that (8b) holds for all $\mathbf{x} \in \Omega$ can be enforced by writing the functions $\mathbf{x} = [x, y]^t$ in the basis Φ and equating coefficients on both sides, which leads to a set of linear equations.

The constraint (8c) asks that the handle position are interpolated at the sample handle points. This raises the question, whether interpolation happens at other, arbitrary handle positions $\mathbf{Q} = (\mathbf{q}_1, \dots, \mathbf{q}_n)$. Namely, is it true that $\mathcal{D}_{\Phi, \Psi}(\mathbf{q}_{\ell}, \mathbf{Q}) = \mathbf{q}_{\ell}$? Indeed, interpolation is guaranteed at all handle positions, as shown in the proposition below.

Proposition 2. *If Ψ contains the linear functions and Φ can interpolate arbitrary values at \mathbf{P} , then (8c) is feasible and implies interpolation at all handle positions $\mathbf{Q} \in \mathbb{R}^{2 \times n}$.*

Proof. Plugging the definition of $\mathcal{D}_{\Phi, \Psi}$ into (8c) leads to

$$\sum_j \left(\sum_i \mathbf{C}_{ij} \varphi_i(\mathbf{p}_{\ell}) \right) \psi_j(\mathbf{Q}_k) = \mathbf{q}_{k, \ell}, \quad (9)$$

for all $\ell = 1, \dots, n$, $k = 1, \dots, m$. Since Ψ contains the linear functions in $\mathbb{R}^{2 \times n}$ we can choose $\mathbf{C}_j^{\ell} \in \mathbb{R}^{2 \times 1}$ so that

$$\sum_j \mathbf{C}_j^{\ell} \psi_j(\mathbf{Q}) = \mathbf{q}_{\ell}, \quad \forall \mathbf{Q} \in \mathbb{R}^{2 \times n}, \ell = 1, \dots, n.$$

Plugging this with $\mathbf{Q} = \mathbf{Q}_k$ into (9) we get

$$\sum_j \left(\sum_i \mathbf{C}_{ij} \varphi_i(\mathbf{p}_{\ell}) \right) \psi_j(\mathbf{Q}_k) = \sum_j \mathbf{C}_j^{\ell} \psi_j(\mathbf{Q}_k),$$

for all $\ell = 1, \dots, n$, $k = 1, \dots, m$. For every fixed ℓ , since Ψ provides a unique interpolation on a set of m generic points \mathbf{Q}_k (or less than m , *e.g.*, for polynomial bases), we have equality in the above equation for all $\mathbf{Q} \in \mathbb{R}^{2 \times n}$. To show feasibility, we note that for every fixed j , the following interpolation problem is solvable using the interpolation property of Φ ,

$$\sum_i \mathbf{C}_{ij} \varphi_i(\mathbf{p}_{\ell}) = \mathbf{C}_j^{\ell}, \quad \forall \ell = 1, \dots, n.$$

