Polynomial 2D Green Coordinates for Polygonal Cages

Élie Michel
Adobe Research
France
emichel@adobe.com

Jean-Marc Thiery
Adobe Research
France
jthiery@adobe.com

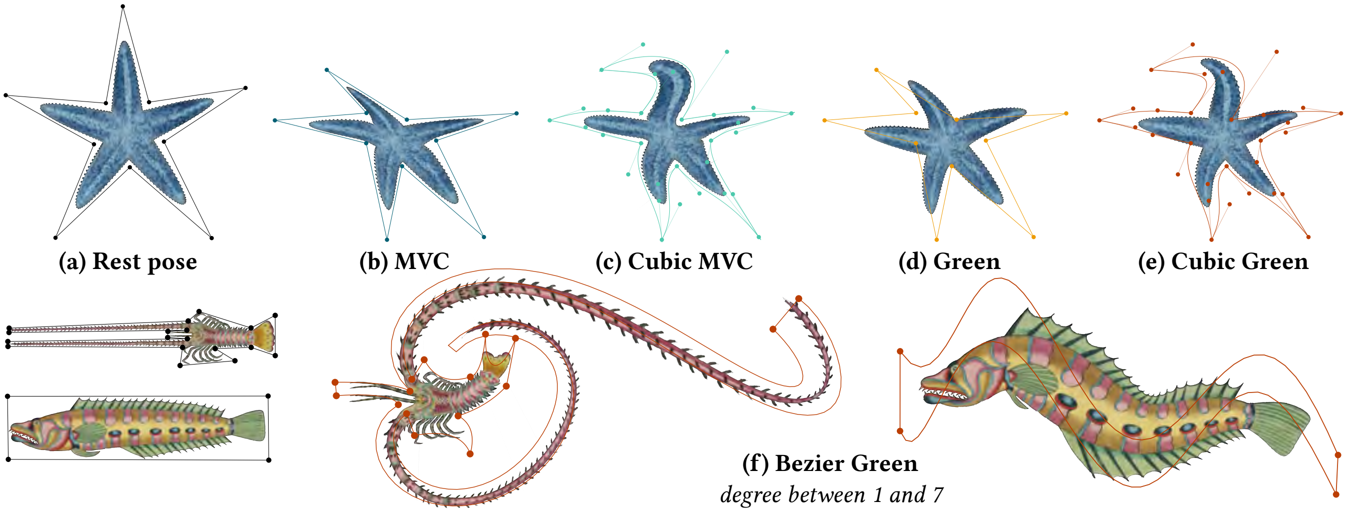


Figure 1: Top row: (a) input image and an embedding polygonal cage; (b-d) deformations obtained using Mean-Value coordinates, Cubic Mean-Value coordinates, and Green coordinates; (e) our conformal deformations obtained with cubic curves. Bottom row: more resuls of our approach, using polynomial curves of various orders (from 1 to 7).

ABSTRACT

Cage coordinates are a powerful means to define 2D deformation fields from sparse control points. We introduce Conformal polynomial Coordinates for closed polyhedral cages, enabling segments to be transformed into polynomial curves of any order. Extending classical 2D Green coordinates, our coordinates result in conformal harmonic deformations that are cage-aware. We demonstrate the usefulness of our technique on a variety of 2D deformation scenarios where curves allow artists to perform intuitive deformations with few input parameters. Our method combines the texture preservation property of conformal deformations together with the expressiveness offered by Bezier controls.

CCS CONCEPTS

• **Computing methodologies** → **Shape modeling**; **Parametric curve and surface models**; **Volumetric models**.

KEYWORDS

cage-based modeling, 2D shape deformation, harmonic functions, conformal deformations, curve-based deformations

ACM Reference Format:
Élie Michel and Jean-Marc Thiery. 2023. Polynomial 2D Green Coordinates for Polygonal Cages. In ,. ACM, New York, NY, USA, 9 pages.

preprint,
2023.

1 INTRODUCTION

Cage coordinates offer artists a powerful means to define scalar and vector fields over a whole planar 2D domain while manipulating only a sparse value constraint on the boundary of this domain. This boundary is typically formed by a polygon, and is referred to as the *cage* (see Fig. 1.a). Cage coordinates are used for instance to easily define deformation fields or color fields in image editing applications. In their simplest form, the user specifies a value f_i for each cage vertex v_i, from which the whole function f is inferred using some regularity constraint. Importantly, the value $f(\eta)$ at location η inside the cage is expressed as a linear combination of the sample values: $f(\eta) = \sum_i \lambda_i(\eta) f_i$. The barycentric weights $\lambda_i(\eta)$ are the *cage coordinates* of the point η. This formal constraint leaves flexibility about the type of user input and regularization prior while ensuring that the cost of evaluating f is bounded by the complexity of the sparse cage. For instance Mean-Value coordinates (MVC) [Floater 2003; Hormann and Floater 2006], Positive Mean-Value coordinates (PMVC) [Lipman et al. 2007] or Harmonic coordinates (HC) [Joshi et al. 2007] belong to this category. Some other types of coordinates use gradients and higher-order derivatives, in order to obtain various properties like, for example, gradient interpolation and/or control [Hou et al. 2017, 2018; Ilbery et al. 2013; Sun et al. 2012], or infer specific differential properties on the induced deformation. Such examples include Green coordinates (GC) [Lipman et al. 2008], that express each point as a blending of both the cage vertex positions and the cage edge normals: $\eta = \sum_i \phi_i(\eta) v_i + \sum_j \psi_j(\eta) n_j$.

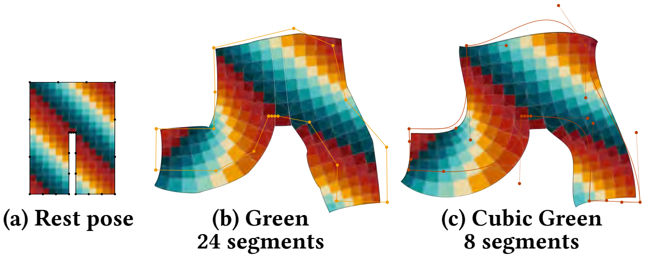


Figure 2: Using the same number of control points, deforming 8 cubic curves (our approach, c) results in smoother and more intuitive editing than deforming 24 straight segments (b).

The mathematical properties of those coordinates as spatial functions, such as smoothness, impact directly the quality of the diffused function. Our work focuses on the case of cage-based deformation, where the function f returns the deformed location $\bar{\eta}$ of a point η. In this context, the properties of (i) smoothness, (ii) interpolation, (iii) detail-preservation and (iv) intuitiveness of the controls are in general the most important ones. Before detailing all of those properties, we can already stress that they generally conflict each others; it is impossible to obtain them all at the same time.

For example, enforcing exact boundary interpolation (using, e.g., MVC in Fig. 1.b) prevents the deformation field from being more than continuous at the cage vertices. Reversely, detail preservation offered by *conformal transformations*, that preserve local angles and thus local texture aspect exactly, renders impossible interpolating the deformation across the cage faces (see GC in Fig. 1.d).

