C^0 Generalized Coons Patches for High-order Cage-based Deformation

KAIKAI QIN, Hangzhou Dianzi University, China
YUNHAO ZHOU, Hangzhou Dianzi University, China
CHENHAO YING, Hangzhou Dianzi University, China
YAJUAN LI, Hangzhou Dianzi University, China
CHONGYANG DENG*, Hangzhou Dianzi University, China

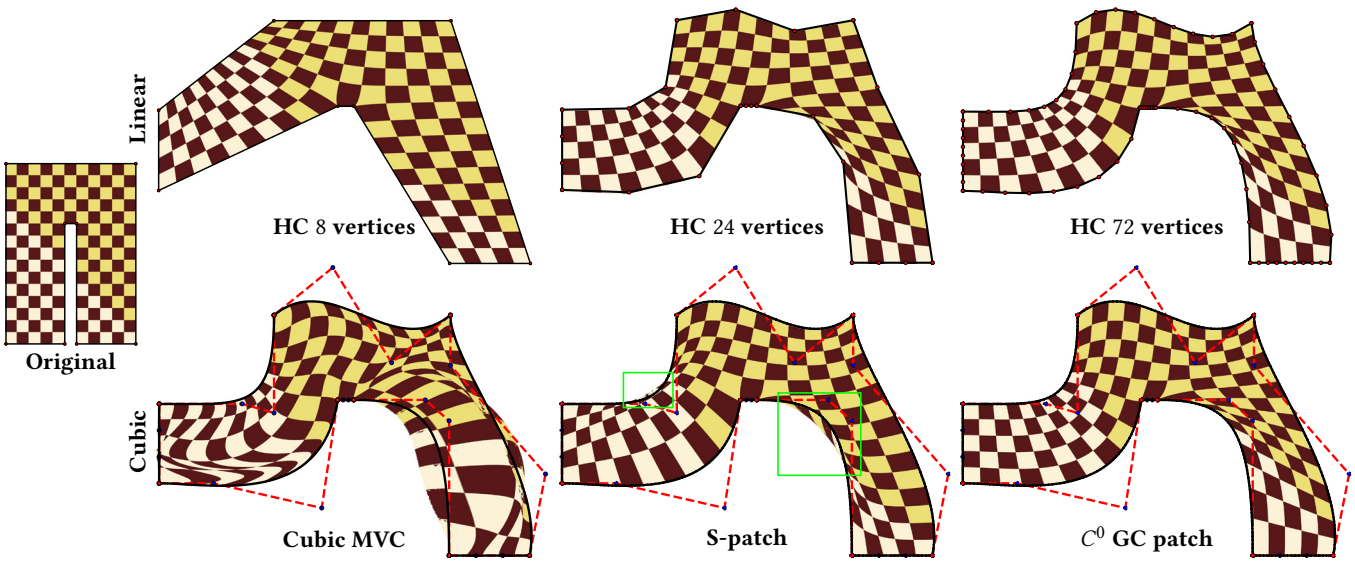

Fig. 1. Deforming the original pant-shape polygon using linear (top row) and high-order (bottom row) cage-based deformation, respectively. Top row from left to right: the results obtained using linear cages of different resolutions, containing 8, 24, and 72 vertices, respectively. Bottom row from left to right: the results obtained using the same cubic cage with different methods, including cubic MVC [Li et al. 2013], selective degree elevation of the S-patch [Smith and Schaefer 2015], and the C^0 Generalized Coons patch. Harmonic coordinates [Joshi et al. 2007] were used for the linear cage-based deformation and the patch evaluation (both the S-patch and the C^0 Generalized Coons patch).

Space deformations deform the ambient space and thus implicitly deform the embedded objects. Free-Form Deformation allows high-order deformation to the embedding space, yet the lattice may fail to conform to the object and involves many internal control points. Cage-based Deformation utilizes a cage space that conforms to the object, obviating the need for additional internal control points, but it is typically linear at edges. In this paper, we propose a simple and general method with both advantages while avoiding their drawbacks, allowing users to implement high-order cage-based deformation. To achieve this goal, we introduce a new parametric transfinite interpolation scheme based on generalized barycentric coordinates, which unifies and generalizes the rectangular and triangular Coons patch. This C^0 Generalized Coons patch can be defined not only over 2D domains but also 3D domains or even higher-dimensional domains, with arbitrary polytopes, even including non-manifold topologies. Moreover, the C^0 Generalized Coons patch has an elegant mathematical expression.

CCS Concepts: • **Computing methodologies → Parametric curve and surface models**; **Volumetric models**; **Shape analysis**.

Additional Key Words and Phrases: Cage-based deformation, free-form curves, generalized barycentric coordinates, transfinite interpolation, Coons patches

*Corresponding author.

Authors' addresses: Kaikai Qin, Hangzhou Dianzi University, Hangzhou, China, qinkaikai@hdu.edu.cn; Yunhao Zhou, Hangzhou Dianzi University, Hangzhou, China, 221070003@hdu.edu.cn; Chenhao Ying, Hangzhou Dianzi University, Hangzhou, China, 221070020@hdu.edu.cn; Yajuan Li, Hangzhou Dianzi University, Hangzhou, China, liyajuan@hdu.edu.cn; Chongyang Deng, Hangzhou Dianzi University, Hangzhou, China, dcy@hdu.edu.cn.

ACM 0730-0301/2024/12-ART220
https://doi.org/10.1145/3687972

ACM Reference Format:
Kaikai Qin, Yunhao Zhou, Chenhao Ying, Yajuan Li, and Chongyang Deng. 2024. C^0 Generalized Coons Patches for High-order Cage-based Deformation. *ACM Trans. Graph.* 43, 6, Article 220 (December 2024), 15 pages. https://doi.org/10.1145/3687972

Table 1. Comparison with existing high-order cage-based deformation methods.

	2D Case	Higher Dimensional Cases	Non-manifold Topologies	Cage Type
Cubic MVC [Li et al. 2013]	✔	✗	✗	Only Cubic curves
Polynomial Green coordinates [1] [Michel and Thiery 2023]	✔	✗	✗	Only Bézier curves
Selective Degree Elevation [Smith and Schaefer 2015]	✔	✔	✗	Only Bézier curves
C^0 GC patches	✔	✔	✔	**Arbitrary Free-Form curves**

[1] Polynomial Green coordinates are entirely distinct from other methods in that they lack the interpolation property but have the conformality property.

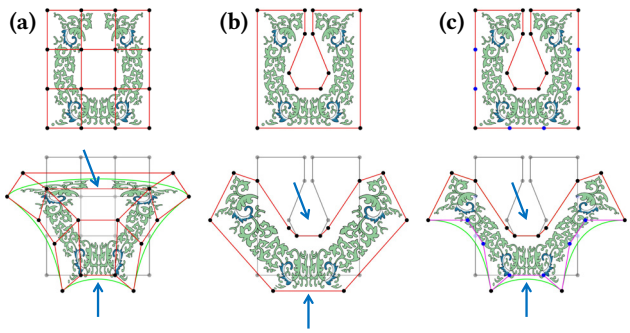

Fig. 2. Image deformation using different methods. **(a)** Free-Form Deformation. **(b)** Cage-based Deformation. **(c)** High-order Cage-based Deformation. Top row: the original object with initial lattice/cage. Bottom row: the deformation results.

1 INTRODUCTION

Deformation, an important technique in computer graphics, widely used for geometric modeling and character animation, involves transforming or mapping the position of each point in the original object to the corresponding position in the deformed object [Botsch et al. 2006]. *Free-Form Deformation* (FFD) [Sederberg and Parry 1986] is a user-friendly technique for deforming geometric models by warping the whole embedding space. The embedding space is usually a parallelepiped represented as a *polynomial* tensor-product Bézier/B-spline volume with a regular control lattice. However, for complex models, the parallelepiped embedding space may not fit them well, which could result in alias artifacts in the deformed models [Botsch et al. 2010]. Besides, the lattices usually include internal control points that are challenging and frustrating to articulate. *Cage-based Deformation* (CBD) [Ju et al. 2005] is a generalization of the lattice-based FFD. Unlike FFD, CBD uses a coarse control mesh that better conforms to the undeformed object. The embedding polyhedral space is represented using *generalized barycentric coordinates* (GBC) [Floater 2015]. Although generalized barycentric interpolation avoids the tedious internal control points, it remains linear (non-smooth) at the boundaries. This results in two issues: firstly, when users need additional degrees of freedom, cage refinement [Hormann and Floater 2006] is necessary, which requires recalculating the cage coordinates; secondly, cage refinement maintains piecewise linear functions on the boundary [Schaefer 2017; Smith and Schaefer 2015]. This work will combine the advantages of both methods while avoiding their drawbacks, allowing users to achieve high-order cage-based deformation. (see Fig. 2).