\square

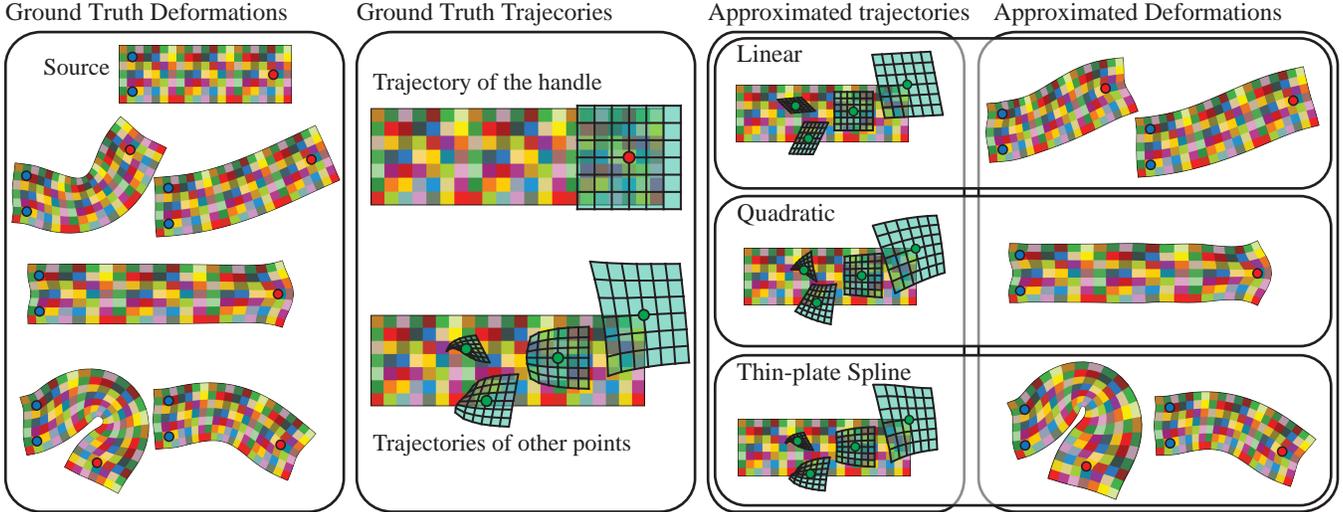


Figure 5: Approximation of deformation trajectories: (Left) We show a collection of deformations obtained by positioning the red handle and solving a S-ARAP deformation problem. (Middle) We observe that trajectories of points, i.e. their displacement as a function of the displacement of the handle, are generally smooth and slowly varying. (Right) Therefore, this dependence can be well-approximated using classical approximation bases, such as polynomials or thin plate splines, resulting in closed-form expressions for computing deformations comparable to the ground truth.

Expressing (8) directly in terms of the unknowns of the model $\mathbf{C}_{ij} = [C_{ij}^x, C_{ij}^y]^T$, where we denote the matrices $C^\nu = (C_{ij}^\nu)$, $\nu \in \{x, y\}$ gives the a linearly constrained least-squares problem solved using Lagrange multipliers (derivation is provided in Appendix A).

3.3 Bootstrapping deformation sampling

In order to solve (8), one needs to supply a set of deformations $\mathcal{D}_\Phi^E(\cdot, \mathbf{Q}_k)$ at a set of samples points $\{\mathbf{Q}_k\}_{k=1}^m \subset \mathbb{R}^{2 \times m}$. Given a set of samples one could optimize (6) for every sample \mathbf{Q}_k independently. However, the process may enjoy a considerable speedup by using an early DefOp approximation for bootstrapping, as follows. We choose Ψ to contain only the constant and linear functions and sample $2n + 1$ deformations. Then, we fit a linear deformation operator (linear here refers to the Ψ basis) by solving (8). This already provides a plausible approximation to the ground truth deformation operator. Second, we use this rough DefOp to initialize the optimization (6) for a larger set of samples $\{\mathbf{Q}_k\}$. Then fit again (8), now with the bigger basis Ψ , to get a better approximation. This procedure can be repeated until sufficiently large sample set $\{\mathbf{Q}_k\}$ has been trained with by a deformation operator $\mathcal{D}_{\Phi, \Psi}$.

Since the initial guesses provided by these rough approximations are already close to the ground truth, only a few iterations are typically required on average to converge. This is in contrast to the hundreds and thousand required if we were to start the optimization from the rest pose.

Defining the samples $\{\mathbf{Q}_k\}$. A reasonable choice of samples $\mathbf{Q}_k = (\mathbf{q}_{k,1}, \dots, \mathbf{q}_{k,n}) \in \mathbb{R}^{2 \times n}$, $k = 1, \dots, m$ is to choose, $\mathbf{q}_{k,\ell}$ uniformly at random in disc of some radius centered at the handle \mathbf{p}_ℓ (see inset). Other options we explored are uniform circles centered at \mathbf{p}_ℓ , Gaussian centered at \mathbf{p}_ℓ , and the Voronoi cells of \mathbf{p}_ℓ w.r.t. \mathbf{P} , or a combination of these.

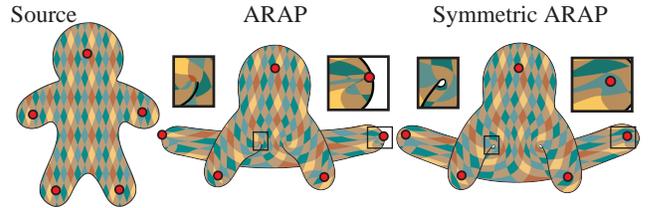


Figure 6: variational deformations using different energies. Note how ARAP generates cusps and foldovers and isometric distortion introduces scaling, while, in contrast, the symmetric ARAP deformation behaves well.

4 Evaluation

This section summarizes the evaluation and experiments performed using the algorithm for approximating deformation operators.

Deformation energies. The choice of energy \mathbf{E} is instrumental to achieving a useful deformation operator. One of the most popular deformation energies is the As-Rigid-As-Possible (ARAP) energy. The ARAP energy of a deformation \mathbf{f}_Φ (see Eq. (1)) for any piecewise-linear basis Φ over a triangulation is defined as

$$\mathbf{E}(\mathbf{f}_\Phi(\cdot)) = \sum_{t \in F} [(\Sigma_t - 1)^2 + (\sigma_t - 1)^2] A_t, \quad (10)$$

where Σ_t, σ_t are the singular values of the constant differential of \mathbf{f}_Φ restricted to triangle t of area A_t .