Finally, considering curved cage elements in lieu of simple straight segments offers a natural means to control both the tangential stretch and the curvature along the deformation cage (see Fig. 2). Such controls were recently introduced in the context of cage-based deformation with Cubic Mean-Value coordinates [Li et al. 2013] (see Cubic MVC in Fig. 1.c), that extend MVC to cages where rest segments are deformed into cubic arcs. Being based on MVC, the resulting deformations are interpolating, at the cost of being only continuous at the boundary vertices and non-conformal.

We present in this work the first 2D cage-based deformation method that allows using polynomial deformation functions of *any* degree (see Fig. 1.e, 1.f, and 2), that are conformal by construction.

1.1 Related Work

A wide variety of modalities for deforming images and 2D vector graphics have been proposed, such as for instance Free Form deformations [Sederberg and Parry 1986], Moving-Least-Squares deformations [Schaefer et al. 2006], or brush-based deformations [De Goes and James 2017]. We focus in this section on the existing works on cage-based deformations, which are closest to our work. Some key concepts apply equivalently in 2D and in 3D, and we discuss related works for 3D deformations when most appropriate.

Interpolating coordinates. Mean-Value coordinates [Floater 2003; Hormann and Floater 2006; Ju et al. 2005] (MVC) make use of the *Mean-value theorem* of harmonic functions, and allow interpolating scalar functions across the cage faces everywhere in space (not just the interior of the cage). Their closed-form expression makes them an ideal subspace in several multiresolution deformation frameworks [Huang et al. 2006] as well as in applications beyond deformation, such as image editing and cloning [Farbman et al. 2009]. In exchange for allowing for extrapolation outside the cage, those coordinates may however be negative, which can result in counter-intuitive deformation/diffusion behaviors.

To solve this issue, Positive Mean-value coordinates [Lipman et al. 2007] (PMVC) consider the portion of the cage that is visible from the evaluation point η, ensuring positivity of the created coordinates. Making use of the GPU to solve for visibility, those coordinates do not come with a closed-form expression.

Harmonic coordinates [Joshi et al. 2007] share similar properties with PMVC: they are also interpolating and positive, as well as defined inside the cage only. They also do not come with closed-form expressions allowing for simple evaluation.

Finally, Cubic MVC [Li et al. 2013] extend MVC, and depart strongly from the previously-cited methods in that they allow deforming each cage segment into curves, specifically cubic arcs. In our work, we aim at providing similar deformation controls, while trading interpolation for conformality and detail preservation.

Note that, all the methods cited above can also be controlled by a curve network rather than a single cage: each region is deformed independently and the continuity across edges of the network is trivially ensured by the *interpolation* property.

Approximating coordinates. Green coordinates (GC), introduced by Lipman et al. [2008] for 2D and 3D deformation, are to this date the main cage-based technique that does not produce an interpolating deformation field. They do however create *conformal* deformations, which guarantees exact local aspect preservation. GC are formally derived from Green's third identity, that allows expressing any harmonic function from the diffusion of its boundary Dirichlet and Neumann conditions. Note that Weber and colleagues introduced in [2009] Cauchy coordinates, that are shown to be equivalent to 2D Green coordinates. Following this, Weber et al. analyzed in [2011] the properties of complex 2D barycentric mappings and demonstrated their expressive power compared with more standard scalar barycentric blending of vector-valued cages.

Weber et al. introduced later Biharmonic coordinates [2012]. Those rely on a higher-order Green identity and require twice the amount of coordinates compared to GC, as twice boundary conditions are taken into account. While the direct setting of these additional boundary conditions has not yet been demonstrated to be an intuitive deformation tool, this results in more flexibility on the deformation energies minimized within a variational framework and allows for higher-quality deformations in this context.

Our work is closely related to GC and Cubic MVC, as we provide harmonic 2D conformal deformations based on Green's third identity for the case of cages made of straight segments being deformed into polynomial curves of any degree. We therefore focus on these two methods to illustrate the novelty of our work, both scientifically and in terms of practical applications.

1.2 Contributions

We present in this paper the following technical contributions:

- a formulation of 2D Green coordinates associated with cages made of properly-oriented, non-intersecting curves, baking in the conformality property into the coordinates;
- closed-form expressions for 2D cages made of segments, that can be deformed into polynomial curves of any degree.

To the best of our knowledge, our work is the first to introduce closed-form expressions for conformal 2D deformations inferred by non-straight curve guides.

2 BACKGROUND: 2D CAGE-BASED CONFORMAL DEFORMATIONS USING GREEN COORDINATES

Using Green's third identity, we can express a harmonic function f in a bounded 2D domain Ω from its boundary conditions as

$$f(\eta) = \underbrace{\int\limits_{\xi\in\partial\Omega} f(\xi)\frac{\partial G}{\partial n_\xi}(\xi,\eta)d\xi}_{:=f_{\mathcal{D}}(\eta)} + \underbrace{\int\limits_{\xi\in\partial\Omega} -G(\xi,\eta)\frac{\partial f}{\partial n_\xi}(\xi)d\xi}_{:=f_{N}(\eta)}, \tag{1}$$

with $G(\xi,\eta) := \frac{1}{2\pi}\log(\|\xi-\eta\|)$ solution to $\triangle_1 G(\xi,\eta) = \triangle_2 G(\xi,\eta) = \delta(\|\xi-\eta\|)$ and n the unit normal of the cage at point ξ. The term $f_{\mathcal{D}}(\eta)$ (resp. $f_N(\eta)$) corresponds to the contribution given by the diffusion of the Dirichlet (resp. Neumann) boundary condition.

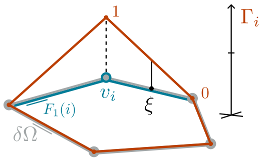
We consider in the following the cage $\partial\Omega$ as a non-intersecting closed polygon made of vertices $v_i \in \mathcal{V}$ connected by edges $e_j \in \mathcal{E}$ (oriented Counter-Clock-Wise by convention). We note deformed quantities with a bar $(\bar{\cdot})$ and rest-pose quantities without it.

In the Green coordinates proposed by Lipman et al. [2008], the following Dirichlet and Neumann conditions are used:

$$f(\xi) = \sum_i \Gamma^i(\xi)\bar{v}_i \tag{2}$$

$$\frac{\partial f}{\partial n_\xi}(\xi) = \sigma_j \bar{n}_j \;\forall\xi\in e_j, \tag{3}$$

where Γ^i is the "hat basis function" that takes value 1 on vertex i, 0 at the other vertices and is linear on each edge, and σ_j and $\bar{n}_j$ are respectively the stretch factor and the normal of the linearly-deformed edge e_j. Both quantities are constant across e_j as they depend on the (constant) edge linear map only.