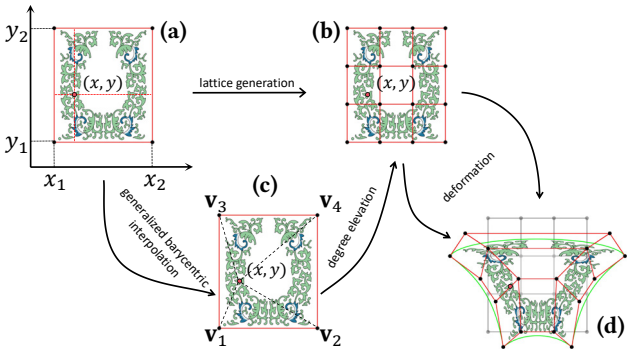

Fig. 3. 2D deformation. FFD process: **(a)⇒(b)⇒(d)**. Degree elevated CBD process: **(a)⇒(c)⇒(b)⇒(d)**.

A generalized barycentric view of FFD

Here, we present the potential connection between FFD and CBD, which seems to have first appeared in [Langer et al. 2008], then was revealed by [Smith and Schaefer 2015], and later by [Schaefer 2017].

For simplicity, let us consider the 2D case. Fig. 3 illustrates the process of 2D deformation. The undeformed object was initially enclosed within a rectangle $\Omega = \{(x, y) \mid x \in [x_1, x_2],\ y \in [y_1, y_2]\}$.

For any point (x, y) inside the rectangle, we calculate its local coordinates (u, v) and the regular control lattice $\{\mathbf{b}_{i,j}\}_{i,j=0,0}^{d_1,d_2}$ (see Fig. 3**(a)-(b)**).

$$(u, v) = \left(\frac{x - x_1}{x_2 - x_1}, \frac{y - y_1}{y_2 - y_1}\right), \tag{1}$$

$$\mathbf{b}_{i,j} = (x_1, y_1) + \left(\frac{i}{d_1}(x_2 - x_1), \frac{j}{d_2}(y_2 - y_1)\right). \tag{2}$$

Thus, the embedding space Ω is represented as

$$(x, y) = \mathbf{S}(u, v) = \sum_{i=0}^{d_1}\sum_{j=0}^{d_2} B_i^{d_1}(u) B_j^{d_2}(v) \cdot \mathbf{b}_{i,j}, \tag{3}$$

where $B_k^d(t) = \frac{d!}{(d-k)!k!}(1-t)^{d-k}t^k$ is the k-th Bernstein basis function of degree d. By moving the control points $\{\mathbf{b}_{i,j}\}_{i,j=0,0}^{d_1,d_2}$, the embedding surface $\mathbf{S}(u, v)$ is deformed, thereby deforming the object as well (see Fig. 3**(d)**).

Indeed, the 2D FFD process described above could be divided into the following two steps.

I. *Generalized barycentric interpolation.* Let $\{\mathbf{v}_k\}_{k=1}^4$ be the vertices of rectangle Ω, then any point (x, y) inside the rectangle is expressed as

$$(x, y) = \sum_{k=1}^{4} \lambda_k \cdot \mathbf{v}_k, \tag{4}$$

where

$$\begin{aligned} \lambda_1 &= (1-u)(1-v), \\ \lambda_2 &= u(1-v), \\ \lambda_3 &= uv, \\ \lambda_4 &= (1-u)v, \end{aligned} \tag{5}$$

are the inverse bilinear coordinates [Floater 2015] with respect to $\{\mathbf{v}_k\}_{k=1}^4$ (see Fig. 3**(c)**).

II. *Degree elevation.* The above generalized barycentric interpolation is exactly a bi-1 tensor-product Bézier surface. Then applying the degree elevation formula of Bézier surfaces to Equation (4) will reproduce Equation (3) (see Fig. 3**(b)**).

From this perspective, the 2D FFD could be interpreted as a degree-elevated 4-sided CBD [Schaefer 2017].

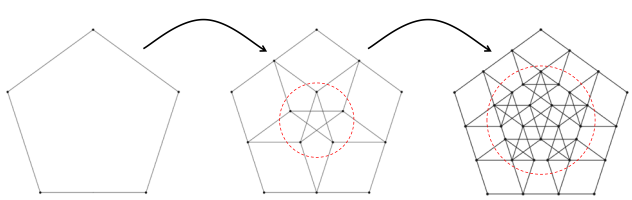

Fig. 4. Degree elevation of a 5-sided S-patch. Left: the linear 5-sided Bézier grid has 5 control points, including 0 internal control point. Middle: the quadratic 5-sided Bézier grid has 15 control points, including 5 internal control points. Right: the cubic 5-sided Bézier grid has 35 control points, including 20 internal control points.

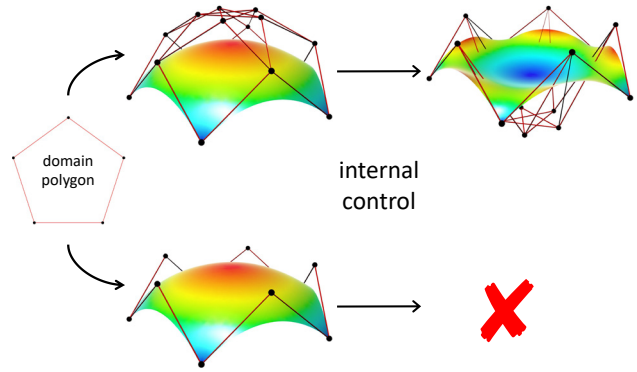

Fig. 5. Control point-based (top) vs. Transfinite interpolation (bottom). By moving the internal control points, we can edit the interior of the control point-based surface without changing the boundaries, which is impossible with the transfinite interpolation surface.

Motivation

The degree elevation can be applied to a polygonal cage to increase its degrees of freedom, which leads to the well-known S-patch [Loop and DeRose 1989]. Although the S-patch has a very elegant mathematical form, it has many internal control points and a complex control mesh structure (see Fig. 4). Since the cage already conforms to the object, we do not need internal control points. Just as Schaefer wrote [Schaefer 2017], "*Moreover, the majority of those control points exist in the interior of the cage, which would make it difficult for a user to manipulate those points. Hence, higher degree cage-based deformations are not practical from a user-interface perspective.*" Therefore, directly using S-patch for high-order CBD is not a suitable choice.

In the field of Computer Aided Geometric Design (CAGD), there exists another method different from control-point-based schemes like the Bézier approach, known as transfinite interpolation, such as the Coons patch [Coons 1967]. A significant drawback of transfinite interpolation is the lack of internal control (see Fig. 5). Here, we quote [Várady et al. 2016], "*The transfinite approach is helpful when we are satisfied with the shape by the automatic settings, however, difficulties may arise if the shape needs to be further modified or optimized in the interior.*" Over the past decade, the CAGD community has been dedicated to adding internal controls to multi-sided parametric transfinite interpolation surfaces [Salvi 2024; Várady et al. 2012; Várady et al. 2016].

The lack of internal control in transfinite interpolation schemes is required for CBD. Thus, parametric transfinite interpolation would be a perfect choice for high-order CBD. To apply to CBD, parametric transfinite interpolation surfaces must meet the following requirements.

- *Arbitrary topology.* As previously stated, cages depend on the model's shape and the user's specifications, generally without topological constraints. Therefore, the parametric transfinite interpolation scheme used for CBD must be able to represent surfaces of arbitrary topology.
- *Linear reproduction.* This implies that the parametric transfinite interpolation surface on the original cage space (linear

cage) needs to be an identity transformation. Thus, the parameterization issue of the original cage space, which is undoubtedly very challenging if the linear reproduction property is not present, can be avoided.

However, existing parametric transfinite interpolation surfaces (cf. Section 2.4) cannot meet the above requirements simultaneously. Almost all patches were developed to address the hole-filling issue in surface modeling, for which the continuity between surfaces is crucial. Therefore, transfinite interpolation surfaces must account for the interpolation of cross-boundary derivatives and corner twists. The constraints of high-order interpolation make it difficult for transfinite interpolation schemes to meet the above two requirements. However, for CBD, there is no need to consider smooth joining between adjacent patches, so C^0 interpolation is sufficient. Recalling S-patches, they can meet the two requirements because they use GBCs as parameters [Schaefer 2017]. The original S-patches [Loop and DeRose 1989] were only defined over convex polygons because they used Wachspress coordinates [Wachspress 1975]. However, using more general barycentric coordinates [Langer et al. 2008; Smith and Schaefer 2015] could lift the topological restrictions. Linear S-patches essentially reproduce generalized barycentric interpolation (so we named it ***generalized barycentric reproduction***), and GBCs satisfy the property of linear reproduction. Therefore, similar to S-patches, it is possible for multi-sided parametric transfinite interpolation surfaces to indirectly meet the linear reproduction property (i.e., satisfy generalized barycentric reproduction) by using GBC-based parameterization.