The ARAP energy has a couple of well-known drawbacks: It tends to create cusps at the handles, and the deformation can have foldovers. Therefore, we chose to test our framework mostly with a novel variant of the ARAP energy, which we call *Symmetric ARAP* (S-ARAP). This energy is defined by replacing the distortion term in (10) with

$$(\Sigma_t - 1)^2 + (\sigma_t^{-1} - 1)^2. \quad (11)$$

The S-ARAP energy has the two key properties: Firstly, it treats shrinkage and expansion symmetrically and as a consequence also

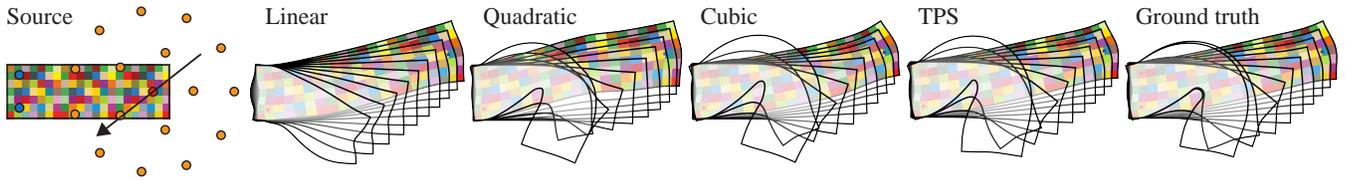


Figure 7: Bar deformations using polynomials and TPS basis functions. Clearly, using linear basis functions is not enough to produce high quality results. Switching to quadratic functions already shows great improvement, while enriching the basis with cubic or using TPS further enhances the quality of the deformation.

prevents triangle flip (the energy explodes at degenerate triangle configuration). Secondly, it maintains the elastic look of ARAP. Fig. 6 shows a comparison between ARAP and S-ARAP when used to deform a gingerbread man model. Note how the traditional ARAP energy causes cusps and foldovers in the deformation, and the isometric distortion energy scales the shape, while the S-ARAP energy behaves well.

Trajectories. Trajectories were mentioned in 3.1. A trajectory is a function encoding the movement of a fixed spatial point \mathbf{x} as a function of the handles \mathbf{Q} . Figure 5 shows another visualization of trajectories of the ground truth S-ARAP deformation operator compared to its approximation with linear, quadratic and TPS (Φ, Ψ)-DefOp. Note that the linear (Φ, Ψ)-DefOp can only represent linear trajectories, while quadratic and TPS can represent more general trajectories that allows handling rotations better.

Approximation capabilities. Fig. 7 shows a case study comparing different handle bases Ψ in deformation operator approximations to the S-ARAP. In this example, we use a relatively small sample size ($m = 17$ samples) and compare the deformations achieved using various bases. We show how the deformation changes as the handle moves from one side of the sampling region to the other. Fig. 11 depicts more approximation results for a set of general shapes.

Figure 2 shows (Φ, Ψ)-DefOp approximations of ARAP. In this case, Φ is a reduced spatial subspace with 100 basis functions, and the TPS handle basis contains 500 RBF. To show that the approximation is robust, we picked several handle positions at random, and compare them to the ground truth. It can be observed that in general, the approximation becomes better, that is, more similar to the ground truth, as more handle basis becomes larger and richer.

Convergence. In Fig. 8 we show a convergence graph of our approach, as the number of samples m increases. We compute an approximated deformation operator for the Aleph model (see Fig. 9), using up to 400 samples. For each approximation, we compute the error based on Eq. (8a), divided by the number of vertices, based on a set of 100 different samples. The bounding box of the model is the unit square. Note that polynomials have limited approximation capabilities, and so increasing the number of samples can only reduce the error to a limit. TPS on the other hand can, in theory, reduce the error to zero.

Performance. Our implementation was made in entirely in Matlab, using L-BFGS to compute the samples. We used an i7 machine clocking at 3.5 Ghz, with 64GB RAM. We compute the (Φ, Ψ)-deformation operator using the process described in Section 3.3. Afterwards, its evaluation requires only an evaluation of the handle basis functions at the specific position, followed by a matrix-vector product. This part can be computed at extremely fast rates, compared to the preprocess step. We evaluated the performance of this step on a much weaker machine, an i7, 1.8 Ghz, with 8 GB RAM. Refer to the included video, where we demonstrate several short deformation sessions. We summarize the timings for the different