Injecting these boundary conditions in Eq. 1 results in the following compact expression for f:

$$\bar{\eta} := f(\eta) = \sum_{i\in\mathcal{V}} \phi_i(\eta)\bar{v}_i + \sum_{j\in\mathcal{E}} \psi_j(\eta)\sigma_j\bar{n}_j \text{ , with} \tag{4}$$

$$\phi_i(\eta) := \int\limits_{\xi\in F_1(i)} \Gamma^i(\xi)\frac{\partial G}{\partial n_\xi}(\xi,\eta)d\xi \tag{5}$$

$$\psi_j(\eta) := \int\limits_{\xi\in e_j} -G(\xi,\eta)d\xi \tag{6}$$

Lipman and colleagues prove in [2008] the following lemma:

Lemma 1 (Lipman's et al. conformality condition). *Assuming a deformation function f given by Eq. (4), f is conformal everywhere strictly inside Ω iif the stretch factor σ_j on edge j is the deformed-edge-length by rest-edge-length ratio:*

$$\sigma_j = \frac{\|\bar{e}_j\|}{\|e_j\|} \;. \tag{7}$$

3 2D CONFORMAL GREEN COORDINATES FOR CURVE-CAGES

We build atop the previously-introduced 2D Green coordinates, and present here our formulation for Green coordinates for cages made of non-intersecting curves $\{c_i\}_i$ that form the boundary $\partial\Omega$ (oriented CCW) of a compact domain Ω.

In this context, Eq. (1) still holds, and, for the sake of simplicity, we can focus our analysis on the contribution of a single curve $c : [0,1] \mapsto \mathbb{R}^2$ (still noting $\bar{c}$ the deformed curve). We note $f^c_{\mathcal{D}}(\eta) := \int_{\xi\in c} f(\xi)\frac{\partial G}{\partial n_\xi}(\xi,\eta)d\xi$ its contribution to the Dirichlet term, and $f^c_{N}(\eta) := \int_{\xi\in c} -G(\xi,\eta)\frac{\partial f}{\partial n_\xi}(\xi)d\xi$ its contribution to the Neumann term.

3.1 General formulation

We present here the general case where the rest curve c and deformed curve $\bar{c}$ can be any C^1 arc. We will then present closed-form expressions for simple families of curves.

Dirichlet term. We can rewrite the Dirichlet term as:

$$f^c_{\mathcal{D}}(\eta) = \int\limits_{t=0}^{1} \bar{c}(t)\; \nabla_1 G(c(t),\eta)\cdot n_{c(t)}\|c'(t)\|dt,$$

where we simply introduced the natural parameterization of the curve in the integral (noting $c'(t)$ the speed vector at t, the infinitesimal linear element is given by $d\xi = \|c'(t)\|dt$ for $\xi = c(t)$).

Since the curve c is oriented CCW, we can write that $n_{c(t)}\|c'(t)\| = c'(t)^\perp$, where $(a,b)^\perp = (b,-a)$. Noting that $\nabla_1 G(\xi,\eta) = \frac{\xi-\eta}{2\pi\|\xi-\eta\|^2}$, we obtain:

$$f^c_{\mathcal{D}}(\eta) = \int\limits_{t=0}^{1} \bar{c}(t)\frac{(c(t)-\eta)\cdot c'(t)^\perp}{2\pi\|c(t)-\eta\|^2}dt. \tag{8}$$

Neumann term. Introducing the conformality condition of Lemma 1 leads to the following reformulation of the Neumann term:

$$\begin{aligned} f^c_{N}(\eta) &= \int\limits_{t\in 0}^{1} -G(c(t),\eta)\overbrace{\frac{\bar{c}'(t)^\perp}{\|\bar{c}'(t)^\perp\|}}^{\bar{n}}\overbrace{\frac{\|\bar{c}'(t)\|}{\|c'(t)\|}}^{\sigma}\overbrace{\|c'(t)\|dt}^{d\xi} \\ &= \int\limits_{t\in 0}^{1} \frac{-1}{2\pi}\log\left(\|c(t)-\eta\|\right)\bar{c}'(t)^\perp dt \end{aligned} \tag{9}$$

To understand why this is the right formulation for the Neumann term incorporating the conformality condition of Lemma 1, one can discretize the curve c into a polyline $\hat{c}$ ($\hat{c} : [0,1] \mapsto \mathbb{R}^2$, with $\hat{c}(t_k) = c_k$, $\hat{c}$ affine on the segment $[t_k, t_{k+1}]$, with $t_k = k/N$), and notice that Lemma 1 implies then:

$$
\begin{aligned}
f_{\mathcal{N}}^{\hat{c}}(\eta) &= \int\limits_{\xi\in\hat{c}} -G(\xi,\eta)\frac{\partial f}{\partial n_\xi}(\xi)d\xi = \sum_{k=0}^{N-1} \int\limits_{\xi\in[c_k,c_{k+1}]} -G(\xi,\eta)\frac{\partial f}{\partial n_\xi}(\xi)d\xi \\
&= \sum_{k=0}^{N-1}\left(\int\limits_{\xi\in[c_k,c_{k+1}]} -G(\xi,\eta)d\xi\right)\sigma_k\bar{n}_k \qquad \textit{(formulation of Lipman et al.)} \\
&= \sum_{k=0}^{N-1}\left(\int\limits_{t=t_k}^{t_{k+1}} -G(\hat{c}(t),\eta)\|\hat{c}'(t)\|dt\right)\frac{(\bar{c}_{k+1}-\bar{c}_k)^\perp}{\|\bar{c}_{k+1}-\bar{c}_k\|}\frac{\|\bar{c}_{k+1}-\bar{c}_k\|}{\|c_{k+1}-c_k\|} \\
&= \sum_{k=0}^{N-1}\int\limits_{t=t_k}^{t_{k+1}} -G(\hat{c}(t),\eta)\frac{(\bar{c}_{k+1}-\bar{c}_k)^\perp}{\|\bar{c}_{k+1}-\bar{c}_k\|}\frac{\|\bar{c}_{k+1}-\bar{c}_k\|}{\|c_{k+1}-c_k\|}\|\hat{c}'(t)\|dt \qquad (10)
\end{aligned}
$$

where $\psi_k(\eta)$ is the usual Green coordinate of η associated with segment $[c_k,c_{k+1}]$, $\bar{n}_k := \frac{(\bar{c}_{k+1}-\bar{c}_k)^\perp}{\|\bar{c}_{k+1}-\bar{c}_k\|}$ is the unit normal on that segment, and $\sigma_k := \frac{\|\bar{c}_{k+1}-\bar{c}_k\|}{\|c_{k+1}-c_k\|}$ is the deformed length over initial length ratio advocated by Lipman et al., resulting in conformal deformations for *any* such polygonal approximation of the curve c. Taking the limit $N \longrightarrow \infty$ results in point-wise convergence towards Eq. (9) (by definition of the Riemann integral).

To formally guarantee conformality of the limit, we need to ensure uniform convergence as well around the neighborhoods of each point η. We sketch a concise proof in the following.

Since we consider points η *strictly inside the cage* (say, at distance 3α from $\partial\Omega$), there exist N^* such that the open disk D_α of radius α and center η is at distance at least α from $\partial\Omega$ and its polygonal approximation $\partial\Omega_N$ for $N > N^*$.

Since all terms we consider (G and all its derivatives, curves c and all their derivatives) are bounded, we can show uniform convergence of Eq. (10) towards Eq. (9) (and similarly for their derivatives w.r.t. η) inside D_α. This is enough to prove that the derivatives match point-wise at the limit (i.e., switching limit and differentiation operators is permitted).