Contributions

In this paper, we first introduced a new multi-sided parametric transfinite interpolation scheme based on GBC-based parameterization [Várady et al. 2024], called the C^0 Generalized Coons (GC) patch, which is a natural generalization of the original Coons patch [Coons 1967] and also the triangular Coons patch [Barnhill 1977; Marshall 1975; Nielson 1979] and could be regarded as a general C^0 case of the Charrot–Gregory patch [Charrot and Gregory 1984] and the ribbon-based C^1 GC patch [Salvi et al. 2014]. The C^0 GC patch has an elegant expression. It defines over 2D domains and 3D domains or even higher-dimensional domains with arbitrary polytopes, including non-manifold topologies, as long as the used GBCs are well-defined. More importantly, it satisfies the property of generalized barycentric reproduction. Then, we apply the proposed C^0 GC patches to higher-order cage-based deformation. It allows for smooth deformation from straight segments to arbitrary free-form curves by introducing additional control points. This results in improved performance in scenarios such as bending or twisting.

Outline

The remainder of this paper is organized as follows. A brief review of related work is presented in Section 2. Then we define the generalized Coons patch in Section 3 and propose the high-order deformation method in Section 4. Some examples are shown in Section 5. We discuss the limitations in Section 6 and conclude this paper in Section 7.

2 RELATED WORK

2.1 Free-Form Deformation

Sederberg and Parry first introduced the FFD technique [Sederberg and Parry 1986] and originally utilized tensor-product Bézier volumes. Subsequently, various extended FFD techniques were proposed, incorporating B-spline [Griessmair and Purgathofer 1989] and NURBS [Lamousin and Waggenspack Jr. 1994]. Recently, THP-spline [Reis and Kosinka 2018] and T-spline [Zhang et al. 2020] methods have also been applied to FFD. The EFFD method [Coquillart 1990] was proposed to remove the restriction of parallel structure on control lattices, enabling support for more general shapes. However, using non-parallel control lattices may lead to difficulty parameterizing the deformation space, requiring solving complex nonlinear systems. Catmull–Clark volumetric subdivision [MacCracken and Joy 1996] is introduced into the FFD framework to support control lattices with arbitrary topologies. Nonetheless, the subdivision-based FFD also faces the parameterization issue of the deformation space and is computationally costly and memory-intensive.

2.2 Cage-based Deformation

CBD was initially introduced by [Ju et al. 2005], where the embedding space is a polyhedron with arbitrary topology (i.e., the cage), which conforms well to the object and is represented using generalized barycentric coordinates. Since generalized barycentric coordinates [Floater 2015] possess linear reproduction property, the generalized barycentric interpolation is an identity mapping on the initial cage space, avoiding reparameterization. Recently, [Ströter et al. 2024] have presented a good survey on CBD. In the past two decades, research on CBD has primarily focused on the construction of various generalized barycentric coordinates, such as man value coordinates (MVC) [Floater et al. 2005; Ju et al. 2005], harmonic coordinates (HC) [Joshi et al. 2007], Green coordinates [Lipman et al. 2008], local barycentric coordinates [Zhang et al. 2014], and others [Budninskiy et al. 2016; Chang et al. 2023; Deng et al. 2020; Dodik et al. 2023; Thiery et al. 2018].

2.3 Deformation using high-order cages

[Li et al. 2013] first introduced the concept of curved cages and proposed the cubic MVC, an extension of MVC. The cubic MVCs allow users to deform a linear cage into a curved cage with cubic boundaries. However, they inherit the drawback of MVC being negative over concave polygons, leading to artifacts in deformations (see Fig. 1). Recently, [Michel and Thiery 2023] proposed polynomial Green Coordinates, an extension of Green coordinates, allowing deformations from linear edges to polynomial edges. Nevertheless, both coordinates are defined only in 2D.

The selective degree elevation algorithm for S-patches proposed by [Smith and Schaefer 2015] is most relevant to our method. They introduced an indirect selective degree elevation method by eliminating the weight deficiency of the basis functions to avoid inserting more control points than desired. Their method can transform linear boundaries into Bézier curve boundaries. However, deforming solely using these selectively inserted control points may result in artifacts (see Fig. 1 and 15). Furthermore, their method cannot be

applied to non-manifold topologies. Table 1 presents a comparison between different approaches.

2.4 Polygonal parametric transfinite interpolation surfaces

Transfinite interpolation surfaces originated from the Coons patch [Coons 1967] formulated in the 1960s, which stands as one of the earliest parametric surface modeling methods in CAGD. The original Coons patch only interpolates the given four boundary curves (C^0), later extended to C^1 [Gregory 1974] and triangular [Barnhill 1977; Marshall 1975; Nielson 1979] interpolation.

Subsequently, transfinite surface interpolation was extended to multi-sided patches. [Charrot and Gregory 1984] first utilized a regular pentagon as the parametric domain to construct a pentagonal G^1 interpolation surface, i.e., the Charrot–Gregory patch, which was later extended to n-sided [Gregory 1986] and G^2 interpolation [Gregory and Hahn 1989]. This construction first defines n corner-subpatches, which are *quad-patches* then blended together using blending functions. This subpatch-based construction (also known as *ribbon-based* construction [Várady et al. 2024]) has influenced nearly all subsequent multi-sided transfinite interpolation surfaces, such as Kato's patch [Kato 1991], C^1 GC patch and composite ribbon patch [Salvi et al. 2014], and others [Qin et al. 2023; Salvi et al. 2023; Várady et al. 2012; Várady et al. 2011]. Although these methods achieve high-order interpolation, they do not meet the two previously mentioned requirements, making them unsuitable for space deformation.

Recently, Salvi [Salvi 2020] introduced a multi-sided generalization of the C^0 Coons patch. Although their work is quite similar to ours, both achieving C^0 interpolation and using GBC-based parameterization, their method continues to rely on subpatch or ribbon-based constructions [Salvi et al. 2014]. It does not satisfy the generalized barycentric reproduction property. In contrast, our method does not require constructing subpatches but instead directly combines the boundary curves and satisfies the property of generalized barycentric reproduction (see Fig. 6).

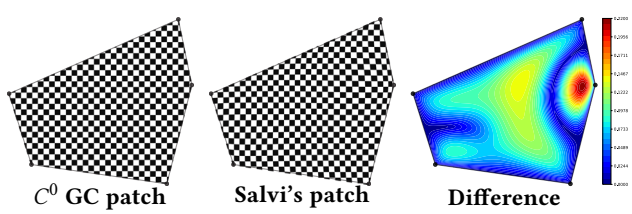

Fig. 6. Comparison with Salvi's [Salvi 2020] patch. When the input boundary configuration matches the domain polygon, the C^0 GC patch can reproduce the parameter space, whereas Salvi's method cannot. Here, we used 2-norm to measure the difference between our and Salvi's patches.

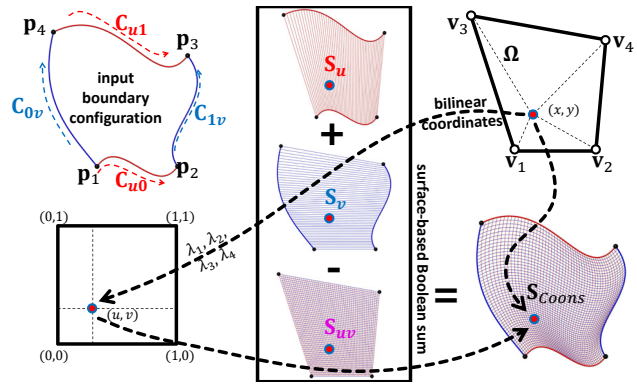

Fig. 7. Surface-based Boolean sum construction of a Coons patch.

3 A GENERALIZATION OF COONS PATCH

3.1 The original C^0 Coons patch

Given 4 boundary curves $\{\mathbf{C}_{u0}(t), \mathbf{C}_{u1}(t), \mathbf{C}_{0v}(t), \mathbf{C}_{1v}(t) \mid t \in [0,1]\}$, they form a loop, that is

$$\begin{aligned}\mathbf{C}_{u0}(0) &= \mathbf{C}_{0v}(0) = \mathbf{p}_1,\\ \mathbf{C}_{u0}(1) &= \mathbf{C}_{1v}(0) = \mathbf{p}_2,\\ \mathbf{C}_{u1}(1) &= \mathbf{C}_{1v}(1) = \mathbf{p}_3,\\ \mathbf{C}_{u1}(0) &= \mathbf{C}_{0v}(1) = \mathbf{p}_4.\end{aligned}$$

Then the Coons patch (see Fig. 7) which interpolates the 4 boundary curves is

$$\mathbf{S}_{Coons}(u,v) = \mathbf{S}_u + \mathbf{S}_v - \mathbf{S}_{uv}, \tag{6}$$

where

$$\begin{aligned}\mathbf{S}_u &= (1-v)\cdot \mathbf{C}_{u0}(u) + v\cdot \mathbf{C}_{u1}(u),\\ \mathbf{S}_v &= (1-u)\cdot \mathbf{C}_{0v}(v) + u\cdot \mathbf{C}_{1v}(v),\\ \mathbf{S}_{uv} &= (1-u)(1-v)\cdot \mathbf{p}_1 + u(1-v)\cdot \mathbf{p}_2\\ &\quad + uv\cdot \mathbf{p}_3 + (1-u)v\cdot \mathbf{p}_4,\end{aligned} \tag{7}$$

and $(u,v) \in [0,1]^2$.