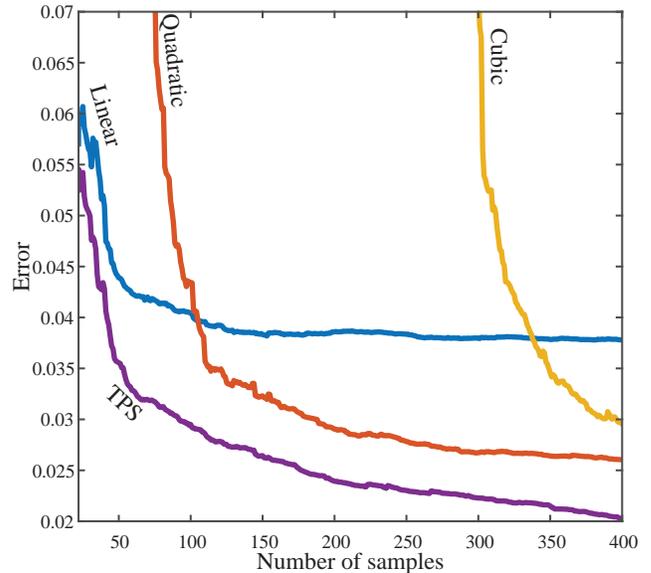


Figure 8: Convergence of the approximation as the number of samples m increases, for the Aleph model (See Fig. 9). Note that a minimum number of samples is required, depending on the number of handles (5 in this case) and the size of the basis, in order to use up all of the degrees of freedom in the model.

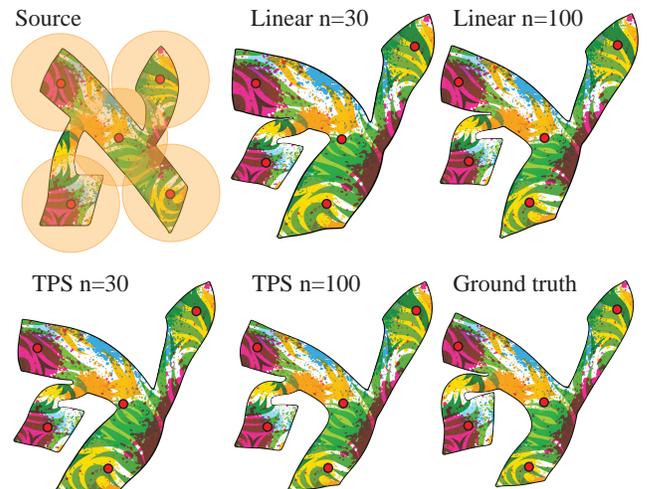


Figure 9: Convergence of the approximation as the number of samples increases, for the Aleph model, see text and Fig. 8 for more details.

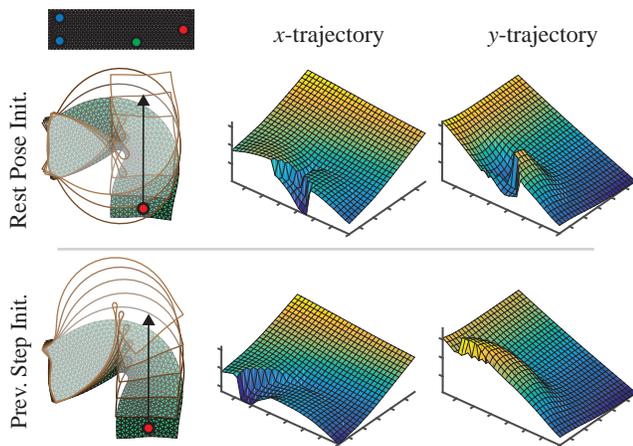


Figure 10: Variational deformations and trajectories based on global (top) and local (bottom) optima. Note the different positions of the discontinuities in the trajectories.

components in Table 1.

5 Conclusions, Limitations and Future Work

We have introduced a method for approximating complicated, non-linear deformation operator using a finite-basis deformation operator. We have demonstrated that such approximation produces comparable deformations to the non-linear operator in fraction of the time.

As mentioned before, variational deformation operators can have discontinuities. These are the result of defining the deformation operator using the minima of a non-convex deformation energy. Consequently, in some cases, two close handle positions can result in very different deformations (See Fig. 10). In practical applications, however, finding the global minimum is not guaranteed, nor desired. Usually, as the user drags a handle, the previous deformation is used as an initial guess for finding a nearby *local* minimum of the deformation energy, which serves as the new deformation. This results in a smoother deformation operator in some cases, but also introduces some ambiguity into the problem, since the deformation depends on previous deformations. In Fig. 10 we show how a smoother trajectory looks, when initializing the optimization problem with a previous deformation. In general, however, we cannot anticipate what the correct local minimum is, since it can change during interaction. We consider this a limitation of our approach, and an interesting avenue for further research. A possible resolution would be to consider utilizing a *multi-valued* approximation approach in order to reproduce the correct deformation.