Taking all input curves of $\partial\Omega$ into account and considering the Dirichlet term as well, we obtain that the presented polygonal approximation (that is conformal at η for any $N > N^*$) converges uniformly towards our formulation, both in value and derivative w.r.t. η. Since the conformality penalty is a simple combination of the derivatives ($E_{\mathsf{conf}}(h) := \|\partial_y(h) - \partial_x(h)^\perp\|^2$ for any function h), and any polygonal approximation f_N is conformal (i.e., $E_{\mathsf{conf}}(f_N) = 0$ for $N > N^*$), we can conclude that

$$
\begin{aligned}
E_{\mathsf{conf}}(f) &= E_{\mathsf{conf}}\left(\lim_{N\to\infty} f_N\right) \\
&= \lim_{N\to\infty} E_{\mathsf{conf}}(f_N) \\
&= 0
\end{aligned}
$$

3.2 Straight segments to polynomial curves

We present now closed-form expressions for a simple family of curves: *straight regular segments* at encoding (i.e., $c(t) = \sum_{k=0}^{1} t^k c_k$) being deformed into polynomials of order N_c (i.e., $\bar{c}(t) = \sum_{m=0}^{N_c} t^m \bar{c}_m$).

3.2.1 Dirichlet term. The Dirichlet term can be rewritten as the integral of a rational fraction:

$$
\begin{aligned}
f_{\mathcal{D}}^{c}(\eta) &= \int\limits_{t=0}^{1} \bar{c}(t)\frac{(c(t)-\eta)\cdot c'(t)^\perp}{2\pi\|c(t)-\eta\|^2}dt \\
&= \int\limits_{t=0}^{1}\sum_{m=0}^{N_c} t^m\bar{c}_m\frac{(c_0-\eta+tc_1)\cdot c_1^\perp}{2\pi\|c(t)-\eta\|^2}dt \\
&= \sum_{m=0}^{N_c}\underbrace{\left(F_m^{(c,\eta)}(c_0-\eta)\cdot c_1^\perp\right)}_{:=\,\phi_m^{(c,\eta)}}\bar{c}_m, \qquad (11)
\end{aligned}
$$

where we define F as:

$$
F_n^{(c,\eta)} := \int\limits_{t=0}^{1}\frac{t^n}{2\pi\|\eta-\sum_{i=0}^{1} t^i c_i\|^2}dt \qquad (12)
$$

A closed-form expression of $F_n^{(c,\eta)}$ is given in Section 3.3.

3.2.2 Neumann term. The Neumann term can be rewritten as

$$
\begin{aligned}
f_{\mathcal{N}}^{c}(\eta) &= \frac{-1}{2\pi}\int\limits_{t=0}^{1}\log\left(\|c(t)-\eta\|\right)\bar{c}'(t)^\perp dt \\
&= \frac{-1}{2\pi}\left[\log\left(\|c(t)-\eta\|\right)\bar{c}(t)^\perp\right]_{t=0}^{1} + \int\limits_{t=0}^{1}\frac{(c(t)-\eta)\cdot c'(t)}{2\pi\|c(t)-\eta\|^2}\bar{c}(t)^\perp dt \\
&= \frac{-1}{2\pi}\left[\log\left(\|c(t)-\eta\|\right)\bar{c}(t)^\perp\right]_{t=0}^{1} + \int\limits_{t=0}^{1}\frac{(c_0-\eta)\cdot c_1 + t\|c_1\|^2}{2\pi\|c(t)-\eta\|^2}\bar{c}(t)^\perp dt \\
&= \sum_{m=1}^{N_c}\underbrace{\left(F_m^{(c,\eta)}(c_0-\eta)\cdot c_1 + F_{m+1}^{(c,\eta)}\|c_1\|^2 - \frac{\log\|c_0+c_1-\eta\|}{2\pi}\right)}_{:=\,\psi_m^{(c,\eta)}}\bar{c}_m^\perp \qquad (13)
\end{aligned}
$$

Note that $\psi_0^{(c,\eta)} = 0$, which is trivial to see beforehand since $\bar{c}_0$ does not appear in the expression of $\bar{c}'(t)$ (see first line of Eq. (13)).

3.2.3 Final expression. The final contribution of curve c (with deformed value $\bar{c}$) to the deformation of η is:

$$
f_{\mathcal{D}}^{c}(\eta) + f_{\mathcal{N}}^{c}(\eta) = \sum_{j=0}^{N_c}\phi_j^{(c,\eta)}\bar{c}_j + \sum_{m=1}^{N_c}\psi_m^{(c,\eta)}\bar{c}_m^\perp
$$

Considering the whole contour $\partial\Omega$, we obtain:

$$
\bar{\eta} = f(\eta) = \sum_{c\in\partial\Omega}\sum_{n=0}^{N_c}\left(\phi_n^{(c,\eta)}\bar{c}_n + \psi_n^{(c,\eta)}\bar{c}_n^\perp\right) \qquad (14)
$$

3.3 Closed-form expression of F_n

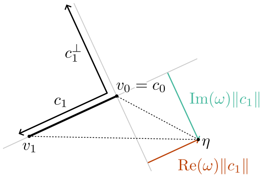
We find the following closed-form expression for $F_n^{(c,\eta)}$:

$$F_n^{(c,\eta)} = \frac{\operatorname{Im}\left(\omega^n D(\omega) + U_n(\omega)\right)}{2\pi \|c_1\|^2 \operatorname{Im}(\omega)} \tag{15}$$

$$\text{with } D(\omega) := \operatorname{Log}(1-\omega) - \operatorname{Log}(0-\omega) \tag{16}$$

$$\text{and } U_n(\omega) := \sum_{k=1}^{n-1} \frac{\omega^k}{n-k} \tag{17}$$

where ω is any of the complex roots of the second order polynomial $\|\eta - \sum_{i=0}^{1} t^i c_i\|^2$ and Log is the complex logarithm. Note that ω is the local coordinates of η in the frame relative to the edge (or its conjugate):


$c_1^\perp$
$v_0 = c_0$
$\operatorname{Im}(\omega)\|c_1\|$
c_1
η
v_1
$\operatorname{Re}(\omega)\|c_1\|$

$$\omega = \frac{c_1 + i c_1^\perp}{\|c_1\|^2} \cdot (\eta - c_0) \tag{18}$$

Mathematical derivation. We prove in the following Eq. 15. The denominator of the integrand is a real polynomial of degree 2 so it can be decomposed into a product of complex monomials:

$$F_n^{(c,\eta)} = \int_{t=0}^{1} \frac{t^n}{P(t)} dt \ \text{, with } \ P(x) =: A \prod_{k=1}^{2} (x - \omega_k) \ ,$$

where $A = 2\pi\|c_1\|^2$ is a normalization factor. Since $P(t)$ is always positive (it is originally a squared norm), its roots are pair-wise conjugate of each others: $\omega_2 = \omega_1^*$.