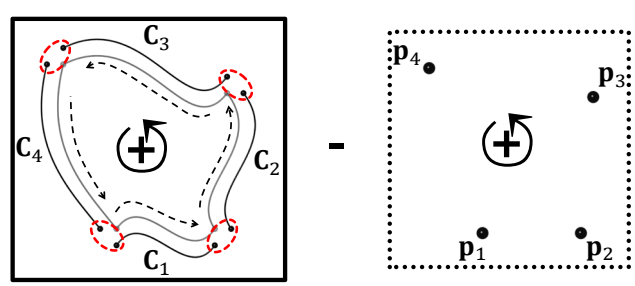

Fig. 8. Curve-based Boolean sum construction of a Coons patch.

3.2 Reparameterization using GBC-based paremeters

Similar to tensor-product Bézier surfaces, we can express the Coons patch in terms of generalized barycentric coordinates. Let Ω be a planar convex quadrilateral with vertices $\{\mathbf{v}_k\}_{k=1}^4$. For each point

$(x, y) \in \Omega$, there exists a unique (u, v) that determines the bilinear coordinates $\{\lambda_k\}_{k=1}^4$ by Equation (5) which satisfy Equation (4). Then we have

$$\begin{aligned} 1 - v = \lambda_1 + \lambda_2, \qquad & u = \frac{\lambda_2}{\lambda_1 + \lambda_2}, \\ u = \lambda_2 + \lambda_3, \qquad & v = \frac{\lambda_3}{\lambda_2 + \lambda_3}, \\ v = \lambda_3 + \lambda_4, \quad & 1 - u = \frac{\lambda_4}{\lambda_3 + \lambda_4}, \\ 1 - u = \lambda_4 + \lambda_1, \quad & 1 - v = \frac{\lambda_1}{\lambda_4 + \lambda_1}. \end{aligned} \tag{8}$$

Assuming

$$\begin{aligned} \mathbf{C}_1(t) &= \mathbf{C}_{\boldsymbol{u0}}(t), \\ \mathbf{C}_2(t) &= \mathbf{C}_{\boldsymbol{1v}}(t), \\ \mathbf{C}_3(t) &= \mathbf{C}_{\boldsymbol{u1}}(1 - t), \\ \mathbf{C}_4(t) &= \mathbf{C}_{\boldsymbol{0v}}(1 - t), \end{aligned} \tag{9}$$

so all the boundary curves are in a counterclockwise direction (see Fig. 8). Substituting Equations (8) and (9) into Equations (6) and (7), we obtain

$$\mathbf{S}_{Coons}(x, y) = \sum_{k=1}^{4} (\lambda_k + \lambda_{k+1}) \cdot \mathbf{C}_k \left(\frac{\lambda_{k+1}}{\lambda_k + \lambda_{k+1}} \right) - \sum_{k=1}^{4} \lambda_k \cdot \mathbf{p}_k. \tag{10}$$

Remark 1. *The GBC-based parameterization $\frac{\lambda_{k+1}}{\lambda_k+\lambda_{k+1}}$ could be regarded as a projection from a point inside Ω onto the k-th edge $\mathbf{e}_k = \{(1 - t) \cdot \boldsymbol{v}_k + t \cdot \boldsymbol{v}_{k+1} \mid t \in [0, 1]\}$. This parameterization originated from the sweeping line parameterization of the Charrot–Gregory patch [Charrot and Gregory 1984] and was later derived by Várady et al. [Várady et al. 2016, 2024] using generalized barycentric coordinates. The value of parameter $\frac{\lambda_{k+1}}{\lambda_k+\lambda_{k+1}}$ must be constrained within the range $[0, 1]$ to ensure its validity. Therefore, the generalized barycentric coordinates defining the patch must satisfy* ***non-negativity****.*

The original Coons patch is constructed by surface-based Boolean sum and the new expression by curve-based Boolean sum. An intuitive explanation is that each corner point lies on two curves, so it is calculated twice and needs to be subtracted once (see Fig. 8). The original Coons patch uses bilinear coordinates, so it only defines over planar convex quadrilaterals, while the new expression is independent of generalized barycentric coordinates. Thus, it can be defined over planar concave quadrilaterals. Fig. 9 shows a case of concave configuration of boundary curves. In this scenario, the original Coons patch exhibits fold-over, whereas the new expression avoids fold-over by utilizing mean value coordinates that are well-defined over concave quadrilaterals.

3.3 Multi-sided generalization

The new expression (Equation (10)) of the Coons patch could be directly generalized to multi-sided cases. Give n boundary curves $\{\mathbf{C}_k(t) \mid t \in [0, 1]\}_{k=1}^n$, they form a loop, that is $\mathbf{C}_k(0) = \mathbf{C}_{k-1}(1) =$

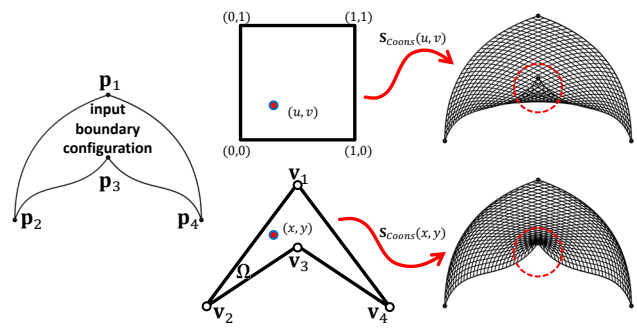

Fig. 9. The original Coons patch $\mathbf{S}_{Coons}(u, v)$ vs. The new expression $\mathbf{S}_{Coons}(x, y)$. From left to right: the input concave configuration, the parameter domains, and the generated Coons patches.

$\mathbf{p}_k$.[1] Let Ω be an n-sided planar polygon with vertices $\{\mathbf{v}_k\}_{k=1}^n$. Then the C^0 generalized Coons patch that defines over Ω and interpolates all prescribed boundary curves is

$$\mathbf{S}_{GC}(x, y) = \sum_{k=1}^{n} (\lambda_k + \lambda_{k+1}) \cdot \mathbf{C}_k \left(\frac{\lambda_{k+1}}{\lambda_k + \lambda_{k+1}} \right) - \sum_{k=1}^{n} \lambda_k \cdot \mathbf{p}_k, \tag{11}$$

where $(x, y) \in \Omega$ and $\{\lambda_k\}_{k=1}^n$ are the generalized barycentric coordinates of (x, y) with respect to the n vertices.

Remark 2. *The internal continuity of a C^0 GC patch depends on the continuity of the generalized barycentric coordinates used and the continuity of the prescribed boundary curves. For instance, if the generalized barycentric coordinates used are C^p continuous within the domain polygon and the boundary curves are C^q continuous, then the resulting C^0 GC patch has a $C^{min\{p,q\}}$ continuity in the interior.*

Theorem 3.1. *[Generalized barycentric reproduction] When the input boundary configuration is exactly the parametric domain polygon Ω, that is*

$$\mathbf{C}_k(t) = (1 - t) \cdot \mathbf{v}_k + t \cdot \mathbf{v}_{k+1}, \quad k = 1, 2, \ldots, n, \tag{12}$$

then the C^0 GC patch will reproduce the generalized barycentric interpolation.

Proof. Substituting Equation (12) into Equation (11), we obtain

$$\begin{aligned} \mathbf{S}_{GC}(x, y) &= \sum_{k=1}^{n} \left\{ (\lambda_k + \lambda_{k+1}) \cdot \left[\left(1 - \frac{\lambda_{k+1}}{\lambda_k + \lambda_{k+1}} \right) \cdot \mathbf{v}_k \right. \right. \\ &\qquad \left. \left. + \frac{\lambda_{k+1}}{\lambda_k + \lambda_{k+1}} \cdot \mathbf{v}_{k+1} \right] \right\} - \sum_{k=1}^{n} \lambda_k \cdot \mathbf{v}_k \\ &= \sum_{k=1}^{n} (\lambda_k \cdot \mathbf{v}_k + \lambda_{k+1} \cdot \mathbf{v}_{k+1}) - \sum_{k=1}^{n} \lambda_k \cdot \mathbf{v}_k \\ &= \sum_{k=1}^{n} \lambda_k \cdot \mathbf{v}_k. \end{aligned}$$

□

[1]Here, we use the cyclic index and hence identify the index $n + 1$ with 1 and the index 0 with n.