Lastly, we note that in our setup, the positions of the handles in the rest pose (*i.e.*, \mathbf{p}_ℓ) remain fixed. However, in reality, the user may wish to reposition them without the need to repeat the preprocess computation. This might be possible if one manages to incorporate \mathbf{p}_ℓ as variables into our problem.

An interesting and natural future work direction is to extend the approach to 3D. The first step would be to examine the trajectories of points in 3D and determine whether they can be approximated well with a finite basis. We leave that for future work.

References

- BOOKSTEIN, F. L. 1989. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Trans. Pattern Anal. Mach. Intell.* 11, 6 (June), 567–585.
- BOTSCH, M., AND KOBELT, L. 2004. An intuitive framework for real-time freeform modeling. In *ACM SIGGRAPH 2004 Papers*, ACM, New York, NY, USA, SIGGRAPH '04, 630–634.
- BOTSCH, M., AND KOBELT, L. 2005. Real-time shape editing using radial basis functions. In *Computer Graphics Forum*, 611–621.
- BOTSCH, M., PAULY, M., GROSS, M., AND KOBELT, L. 2006. Primo: Coupled prisms for intuitive surface modeling. In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, SGP '06, 11–20.
- CHAO, I., PINKALL, U., SANAN, P., AND SCHRÖDER, P. 2010. A simple geometric model for elastic deformations. *ACM Trans. Graph.* 29, 4 (July), 38:1–38:6.
- FLOATER, M. S. 2003. Mean value coordinates. *Comput. Aided Geom. Des.* 20, 1 (Mar.), 19–27.
- HILDEBRANDT, K., SCHULZ, C., TYCOWICZ, C. V., AND POLTHIER, K. 2011. Interactive surface modeling using modal analysis. *ACM Trans. Graph.* 30, 5 (Oct.), 119:1–119:11.
- HORMANN, K., AND FLOATER, M. S. 2006. Mean value coordinates for arbitrary planar polygons. *ACM Transactions on Graphics* 25, 4 (Oct.), 1424–1441.
- HUANG, J., SHI, X., LIU, X., ZHOU, K., WEI, L.-Y., TENG, S.-H., BAO, H., GUO, B., AND SHUM, H.-Y. 2006. Subspace gradient domain mesh deformation. *ACM Trans. Graph.* 25, 3 (July), 1126–1134.
- IGARASHI, T., MOSCOVICH, T., AND HUGHES, J. F. 2005. As-rigid-as-possible shape manipulation. *ACM Trans. Graph.* 24, 3 (July), 1134–1141.
- JACOBSON, A., AND SORKINE, O. 2011. Stretchable and twistable bones for skeletal shape deformation. *ACM Transactions on Graphics (proceedings of ACM SIGGRAPH ASIA)* 30, 6, 165:1–165:8.
- JACOBSON, A., DENG, Z., KAVAN, L., AND LEWIS, J. 2014. Skinning: Real-time shape deformation. In *ACM SIGGRAPH 2014 Courses*.
- JOSHI, P., MEYER, M., DE ROSE, T., GREEN, B., AND SANOCKI, T. 2007. Harmonic coordinates for character articulation. *ACM Trans. Graph.* 26, 3 (July).
- JU, T., SCHAEFER, S., AND WARREN, J. 2005. Mean value coordinates for closed triangular meshes. *ACM Trans. Graph.* 24, 3 (July), 561–566.
- KAVAN, L., AND SORKINE, O. 2012. Elasticity-inspired deformers for character articulation. *ACM Transactions on Graphics (proceedings of ACM SIGGRAPH ASIA)* 31, 6, 196:1–196:8.
- KIRCHER, S., AND GARLAND, M. 2008. Free-form motion processing. *ACM Trans. Graph.* 27, 2 (May), 12:1–12:13.
- LÉVY, B., PETITJEAN, S., RAY, N., AND MAILLOT, J. 2002. Least squares conformal maps for automatic texture atlas generation. In *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques*, ACM, New York, NY, USA, SIGGRAPH '02, 362–371.