We also note $Q_k = \frac{P(x)}{x - \omega_k}$ the polynomial P without its k-th monomial, so that we can write the following (assuming that x equals none of the ω_k):

$$\sum_{k=1}^{2} \frac{x^n}{x - \omega_k} = \frac{x^n}{P(x)} \sum_{k=1}^{2} Q_k(x)$$

We write $Q = \sum_k Q_k$ and since this sum of polynomials has degree 1 we can also write $Q(x) = \sum_{j=0}^{1} q_j x^j$ where $q_j \in \mathbb{C}$:

$$\sum_{k=1}^{2} \frac{x^n}{x - \omega_k} = \sum_{j=0}^{1} q_j \frac{x^{n+j}}{P(x)} \tag{19}$$

We then use the following lemma:

Lemma 2. *Let $\omega \in \mathbb{C}$ and $g_n^{(\omega)} : x \mapsto \frac{x^n}{x-\omega}$. A primitive of $g_n^{(\omega)}$ for x such that $x - \omega \notin \mathbb{R}^-$ is $G_n^{(\omega)} : x \mapsto \omega^n \operatorname{Log}(x-\omega) + \sum_{l=1}^{n} \frac{x^l}{l} \omega^{n-l}$.*

Proof. We can see that if $x - \omega \notin \mathbb{R}^-$, we have

$$\begin{aligned} G_n^{(\omega)\prime}(x) &= \frac{\omega^n}{x-\omega} + \sum_{l=1}^{n} x^{l-1}\omega^{n-l} \\ &= \frac{\omega^n}{x-\omega} + \sum_{l=1}^{n} \frac{x^{l-1}\omega^{n-l}(x-\omega)}{x-\omega} \\ &= \frac{\omega^n}{x-\omega} + \sum_{l=1}^{n} \frac{x^l \omega^{n-l}}{x-\omega} - \sum_{l'=0}^{n-1} \frac{x^{l'}\omega^{n-l'}}{x-\omega} \\ &= \frac{x^n}{x-\omega} = g_n^{(\omega)}(x) \end{aligned}$$

□

Since η is strictly inside the cage, none of the ω_k belongs to the range $(0, 1)$. Hence we can apply this lemma to integrate Eq. 19:

$$\left[\sum_{k=1}^{2} \omega_k^n \operatorname{Log}(x - \omega_k) + \sum_{k=1}^{2}\sum_{l=1}^{n} \frac{x^l}{l} \omega_k^{n-l} \right]_{x=0}^{1} = \sum_{j=0}^{1} q_j F_{n+j}^{(c,\eta)}$$

We remark that, $\forall x \in \mathbb{R}$:

$$\begin{aligned} \omega_2^n \operatorname{Log}(x - \omega_2) &= \left(\omega_1^*\right)^n \operatorname{Log}\left(x - \omega_1^*\right) \\ &= \left(\omega_1^n \operatorname{Log}\left(x - \omega_1\right)\right)^* . \end{aligned}$$

Since $\forall z \in \mathbb{C}, z + z^* = 2\operatorname{Re}(z)$, the previous equation becomes:

$$2\operatorname{Re}\left(\omega_1^n D(\omega_1) + \sum_{l=1}^{n} \frac{\omega_1^{n-l}}{l}\right) = \sum_{j=0}^{1} q_j F_{n+j}^{(c,\eta)}$$

where we recall that $D(\omega) = \operatorname{Log}(1-\omega) - \operatorname{Log}(0-\omega)$.

With the notation $\omega_1 = \omega$ and $\omega_2 = \omega^*$, we have:

$$\begin{aligned} P(x) &= A(x-\omega)(x-\omega^*) \\ Q(x) &= 2A(x - \operatorname{Re}(\omega)). \end{aligned}$$

It follows that:

$$\frac{1}{A}\operatorname{Re}\left(\omega^n D(\omega) + \sum_{l=1}^{n} \frac{\omega^{n-l}}{l}\right) = F_{n+1}^{(c,\eta)} - \operatorname{Re}(\omega) F_n^{(c,\eta)} .$$

We note $V_n(\omega) = \omega^n D(\omega) + U_n(\omega)$, and thus derive:

$$F_{n+1}^{(c,\eta)} = \operatorname{Re}(\omega) F_n^{(c,\eta)} + \frac{1}{A}\operatorname{Re}\left(V_n(\omega) + \frac{1}{n}\right) \tag{20}$$

with the convention that $\frac{1}{n} = 0$ if $n = 0$ to alleviate notations. Note that in the implementation we use $W_n = V_n + 1/n$ to get a nicer recurrent relation. If we assume recursively that $F_n^{(c,\eta)} = \frac{\operatorname{Im}(V_n(\omega))}{A\operatorname{Im}(\omega)}$, we find that:

$$\begin{aligned} A\operatorname{Im}(\omega) F_{n+1}^{(c,\eta)} &= \operatorname{Re}(\omega)\operatorname{Im}(V_n(\omega)) + \operatorname{Im}(\omega)\operatorname{Re}(V_n(\omega)) + \frac{\operatorname{Im}(\omega)}{n} \\ &= \operatorname{Im}(\omega V_n(\omega)) + \operatorname{Im}\left(\frac{\omega}{n}\right) \\ &= \operatorname{Im}\left(\sum_{k=1}^{n-1} \frac{\omega^{n+1-k}}{k} + \frac{\omega}{n}\right) \\ &= \operatorname{Im}(V_{n+1}(\omega)) \end{aligned}$$

Thus we get $F_{n+1}^{(c,\eta)} = \frac{\operatorname{Im}(V_{n+1}(\omega))}{A\operatorname{Im}(\omega)}$. We also verify that $F_0 = \frac{Im(V_0(\omega))}{A\operatorname{Im}(\omega)} = \frac{Im(D(\omega))}{A\operatorname{Im}(\omega)}$, which ends the proof.

3.4 Implementation details

We give the encoding procedure for straight rest edges in Alg. 1.

Encoding. In practice, we use the relation of Eq. (20) to recursively compute the terms $F_n^{(c,\eta)}$ up to $n = N + 1$:

$$\begin{aligned} W_0 &= D(\omega) \quad &;\quad F_0 &= \frac{\operatorname{Im}(BW_0)}{\operatorname{Im}(\omega)} \\ W_{n+1} &= \omega W_n + \tfrac{1}{n+1} \quad &;\quad F_{n+1} &= \operatorname{Re}(\omega F_n + BW_n), \end{aligned}$$

where $B = \frac{1}{2\pi\|c_1\|^2} \in \mathbb{R}$. We use the following expression for $D(\omega)$:

$$D(\omega) = \frac{1}{2}\log\left(1 + \frac{1 - 2\operatorname{Re}(\omega)}{\|\omega\|^2}\right) + i\ \operatorname{atan2}\left(\operatorname{Im}(\omega), \|\omega\|^2 - \operatorname{Re}(\omega)\right) \tag{21}$$

Numerical stability. $D(\cdot)$ is well defined for $\eta \notin c$: The $\log(\cdot)$ is ill-defined only for $\|\omega\|^2 = 0$ or $1 - 2\operatorname{Re}(\omega) \leq -\|w\|^2$, which is equivalent to $\omega = 0$ or $\omega = 1$ (i.e., for η on the extremities of c). The $\arctan(\cdot)$ is well defined for $\omega \neq 0$ or $\omega \neq 1$ as well.

Similarly, F_0 is well defined as long as $\operatorname{Im}(\omega) \neq 0$ (i.e., η not on the line directed by c). For $\operatorname{Im}(\omega) = 0$, the imaginary part of $D(\omega)$ is equivalent to $\operatorname{Im}(\omega)/(\|\omega\|^2 - \operatorname{Re}(\omega))$ in this case (since $\tan(x) \simeq x$ for $x \simeq 0$), and therefore we obtain the special case for $F_0 : F_0 = B/(\|\omega\|^2 - \operatorname{Re}(\omega))$ when $\operatorname{Im}(\omega) = 0$, which is again valid as long as $\omega \neq 0$ and $\omega \neq 1$ (i.e., for η on the extremities of c).