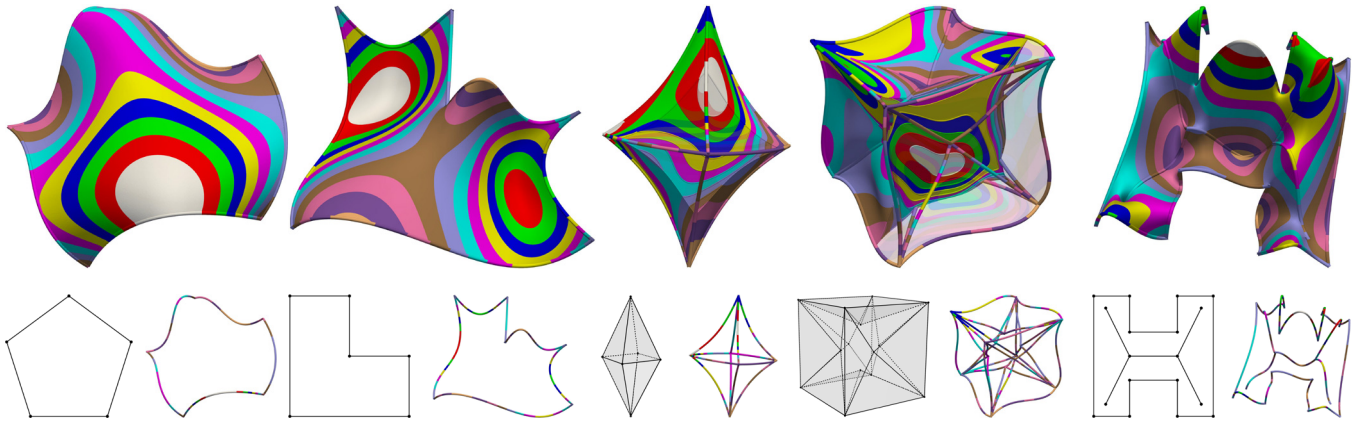

Fig. 10. C^0 GC patches defined over various domains. Top row: the C^0 GC patches that interpolated given boundary curves. Bottom row: corresponding domains and input boundary configurations. From left to right: convex polygonal; concave polygonal; convex polyhedral; concave polyhedral; and non-manifold. HCs were used for the patch evaluation.

In addition, for the case of triangles, the C^0 GC patch will reproduce the triangular Coons patch [Barnhill 1977; Marshall 1975; Nielson 1979]. Therefore, the C^0 GC patch unifies and generalizes both the Coons and the triangular Coons patch, much like the S-patch unifies and generalizes the tensor-product and triangular Bézier patch.

3.4 Link with previous work

The C^0 GC patch (Equation (11)) has strong connections with previous work, particularly with the Charrot–Gregory patch [Charrot and Gregory 1984] and the ribbon-based C^1 GC patch [Salvi et al. 2014]. The C^0 GC patch is a general C^0 case of both, where the subpatches or ribbons must degenerate into curves.

Link with Charrot–Gregory patch. For Equation (11), if we use a regular polygon as the parametric domain and apply Wachspress coordinates as the parameters, the resulting surface will be equivalent to the C^0 version of the Charrot–Gregory patch.

Link with the ribbon-based C^1 GC patch. This patch employs a surface-based Boolean sum construction with constrained parameterization. However, for C^0 interpolation, the cross-boundary derivatives will be removed, and the surface-based Boolean sum construction will degenerate into a curve-based Boolean sum. Then, Equation (11) will be reproduced by replacing the constrained parameterization with GBC-based parameterization in the survey [Várady et al. 2024].

3.5 Higher Dimensions and Non-manifold Topologies

The C^0 GC patch can be easily extended to higher dimensions. To the best of our knowledge, there is very little work concerning interpolation over polytopes in higher dimensions, such as [Randrianarivony 2011] (G^0 interpolation over convex polytopes) and [Qin et al. 2024] (G^2 interpolation over simple convex polyhedra).

Given n_e boundary curves $\{\mathbf{C}_{i,j}(t) \mid t \in [0,1]\}$ in $\mathbb{R}^{d+m}$ with $m \geq 0$, they form a curved polyhedral skeleton $\mathcal{P}$ with n corner points $\{\mathbf{p}_k\}_{k=1}^n$ and satisfy

$$\mathbf{C}_{i,j}(0) = \mathbf{p}_i, \quad \mathbf{C}_{i,j}(1) = \mathbf{p}_j.$$

Φ is a polytope in $\mathbb{R}^d$ with n vertices $\{\mathbf{v}_k\}_{k=1}^n$ and n_e edges $\mathbf{E} = \{\mathbf{e}_{i,j} = (1-t)\cdot\mathbf{v}_i + t\cdot\mathbf{v}_j \mid t \in [0,1]\}$, which is homeomorphic to $\mathcal{P}$. Then, the C^0 GC patch that defines over Φ and interpolates all prescribed boundary curves is

$$\mathbf{S}_{GC}(\mathbf{x}) = \sum_{\mathbf{e}_{i,j}\in\mathbf{E}} (\lambda_i + \lambda_j)\cdot\mathbf{C}_{i,j}\left(\frac{\lambda_j}{\lambda_i+\lambda_j}\right) - \sum_{k=1}^{n}(d_k - 1)\cdot\lambda_k\cdot\mathbf{p}_k, \tag{13}$$

where $\mathbf{x} \in \Phi$ and $\{\lambda_k\}_{k=1}^n$ are the generalized barycentric coordinates of $\mathbf{x}$ with respect to the n vertices, and d_k is the degree of $\mathbf{v}_k$.

Equation (11) is a particular case of Equation (13), as in the 2D scenario, each vertex of the polygon has an ordinary degree of 2. By simple calculation, we can derive the following two corollaries. Since the proofs follow the same approach as Theorem 3.1, we omit them here.

Corollary 1. *The 3D C^0 GC patch also satisfies the property of generalized barycentric reproduction.*

Corollary 2. *Each boundary surface of a 3D C^0 GC patch is a 2D GC patch.*

Moreover, the C^0 GC patch defined by Equation (13) can be extended to non-manifold topologies. For simplicity's sake, consider the 2D case. Inside a planar polygon are some additional isolated vertices, linear edges, or planar faces. For an isolated vertex $\mathbf{v}_k$, it does not lie on any edge, so its corresponding point $\mathbf{p}_k$ is not calculated in the first term of the right-hand side of Equation (13). But its degree is 0, so the corresponding point will be added once in the second term, i.e., $-(0-1)\cdot\lambda_k\cdot\mathbf{p}_k$. For internal linear edges and planar faces, the situation is similar. Therefore, the C^0 GC patch still works for such non-manifold topologies, as long as we utilize generalized barycentric coordinates well-defined over non-manifold topologies, e.g., harmonic coordinates [Joshi et al. 2007]. Fig. 10 shows C^0 GC patches defined over 2D and 3D with various topologies.

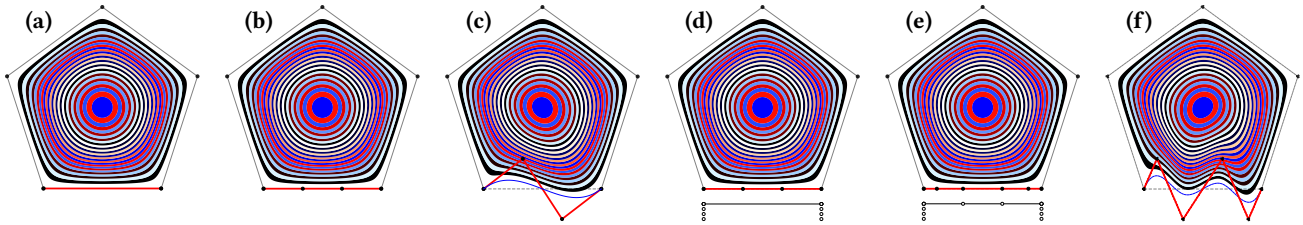

Fig. 11. High-order Cage-based Deformation using free-form curves. **(a)** The original linear cage. **(b)** The bottom linear edge is elevated into cubic Bézier form. **(c)** The bottom cubic Bézier edge is deformed. **(d)** The bottom degree elevated linear edge is transformed into cubic B-spline form with an open uniform knot vector $[0, 0, 0, 0, 1, 1, 1, 1]$. **(e)** The bottom cubic B-spline edge is knot-inserted. **(f)** The bottom knot-inserted cubic B-spline edge is deformed.

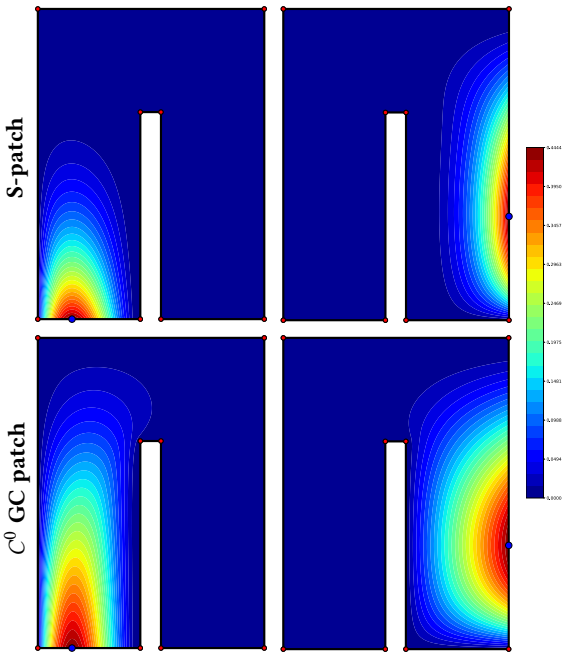

Fig. 12. Basis functions associated with some inserted control points in Fig. 1. The boundary control points inserted by the C^0 GC patch have a stronger global influence than the S-patch. Top row: S-patch. Bottom row: C^0 GC patch.