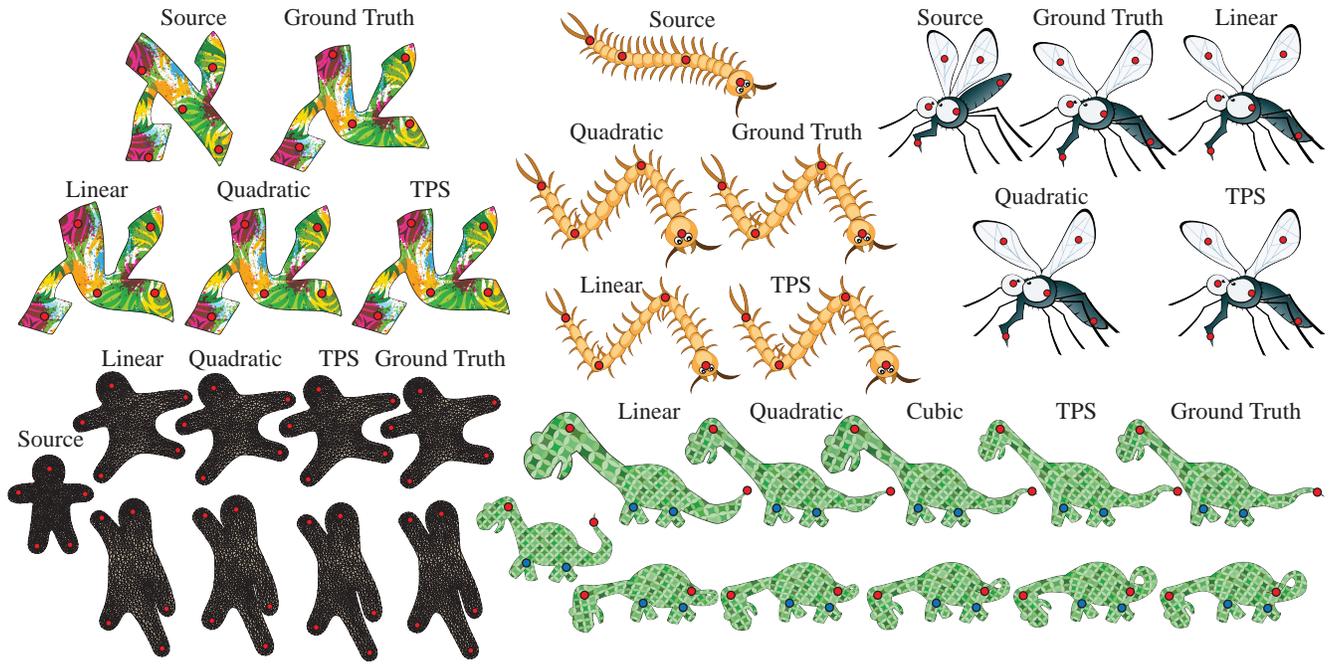


Figure 11: Various result obtained in different settings. In all of these examples, excluding the ground truths, deformations are achieved in real-time (see also the included video). Note that all of these approximation are visually very close to their respective ground truths

Model	Basis	#Tris	#Verts	#Handles	#Samples	Ψ	Φ	Preprocess (s)	LS (ms)	Evaluation (FPS)
Bar	TPS	1118	614	1	17	20	1118	2	2.5	$\geq 4K$
Dino	TPS	967	1706	2	361	367	967	171	43	$\geq 4K$
Aleph	Linear	5461	2889	5	500	11	5461	2500	11	$\geq 4K$
Aleph	Cubic	5461	2889	5	500	268	5461	2500	25	2600
Aleph	TPS	5461	2889	5	500	511	5461	2500	91	2050
Octopus	Linear	7258	4046	9	1000	19	100	1200	1	3500
Octopus	Quad	7258	4046	9	1000	190	100	1200	20	2250
Octopus	TPS	7258	4046	9	1000	1019	100	1200	110	1650
Crab	Linear	3118	2090	11	500	23	100	380	1	$\geq 4K$
Crab	Quad	3118	2090	11	500	276	100	380	16	2800
Crab	TPS	3118	2090	11	500	523	100	380	41	2300

Table 1: Timing for the different stages in our algorithm for several models presented in the paper, in several settings. Note that for the Octopus and Crab model we used a reduced subspace with 100 basis functions. Also note that on our machine the maximal framerate seems to be capped at around 4300 FPS.

LEWIS, J. P., CORDNER, M., AND FONG, N. 2000. Pose space deformation: A unified approach to shape interpolation and skeleton-driven deformation. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, SIGGRAPH '00, 165–172.

LIPMAN, Y., SORKINE, O., LEVIN, D., AND COHEN-OR, D. 2005. Linear rotation-invariant coordinates for meshes. *ACM Trans. Graph.* 24, 3 (July), 479–487.

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

LIU, L., ZHANG, L., XU, Y., GOTSMAN, C., AND GORTLER, S. J. 2008. A local/global approach to mesh parameterization. In *Proceedings of the Symposium on Geometry Processing*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, SGP '08, 1495–1504.

MANSON, J., AND SCHAEFER, S. 2011. Hierarchical deformation

of locally rigid meshes. *Computer Graphics Forum* 30, 8, 2387–2396.