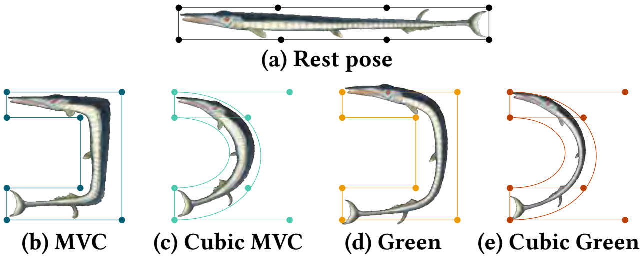
Run-time. Our coordinates being based on the integral convolutions of *monomials*, the artist can precompute the coordinates up to a given order N, and still choose at run-time to change the degree $k \leq N$ of the editing curves without coordinate recomputation. Indeed, computing the deformation simply requires decomposing the artist's Bézier curves onto the usual monomials basis, which can be done per-curve once at every frame without hindering performances. Such flexibility could not be as easily obtained, should we have used the Bézier basis instead. If the artist desires manipulating curves of higher degree than originally anticipated however, it will be necessary to recompute new monomial coordinates. While the low-degree coordinates will match the ones that were previously computed, it is not clear that past computations can be reused without having to store many intermediate ones (detailed in Alg. 1), and recomputation from scratch appears necessary in this case.

ALGORITHM 1: Encoding of a rest position η into our polynomial Green coordinates. This must be done for each CCW-oriented edge c of the rest cage. N is the degree of the deformed curve $\bar{c}$.

```
Data: A point $\eta$ from inside the rest cage and a directed edge $c$ of
      the rest cage represented by its vertices $v_0$ and $v_1$.
Result: The arrays $\boldsymbol{\phi}$ and $\boldsymbol{\psi}$ holding the coordinates $\phi_n^{(c,\eta)}$ and
      $\psi_n^{(c,\eta)}$ of the point $\eta$ with respect to $c$.
$x \leftarrow \frac{\eta - v_0}{\|\eta - v_0\|} \cdot \frac{v_1 - v_0}{\|v_1 - v_0\|}$;   $y \leftarrow \sqrt{1 - x^2}$;   $\omega \leftarrow \frac{\|\eta - v_0\|}{\|v_1 - v_0\|}(x + iy)$;
$B \leftarrow \frac{1}{2\pi\|v_1 - v_0\|^2}$;
$W \leftarrow D(\omega)$ ;                                    // see Eq. (21)
$F \leftarrow |\operatorname{Im}(\omega)| < \epsilon \ ? \ \frac{B}{\|\omega\|^2 - \operatorname{Re}(\omega)} \ : \ \frac{B\operatorname{Im}(W)}{\operatorname{Im}(\omega)}$;
for $n \leftarrow 0$ to $N + 1$ do
    $\boldsymbol{F}[n] \leftarrow F$;
    $F \leftarrow \operatorname{Re}(\omega)F + B\operatorname{Re}(W)$;
    $W \leftarrow \omega W + \frac{1}{n+1}$;
end
$\alpha \leftarrow -(\eta - v_0) \cdot (v_1 - v_0)^{\perp}$;   $\beta \leftarrow -(\eta - v_0) \cdot (v_1 - v_0)$;
$\gamma \leftarrow \|v_1 - v_0\|^2$;   $\delta \leftarrow -\frac{1}{2\pi}\log(\|v_1 - \eta\|)$;
for $n \leftarrow 0$ to $N$ do
    $\boldsymbol{\phi}[n] \leftarrow \alpha\boldsymbol{F}[n]$;
    $\boldsymbol{\psi}[n] \leftarrow \beta\boldsymbol{F}[n] + \gamma\boldsymbol{F}[n + 1] + \delta$;
end
$\boldsymbol{\psi}[0] \leftarrow 0$;
```

4 RESULTS AND COMPARISONS

We present in Fig. 5 various examples of image deformation, comparing our approach to other cage-based deformation methods and showing that it provides smooth and controllable results both for organic and mechanical images. Fig. 3 highlights that enabling a curved deformation of the cage provides a regularity that cannot be matched with straight segment, even when using an equivalent number of control points. Among the two methods that handle curved deformation, Fig. 4 shows that ours is the only one to preserve the local texture details. Our method is also able to go higher in the polynomial degrees of the curved segments (Fig. 1.f and 6).


(a) Rest pose

(b) MVC (c) Cubic MVC (d) Green (e) Cubic Green

Figure 3: Using as many control points, our Cubic Green coordinates (e) bring to GC (d) the same smoothness that Cubic MVC (c) bring to MVC (b).

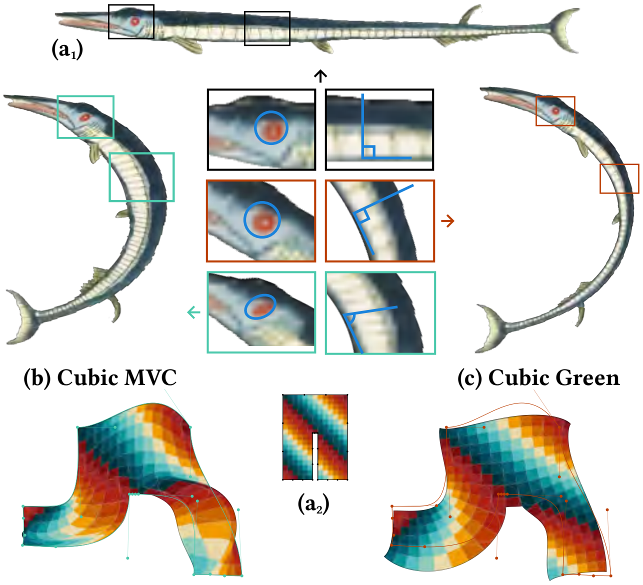


Figure 4: Compared to Cubic MVC (b), our conformal Cubic Green coordinates (c) preserve the local texture aspect of the original image (a_1,a_2).

Comparison with GC. Our technique extends formally Green Coordinates [Lipman et al. 2008] to the case of polynomial curves. While it is *feasible* to build a multiresolution framework by discretizing polynomial curves into segments, and summing up the contributions of the various segments to obtain the final deformation function, this approach has a few shortcomings. First, approaching smooth curves by a collection of straight segments does not lead to similarly-looking corresponding shape deformations (see Fig. 2 and Fig. 3). Second, the artist has to know *in advance* where she desires approximating curvature, as those are the regions that require more sampling of the curves. If *animation* is considered, beyond *static deformation*, this might be rendered impossible in practice without sacrificing quality. Finally, this approach requires storing more coordinates, and a more complex implementation.