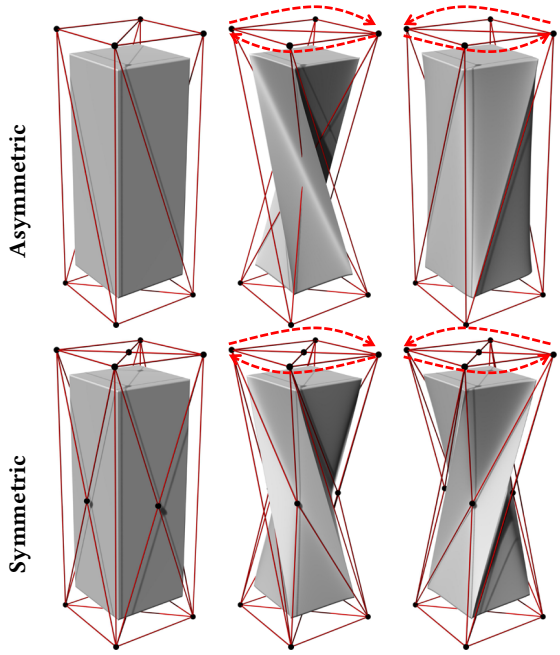

Fig. 13. Twisting a bar by $\frac{\pi}{2}$ in clockwise and counterclockwise directions using asymmetrically and symmetrically triangulated cages. Top row: each quadrilateral is directly split into 2 triangles. Bottom row: each quadrilateral is split into 4 triangles by the centroid. MVCs were used for deformation.

4 HIGH-ORDER CAGE-BASED DEFORMATION

We derive the high-order cage-based deformation method from the C^0 GC patch introduced above. The key idea is first to reproduce the entire cage space using the C^0 GC patch with linear boundaries (i.e., Equation (12)) and then deform the cage space by modifying these boundary curves.

4.1 Deformation using free-form curves

Bézier curves are the simplest and most commonly used free-form curves. It can be found that $\mathbf{C}_k(t)$ (Equation (12)) is a degree 1 Bézier curve determined by the *Bézier* points $\mathbf{v}_k$ and $\mathbf{v}_{k+1}$. Then, by using the degree elevation formula, we can obtain a curve of a higher degree, which preserves the same geometry but has more control points. Finally, moving these control points generates a desired Bézier curve, meaning the linear cage is deformed into a curved cage (see Fig. 11).

When using Bézier curves, our method looks similar to [Smith and Schaefer 2015]. However, it is important to note that our method differs fundamentally from theirs. Their method essentially utilizes the elevation algorithm of S-patches (i.e., multivariate Bernstein polynomials), whereas our method only employs the degree elevation of Bézier curves (i.e., univariate Bernstein polynomials). Fig. 12 shows that the same control points inserted by the two methods have different influences in the domain.

Bézier curves may not fit the control polyline well for high degrees. Then, using B-spline curves would be a good choice, as they possess local support. Here, we use cubic B-splines, which have a global second-order continuity. First, the original linear edge $\mathbf{C}_k(t)$ (Equation (12)) is elevated to a cubic Bézier curve with a colinear

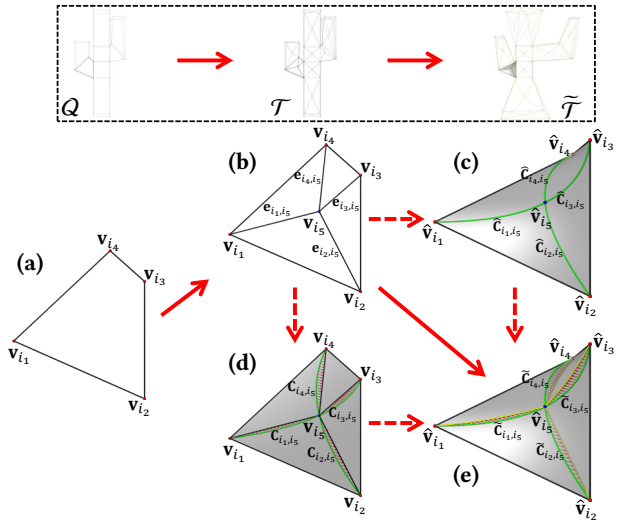

Fig. 14. Automatic setting of the deformed splitting edges. Q: the original quadrilateral cage. $\mathcal{T}$: the original symmetrically triangulated cage with linear splitting edges. $\widetilde{\mathcal{T}}$: the deformed high-order triangular cage with high-order splitting edges. **(a)** The original quadrilateral with vertices $\{\mathbf{v}_{i_k}\}_{k=1}^4$. **(b)** The original quadrilateral is symmetrically triangulated via the centroid $\mathbf{v}_{i_5}$ and the 4 splitting linear edges $\{\mathbf{e}_{i_k,i_5}\}_{k=1}^4$. **(c)** The diagonal (green) curves $\{\widehat{\mathbf{C}}_{i_k,i_5}\}_{k=1}^4$ of the bilinear patch determined by the updated 4 vertices $\{\widehat{\mathbf{v}}_{i_k}\}_{k=1}^4$. **(d)** The diagonal (green) curves $\{\mathbf{C}_{i_k,i_5}\}_{k=1}^4$ of the bilinear patch determined by the original 4 vertices $\{\mathbf{v}_{i_k}\}_{k=1}^4$. **(e)** The deformed splitting edges $\{\widetilde{\mathbf{C}}_{i_k,i_5} = \widehat{\mathbf{C}}_{i_k,i_5} + (\mathbf{e}_{i_k,i_5} - \mathbf{C}_{i_k,i_5})\}_{k=1}^4$ (yellow curves).

control polyline. Then, transform the degree elevated cubic Bézier curve into a cubic B-spline curve with an open uniform knot vector $[0, 0, 0, 0, 1, 1, 1, 1]$. Next, by inserting knots, we obtain additional control points. Finally, by moving the control points, the original linear edge is deformed into a cubic B-spline curve (see Fig. 11).

In addition, by introducing weights, we can use NURBS curves with more substantial representation capabilities (see Fig. 16).

4.2 Pseudo-quad cage-based deformation

As stated in the survey [Ströter et al. 2024], most 3D generalized barycentric coordinates rely on triangular cages, with only a few cage coordinates being well-defined on polygon meshes. For designers or artists, the more common choice might be quadrilateral meshes. QMVC [Thiery et al. 2018] handles non-planar quadrilateral meshes, but it may produce negative values for non-convex cages. HC [Joshi et al. 2007] handles planar polygon meshes and ensure non-negativity, but it may encounter difficulties when dealing with non-planar quadrilateral meshes [Ströter et al. 2024]. Therefore, it is often necessary to triangulate the quadrilateral cages. Fig. 13 demonstrates that using asymmetrically triangulated quadrilateral cages for deformation results in counterintuitive asymmetric outcomes despite symmetric operations. On the contrary, the symmetry during deformation is preserved using symmetrically triangulated quadrilateral cages.

However, deformation using symmetrically triangulated quadrilateral cages still leads to undesired triangulation artifacts. We propose a *pseudo-quad cage-based deformation* method to address this issue. An important observation is that the splitting edges cause these artifacts. Triangulation leads to non-smoothness on the quadrilateral faces, specifically along the splitting edges, and this non-smoothness will affect the interior (see Fig. 13). Therefore, appropriately adjusting these splitting edges would reduce triangulation artifacts.

We adopt a simple and effective strategy to automatically set the deformed splitting edges. For a quadrilateral $\mathbf{v}_{i_1}\mathbf{v}_{i_2}\mathbf{v}_{i_3}\mathbf{v}_{i_4}$, it is split into 4 triangles by its centroid $\mathbf{v}_{i_5} = \sum_{k=1}^4 \mathbf{v}_{i_k}/4$, where $\{\mathbf{e}_{i_k,i_5}\}_{k=1}^4$ are the 4 linear splitting edges. During the deformation process, we only need to manually update the positions of the vertices of the original quadrilateral, i.e., $\{\mathbf{v}_{i_k}\}_{k=1}^4 \rightarrow \{\widehat{\mathbf{v}}_{i_k}\}_{k=1}^4$. $\mathbf{v}_{i_5}$ is transformed into the centroid $\widehat{\mathbf{v}}_{i_5}$ of the new quadrilateral $\widehat{\mathbf{v}}_{i_1}\widehat{\mathbf{v}}_{i_2}\widehat{\mathbf{v}}_{i_3}\widehat{\mathbf{v}}_{i_4}$. To reduce triangulation artifacts, these linear splitting edges $\{\mathbf{e}_{i_k,i_5}\}_{k=1}^4$ are mapped to high-order curved edges. A simple and natural choice is to map them to the diagonal curves $\{\widehat{\mathbf{C}}_{i_k,i_5}\}_{k=1}^4$ of the bilinear patch determined by $\{\widehat{\mathbf{v}}_{i_k}\}_{k=1}^4$. But this raises a problem: when the vertices of the quadrilateral fall back to their initial positions, these diagonal curves $\{\mathbf{C}_{i_k,i_5}\}_{k=1}^4$ usually cannot reproduce the original linear splitting edges unless the quadrilateral is a parallelogram. To overcome this problem, we apply displacements to these high-order curved splitting edges so that they can fall back to their original linear form. That is, the original 4 linear splitting edges are deformed into 4 high-order curved splitting edges $\{\widetilde{\mathbf{C}}_{i_k,i_5} = \widehat{\mathbf{C}}_{i_k,i_5} + (\mathbf{e}_{i_k,i_5} - \mathbf{C}_{i_k,i_5})\}_{k=1}^4$ (see Fig. 14).