MCADAMS, A., ZHU, Y., SELLE, A., EMPEY, M., TAMSTORF, R., TERAN, J., AND SIFAKIS, E. 2011. Efficient elasticity for character skinning with contact and collisions. *ACM Trans. Graph.* 30, 4 (July), 37:1–37:12.

MERRY, B., MARAIS, P., AND GAIN, J. 2006. Animation space: A truly linear framework for character animation. *ACM Trans. Graph.* 25, 4 (Oct.), 1400–1423.

MOHR, A., AND GLEICHER, M. 2003. Building efficient, accurate character skins from examples. In *ACM SIGGRAPH 2003 Papers*, ACM, New York, NY, USA, SIGGRAPH '03, 562–568.

SEDERBERG, T. W., AND PARRY, S. R. 1986. Free-form deformation of solid geometric models. In *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques*, ACM, New York, NY, USA, SIGGRAPH '86, 151–160.

SHEFFER, A., AND KRAEVOY, V. 2004. Pyramid coordinates for morphing and deformation. In *Proceedings of the 3D Data Processing, Visualization, and Transmission, 2Nd International Symposium*, IEEE Computer Society, Washington, DC, USA, 3DPVT '04, 68–75.

SLOAN, P.-P. J., ROSE, III, C. F., AND COHEN, M. F. 2001. Shape by example. In *Proceedings of the 2001 Symposium on Interactive 3D Graphics*, ACM, New York, NY, USA, I3D '01, 135–143.

SORKINE, O., AND ALEXA, M. 2007. As-rigid-as-possible surface modeling. In *Proceedings of EUROGRAPHICS/ACM SIGGRAPH Symposium on Geometry Processing*, 109–116.

SORKINE, O., COHEN-OR, D., LIPMAN, Y., ALEXA, M., RÖSSL, C., AND SEIDEL, H.-P. 2004. Laplacian surface editing. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, ACM, 175–184.

WANG, X. C., AND PHILLIPS, C. 2002. Multi-weight enveloping: Least-squares approximation techniques for skin animation. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ACM, New York, NY, USA, SCA '02, 129–138.

WANG, Y., JACOBSON, A., BARBIC, J., AND KAVAN, L. 2015. Linear subspace design for real-time shape deformation. *ACM Trans. Graph.* 34, 4.

WEBER, O., SORKINE, O., LIPMAN, Y., AND GOTSMAN, C. 2007. Context-aware skeletal shape deformation. *Computer Graphics Forum (Proceedings of EUROGRAPHICS)* 26, 3, 265–273.

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. *Computer Graphics Forum* 30, 5 (Aug.), 1533–1542. Proceedings of SGP.

YU, Y., ZHOU, K., XU, D., SHI, X., BAO, H., GUO, B., AND SHUM, H.-Y. 2004. Mesh editing with poisson-based gradient field manipulation. In *ACM Transactions on Graphics (TOG)*, vol. 23, ACM, 644–651.

ZHANG, J., DENG, B., LIU, Z., PATANÈ, G., BOUAZIZ, S., HORMANN, K., AND LIU, L. 2014. Local barycentric coordinates. *ACM Transactions on Graphics* 33, 6 (Dec.). Proceedings of SIGGRAPH Asia, to appear.

Appendix A

We derive a closed form expression of Eq. 8a for the general case where Φ is a reduced subspace. We assume that the spatial basis function φ_i are given as linear combinations of a set of primitive basis function, e.g. hat-functions, such that $\Phi = \mathbf{D}\tilde{\Phi}$. To make the derivation more concise, we write the (Φ, Ψ) -DefOp in matrix form.

$$\mathcal{D}_{\Phi, \Psi}(\mathbf{x}, \mathbf{Q}) := \left(\Phi(\mathbf{x})^T \mathbf{C}^\nu \Psi(\mathbf{Q}) \right)_\nu, \quad (12)$$

where $\mathbf{C}^\nu = (C_{ij}^\nu)_{ij}$ is a matrix containing the coefficients for $\nu = x, y$. Assuming $\mathcal{D}^E(\mathbf{x}, \mathbf{Q}_k)$ is expressed in the spatial basis $\tilde{\Phi}$, namely

$$\mathcal{D}_{\Phi}^E(\mathbf{x}, \mathbf{Q}_k) = \sum_{\tau} \tilde{\mathbf{C}}_{\tau k}^\nu \varphi_{\tau}(\mathbf{x}) = \left(\tilde{\mathbf{C}}_k^\nu \tilde{\Phi}(\mathbf{x}) \right)_\nu,$$