Comparison with Cubic MVC. Cubic MVC are strongly related to our work, in many ways. They build atop the Green identity on the unit disk, to diffuse both Dirichlet and Neumann conditions on the unit projection sphere, which is a key ingredient to define MVC and all its variants [Hormann and Floater 2006; Ju et al. 2005; Lipman et al. 2007; Thiery et al. 2018]. Cubic MVC allow interpolating both boundary conditions and defining the deformation function everywhere in space (not just inside the cage). Providing exact interpolation as well as *extrapolation* comes at a cost: it is then impossible to ensure conformal deformations. Contrary to MVC and Cubic MVC, we trade the interpolation property for the conformality one, and ensure that the *local* aspect of details is preserved exactly, which is emphasized in our various results. Similarly, we trade the *extrapolation* property for the *cage-awareness* of the standard Green coordinates, namely we obtain satisfactory results even if limbs are extremely close in the rest-pose (see Fig. 5).

Note that Li and colleagues [2013] show examples of deformations using *curved-cages*; however they rely on a *cage-straightening* step in order to define an intermediate *straight-cage* and intermediate image, which then becomes *in fine* the real input configuration fed to their deformation framework. We focus our work on *straight cages* as well and leave the construction of conformal deformations with input curved-cages for future work.

5 DISCUSSION

Link with Cauchy coordinates. As already stressed, our work can be seen as an extension of classical Green coordinates, which, when tuned for conformality using the condition of Lemma 1, are equivalent to Cauchy coordinates [2009]. As shown by Weber and colleagues, those are simpler to derive than Green coordinates.

Extending Cauchy coordinates instead of Green coordinates could probably be done – resulting in formulas that would be equivalent to ours, and it would probably bring other benefits.

Limitations and Future work. Our work has several limitations.

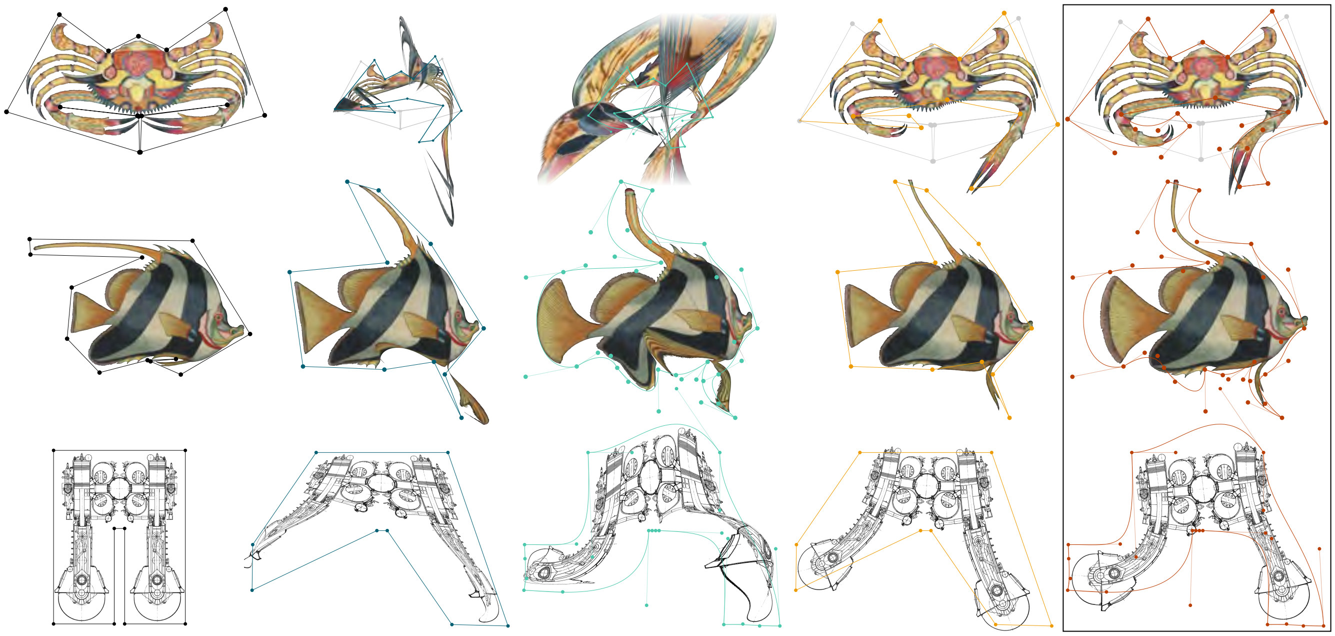
The first one, which is shared by all existing works on 2D cage-based editing, is that we require cages made of straight segments at encoding. This prevents the user from deforming simple shapes with few parameters, if those shapes are not in a straight pose at encoding. For example, while a simple box is used to deform a straight snake, the same "curved box" cannot be used to deform a "curved snake". Extending or work to allow for arbitrarily complex rest-pose curves is our main future work.

Secondly, our approach requires setting by hand all the deformation curves. Designing variational frameworks such as [Weber et al. 2012] to allow artists to deform shapes with very few control points in an interesting avenue for future work. It will require deriving closed-form expressions for the derivatives of our coordinates.

The third limitation is that we require the artist to provide the rest cage. While designing by hand 2D polygonal cages is much less tedious than designing cages for 3D deformations, (semi-)automating this task might render our approach usable to a broader audience.

Finally, as we provide conformal deformations only, the shape deformations do not always follow closely the deformation cage, as conformality and interpolation are conflicting properties. Allowing the user to specify regions where conformality might be relaxed to better follow the deformed cage might be interesting. This might be done by mixing various types of boundary conditions or by damping the Neumann control points $\bar{c}^{\perp}$ in Eq. (14), in the spirit of what Lipman and colleagues proposed in [2008] (Section 3, Fig.7), but this remains to be investigated.

Conclusion. We presented in this work the first method to deform 2D shapes using cages made of straight segments being deformed into polynomial curves of arbitrary order, and the first method to provide conformal deformations using curve guides. Our contribution can be seen as an extension of the popular 2D Green coordinates, which have been the backbone of many subsequent works, to the case of curve-cages. On the practical side, polynomial curves (such as for example Bézier curves) being a key component of all common 2D editing and modeling tools, we believe that our approach will find many practical use cases both in industry and within the general Graphics community.



(a) Rest pose **(b) MVC** **(c) Cubic MVC** **(d) Green** **(e) Cubic Green**

Figure 5: Results of our approach for *cubic* curves (e). We compare with MVC (b), Cubic MVC (c), and usual Green coordinates (d). More results featuring cages made of higher-degree curves are given in Fig. 6.

REFERENCES

Fernando De Goes and Doug L James. 2017. Regularized kelvinlets: sculpting brushes based on fundamental solutions of elasticity. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 1–11.

Zeev Farbman, Gil Hoffer, Yaron Lipman, Daniel Cohen-Or, and Dani Lischinski. 2009. Coordinates for instant image cloning. *ACM Transactions on Graphics (TOG)* 28, 3 (2009), 1–9.

Michael S Floater. 2003. Mean value coordinates. *Computer aided geometric design* 20, 1 (2003), 19–27.

Kai Hormann and Michael S Floater. 2006. Mean value coordinates for arbitrary planar polygons. *ACM Transactions on Graphics (TOG)* 25, 4 (2006), 1424–1441.

Fei Hou, Qian Sun, Zheng Fang, Yong-Jin Liu, Shi-Min Hu, Hong Qin, Aimin Hao, and Ying He. 2017. Poisson vector graphics (pvg) and its closed-form solver. *arXiv preprint arXiv:1701.04303* (2017).