5 RESULTS

In this section, we present some examples, including 2D deformation and 3D deformation, to demonstrate the advantages of our method.

5.1 2D deformation

2D deformation using cubic Bézier curves. Fig. 1 presents a comparison of different methods, including linear CBD, cubic MVC [Li et al. 2013], selective degree elevation of the S-patch [Smith and Schaefer 2015], and the C^0 GC patch. We used linear cages of different resolutions for linear CBD, containing 8 vertices, 24 vertices, and 72 vertices, respectively. The other methods used the same cubic cage with 8 vertices. All methods except for cubic MVC used HC for the calculations. The deformation results of linear CBD are non-smooth at the boundaries, whereas high-order methods maintain smooth boundaries. Cubic MVC produces severe artifacts because it has negative values for concave polygons. As for the selective degree elevation method, the inserted boundary control points belong to a high-order S-patch and have weak control over the interior, so artifacts (green rectangle) in the result occur near the boundary. In contrast, the C^0 GC patch-based method propagates boundary deformations to the interior in a reasonable manner (see Fig. 12). Compared to other methods, our approach produces a better result, evident in smoother deformations and fewer artifacts.

2D deformation using B-spline curves. Fig. 15 shows the differences in high-order deformation results between using Bézier and B-spline curves with the same cages. When deforming some long edges, using B-spline curves with locality for deformation might yield results that

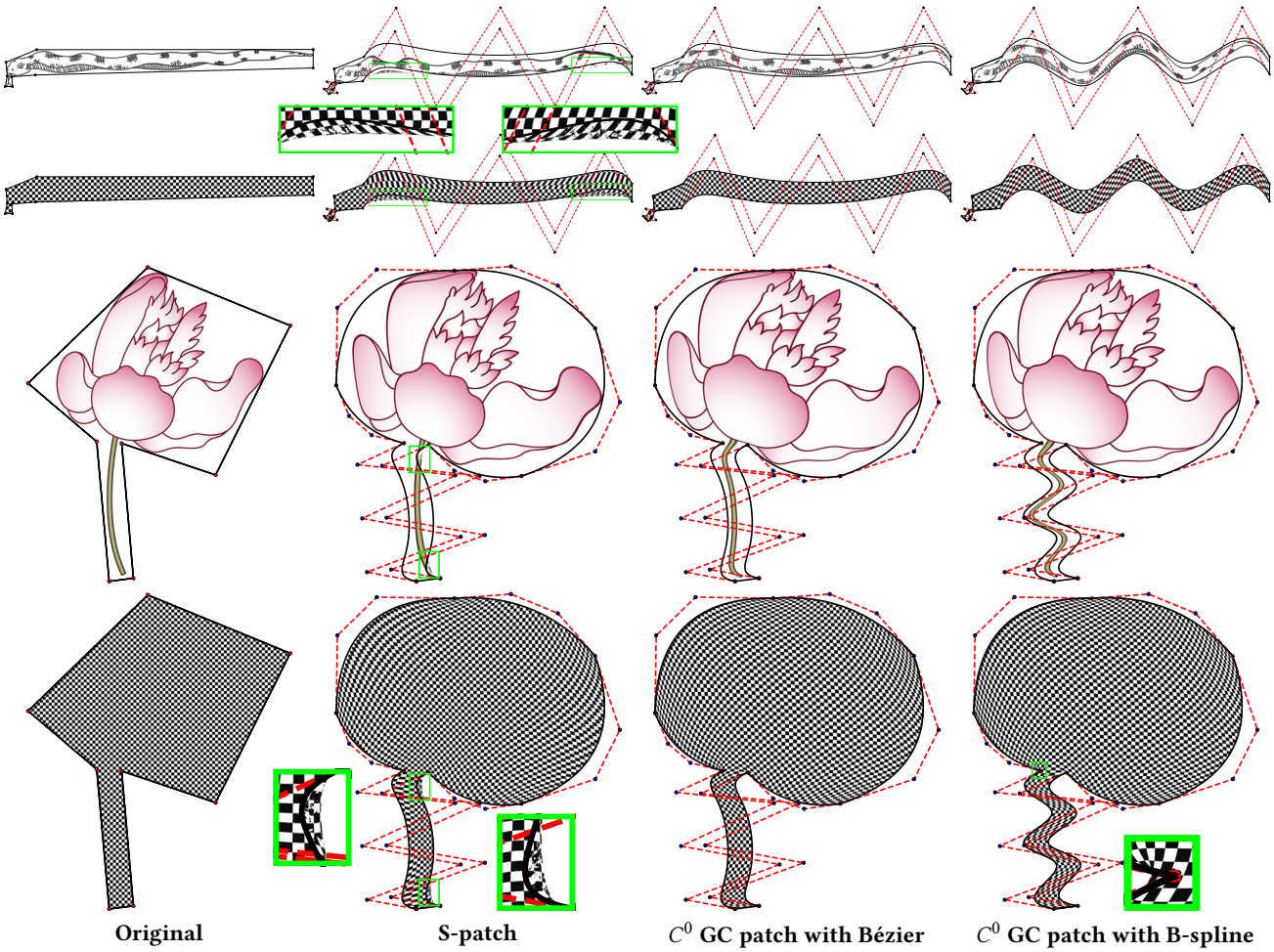

Fig. 15. **Snake** & **Lotus**. From left to right: the original objects; high-order CBD results by selective degree elevation of the S-patch; high-order CBD results by C^0 GC patches with Bézier boundaries; and high-order CBD results by C^0 GC patches with B-spline boundaries. Green rectangle: fold-over. HCs were used for the patch evaluation (both the S-patches and the C^0 GC patches).

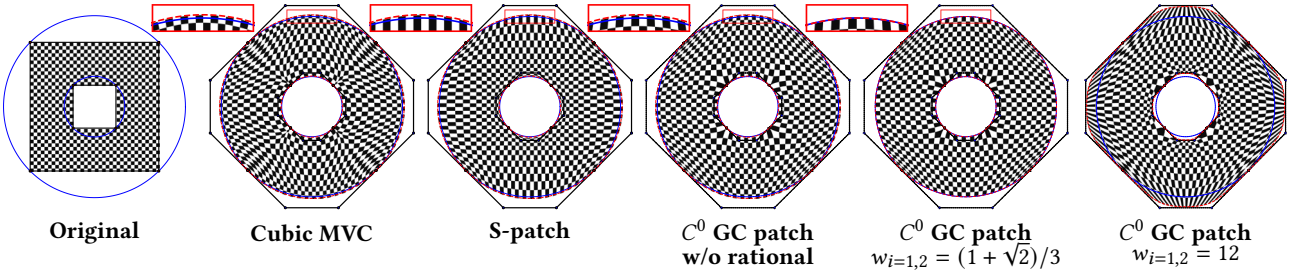

Fig. 16. The original object is an octagon with a hole, and the vertices lie on two circles (blue solid lines). Using the same cubic cage but different methods of deformation, each linear edge was transformed into a (rational) cubic Bézier curve (red dashed lines). Cubic MVC and selective degree elevation method only handle polynomial boundaries, but the C^0 GC patch is also applicable to rational polynomial boundaries. HCs were used for the patch evaluation (both the S-patch and C^0 GC patches).