then we have,

$$\mathbf{E} = \sum_{k=1}^m \int_{\Omega} \left\| \mathcal{D}_{\Phi}^E(\mathbf{x}, \mathbf{Q}_k) - \mathcal{D}_{\Phi, \Psi}(\mathbf{x}, \mathbf{Q}_k) \right\|^2 d\mathbf{x} = \quad (13)$$

$$\sum_{k=1, \nu}^m \int_{\Omega} \left\| \tilde{\mathbf{C}}_k^\nu \tilde{\Phi}(\mathbf{x}) - \Psi(\mathbf{Q}_k)^T \mathbf{C}^\nu \Phi(\mathbf{x}) \right\|^2 d\mathbf{x} = \quad (14)$$

$$\sum_{k=1, \nu}^m \int_{\Omega} \left\| \tilde{\mathbf{C}}_k^\nu \tilde{\Phi}(\mathbf{x}) - \Psi_k^T \mathbf{C}^\nu \mathbf{D} \tilde{\Phi}(\mathbf{x}) \right\|^2 d\mathbf{x} = \quad (15)$$

$$\sum_{k=1}^m \int_{\Omega} \left\| \left(\tilde{\mathbf{C}}_k^\nu - \Psi_k^T \mathbf{C}^\nu \mathbf{D} \right) \tilde{\Phi}(\mathbf{x}) \right\|^2 d\mathbf{x}. \quad (16)$$

where we have made $\Psi_k^T := \Psi(\mathbf{Q}_k)^T$. By expanding (13) we get

$$\sum_{k, \nu} \left(\tilde{\mathbf{C}}_k^\nu - \Psi_k^T \mathbf{C}^\nu \mathbf{D} \right) A \left(\tilde{\mathbf{C}}_k^\nu - \Psi_k^T \mathbf{C}^\nu \mathbf{D} \right)^T \quad (17)$$

where $A_{i, i'} = \int_{\Omega} \tilde{\varphi}_i(\mathbf{x}) \tilde{\varphi}_{i'}(\mathbf{x}) d\mathbf{x}$ is the well-known mass matrix. By expanding (13) we get,

$$\mathbf{E} = \sum_{k, \nu} \Psi_k^T \mathbf{C}^\nu \mathbf{D} A \mathbf{D}^T (\mathbf{C}^\nu)^T \Psi_k - 2 \Psi_k^T \mathbf{C}^\nu \mathbf{D} A (\tilde{\mathbf{C}}_k^\nu)^T + (\tilde{\mathbf{C}}_k^\nu)^T A \tilde{\mathbf{C}}_k^\nu \quad (18)$$

This can be written using the matrix *trace* as follows,

$$\mathbf{E} = \sum_{\nu} \left[\text{tr} \left(\mathbf{C}^\nu \mathbf{D} A \mathbf{D}^T (\mathbf{C}^\nu)^T B \right) + \text{tr} \left(E^\nu \mathbf{C}^\nu \mathbf{D} A \right) \right] \quad (19)$$

where $B = \sum_{k=1}^m \Psi_k \Psi_k^T$ and $E^\nu = -2 \sum_{k=1}^m (\Psi_k)^T \tilde{\mathbf{C}}_k^\nu$.

The constraints (8b) and (8c) can be expressed as a linear system

$$\Psi_0^T \mathbf{C}^\nu = \mathbf{C}_0^\nu, \quad \mathbf{C}_0^\nu \mathbf{D} \Phi(\mathbf{x}) = \nu \quad (20)$$

$$\Psi_k^T \mathbf{C}^\nu \Phi(\mathbf{p}_\ell) = q_{k, \ell}^\nu \quad (21)$$

where Ψ_0 are the values of the spatial basis functions at the initial position, and \mathbf{C}_0^ν are the coefficient *vectors* for the spatial basis that reproduce the rest pose. These equations can be vectorized w.r.t. \mathbf{C}^ν using the Kronecker product and written as a large system

$$\text{Hvec}(\mathbf{C}^\nu) = b.$$

In order to solve (8), we write the Lagrangian of the problem,

$$\mathcal{L} = \mathbf{E} + \lambda^T (\text{Hvec}(\mathbf{C}^\nu) - b)$$

The derivative of (19) is

$$\nabla_{\mathbf{C}^\nu} \mathbf{E} = 2 \mathbf{D} A \mathbf{D}^T (\mathbf{C}^\nu)^T B + \mathbf{D} A E^\nu. \quad (22)$$

and by vectorizing we get

$$\nabla_{\text{vac}(\mathbf{C}^\nu)} \mathbf{E} = (B \otimes 2 \mathbf{D} A \mathbf{D}^T) \text{vec}(\mathbf{C}^\nu) + \mathbf{D} A E.$$

And the linear system to solve easily follows.