Fei Hou, Qian Sun, Zheng Fang, Yong-Jin Liu, Shi-Min Hu, Hong Qin, Aimin Hao, and Ying He. 2018. Poisson vector graphics (PVG). *IEEE transactions on visualization and computer graphics* 26, 2 (2018), 1361–1371.

Jin Huang, Xiaohan Shi, Xinguo Liu, Kun Zhou, Li-Yi Wei, Shang-Hua Teng, Hujun Bao, Baining Guo, and Heung-Yeung Shum. 2006. Subspace gradient domain mesh deformation. In *ACM SIGGRAPH 2006 Papers*. 1126–1134.

Peter Ilbery, Luke Kendall, Cyril Concolato, and Michael McCosker. 2013. Biharmonic diffusion curve images from boundary elements. *ACM Transactions on Graphics (TOG)* 32, 6 (2013), 1–12.

Pushkar Joshi, Mark Meyer, Tony DeRose, Brian Green, and Tom Sanocki. 2007. Harmonic coordinates for character articulation. *ACM Transactions on Graphics (TOG)* 26, 3 (2007), 71–es.

Tao Ju, Scott Schaefer, and Joe Warren. 2005. Mean value coordinates for closed triangular meshes. In *ACM Siggraph 2005 Papers*. 561–566.

Xian-Ying Li, Tao Ju, and Shi-Min Hu. 2013. Cubic Mean Value Coordinates. *ACM Transactions on Graphics* 32, 4 (2013), 126:1–10.

Yaron Lipman, Johannes Kopf, Daniel Cohen-Or, and David Levin. 2007. GPU-assisted positive mean value coordinates for mesh deformations. In *Symposium on geometry processing*.

Yaron Lipman, David Levin, and Daniel Cohen-Or. 2008. Green coordinates. *ACM ToG* 27, 3 (2008), 1–10.

Scott Schaefer, Travis McPhail, and Joe Warren. 2006. Image deformation using moving least squares. In *ACM SIGGRAPH 2006 Papers*. 533–540.

Thomas W Sederberg and Scott R Parry. 1986. Free-form deformation of solid geometric models. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*. 151–160.

Xin Sun, Guofu Xie, Yue Dong, Stephen Lin, Weiwei Xu, Wencheng Wang, Xin Tong, and Baining Guo. 2012. Diffusion curve textures for resolution independent texture mapping. *ACM Transactions on Graphics (TOG)* 31, 4 (2012), 1–9.

Jean-Marc Thiery, Pooran Memari, and Tamy Boubekeur. 2018. Mean value coordinates for quad cages in 3D. *ACM Transactions on Graphics (TOG)* 37, 6 (2018), 1–14.

Ofir Weber, Mirela Ben-Chen, Craig Gotsman, et al. 2009. Complex barycentric coordinates with applications to planar shape deformation. In *Computer Graphics Forum*, Vol. 28. Citeseer, 587.

Ofir Weber, Mirela Ben-Chen, Craig Gotsman, and Kai Hormann. 2011. A complex view of barycentric mappings. In *Computer Graphics Forum*, Vol. 30. Wiley Online Library, 1533–1542.

Ofir Weber, Roi Poranne, and Craig Gotsman. 2012. Biharmonic coordinates. In *Computer Graphics Forum*, Vol. 31. Wiley Online Library, 2409–2422.

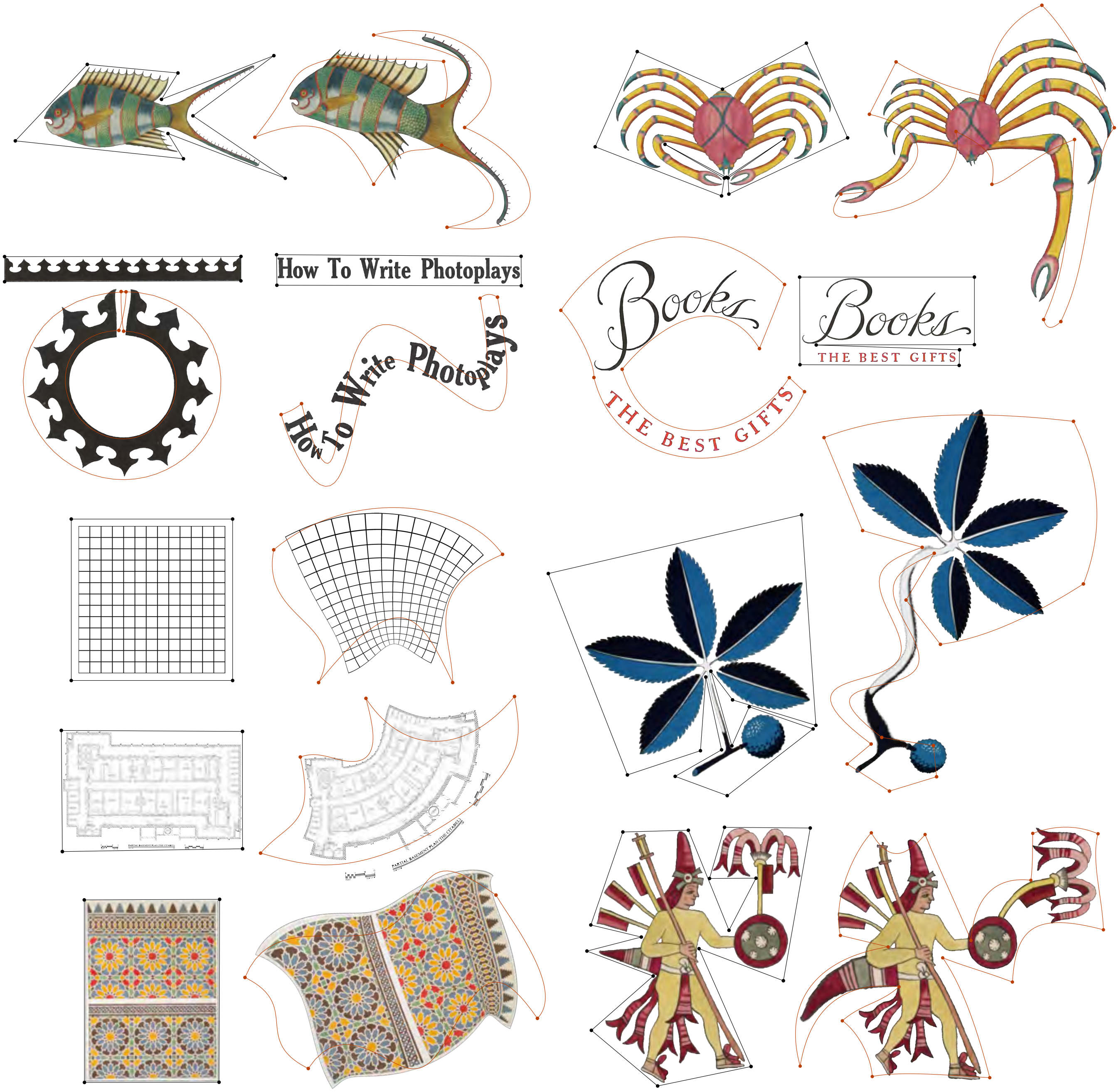


Figure 6: Extra results of our approach, using polynomial curves of various degrees.