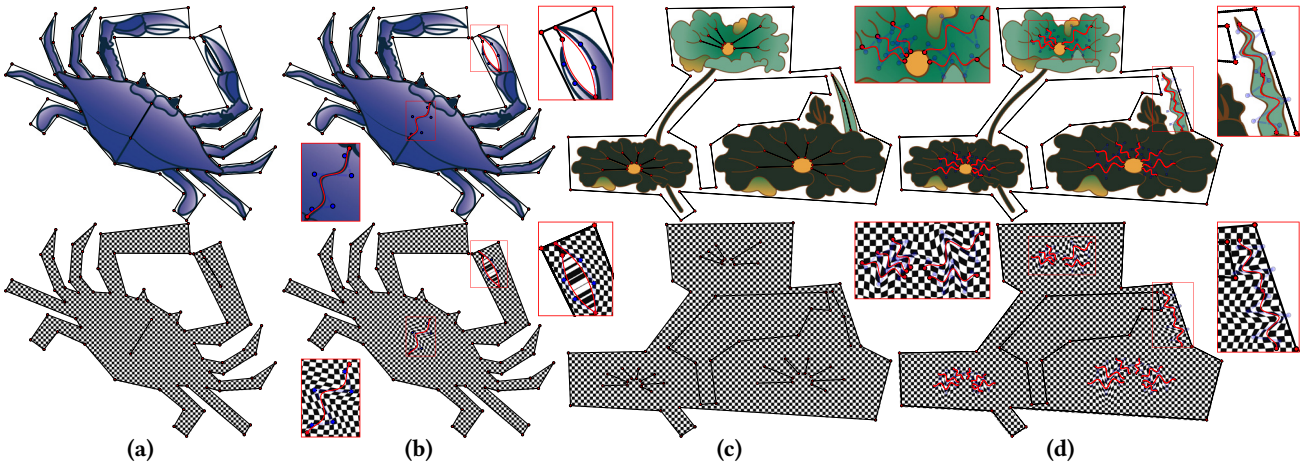

Fig. 17. **Crab** & **Lotus-leaves**. **(a)** & **(c)** the original objects. **(b)** & **(d)** the deformed objects. HCs were used for the patch evaluation.

better meet the user's expectations. We also present the deformation results using selective degree elevation of the S-patch. The figure shows that their method produces severe counterintuitive artifacts (green rectangle) during deformations, even when the deformations are not large-scale. For the **Lotus** image, our method produces slight folding (green rectangle) near the corner when using B-spline because the local deformation, caused by the locality, is too severe.

2D deformation using rational Bézier curves. Unlike existing methods that only handle polynomial boundaries, our method also handles rational polynomial boundaries. Fig. 16 illustrates the deformation results obtained using rational cubic Bézier curves. By introducing weights for control points, the C^0 GC patch has greater representational capability. The figure shows that by setting the weights to $(1, (1+\sqrt{2})/3, (1+\sqrt{2})/3, 1)$, our method transformed each linear edge into a quarter-circle, which cubic MVC and selective degree elevation method can not. By increasing the weights of the two middle control points of each edge, the boundary curve will be pulled toward the control polyline, thereby deforming the interior.

2D deformation of non-manifold topologies. The C^0 GC patch-based method handles non-manifold topologies as long as the used generalized barycentric coordinates are well-defined. Fig. 17 presents the deformation results of non-manifold topologies. The figure presents that the deformation is not smooth when crossing internal boundaries, as HCs are only C^0 at the boundaries. However, users still benefit from this deformation, as shown in the figure, by introducing additional isolated edges to edit the feature lines in the images.

5.2 3D deformation

3D deformation using cubic Bézier curves. Fig. 18 compares the deformation results of twisting a bar using different methods. When using the original linear cage (8 vertices) to twist the bar by π, the result was inferior, with the middle part collapsing into a single point. Using a refined cage (16 vertices) could improve the linear deformation results; however, the linear boundaries still affect the interior, leading to a visually "unsmooth" appearance. High-order deformation produces smoother results. However, using selective degree elevation of the S-patch for deformation, the bar exhibits a certain degree of shrinkage, similar to the 2D case, due to the weaker internal control exerted by inserted boundary points. Compared to other methods, the result obtained using a C^0 GC patch for deformation is more natural and smoother. Fig. 19 shows a similar comparison of bending the bar.

Pseudo-quad cage-based deformation. Fig. 20 shows the deformation of a complex model **Spikybox**. The original cage is a convex quadrilateral cage. Deformation using MVC with triangulated quadrilateral cages results in apparent triangulation artifacts. Our pseudo-quad cage-based deformation significantly reduces the undesired artifacts. The result is similar to that obtained using QMVC. Fig. 21 shows a more complex example of the deformation of the **Cactus** model, where the original cage is a non-planar concave quadrilateral cage. In this case, QMVC deformation resulted in counterintuitive artifacts (red rectangle) due to negative coordinates. Deformation using HC with triangulated quadrilateral cages avoids these counterintuitive artifacts but introduces triangulation artifacts (blue rectangle). The pseudo-quad cage-based deformation method avoids these artifacts and produces a result more closely aligns with expectations. However, we must emphasize that our method is merely a compromise as we just use triangulated quadrilateral cages.

6 LIMITATIONS

The first limitation of our method is that the generalized barycentric coordinates we used must satisfy the non-negativity. When negative generalized barycentric coordinates are input, extreme artifacts will result. In contrast, previous work [Langer et al. 2008; Smith and Schaefer 2015] has shown that S-patches still work even if the generalized barycentric coordinates have negative values. Nevertheless, S-patches cannot avoid counterintuitive artifacts caused by negative coordinates. Over the past two decades, researchers have been dedicated to constructing non-negative generalized barycentric coordinates [Chang et al. 2023; Dodik et al. 2023; Hormann and

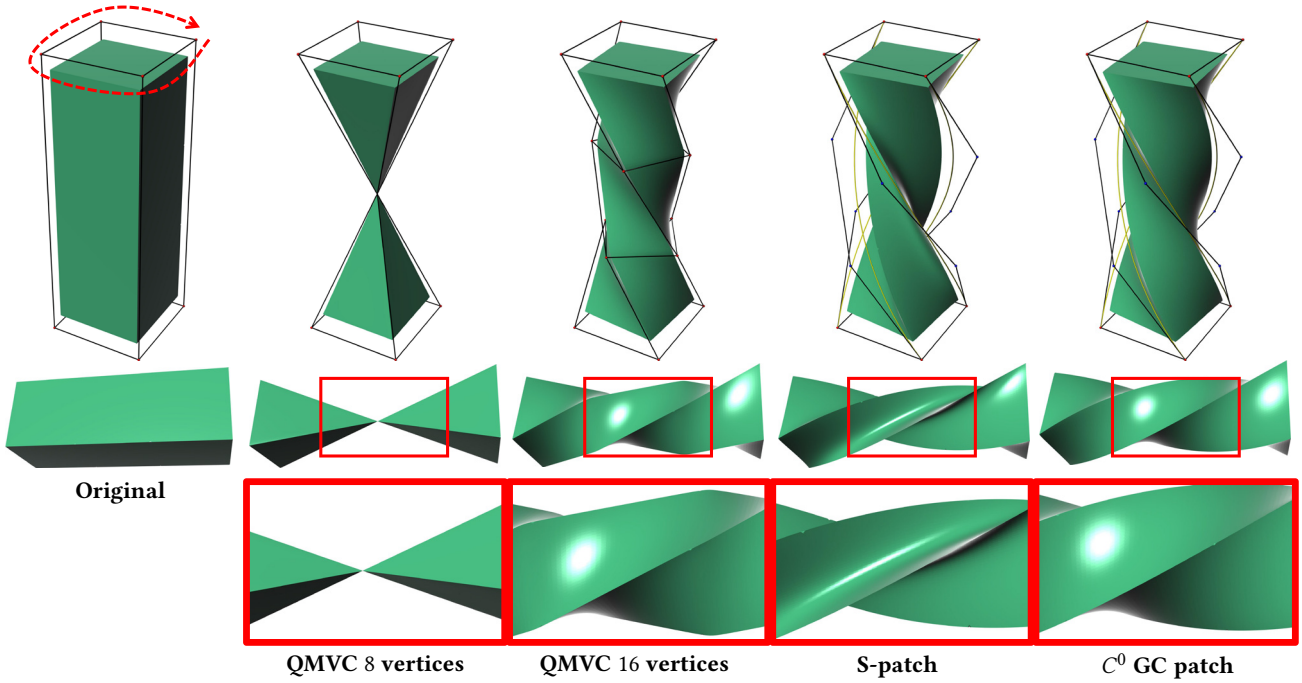

Fig. 18. **Bar-twisting**. Twisting a bar by π in a clockwise direction. From left to right: the original object; QMVC deformation with 8 vertices; QMVC deformation with 16 vertices; deformation using selective degree elevation of the S-patch; and deformation using C^0 GC patch. QMVCs were used for the patch evaluation (both the S-patch and the C^0 GC patch).

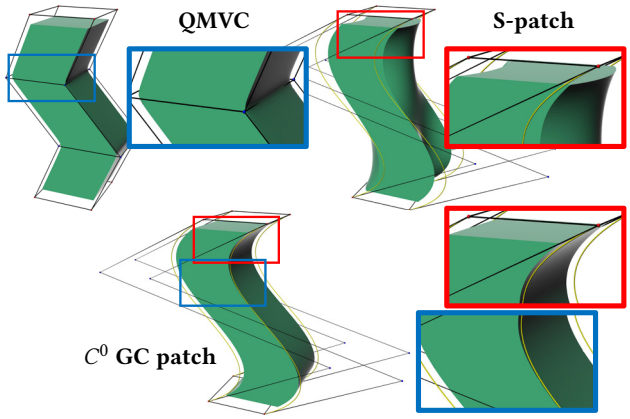

Fig. 19. **Bar-bending**. Top left: QMVC deformation with 16 vertices. Top right: deformation using selective degree elevation of the S-patch. Bottom: deformation using C^0 GC patch. QMVCs were used for the patch evaluation (both the S-patch and the C^0 GC patch).

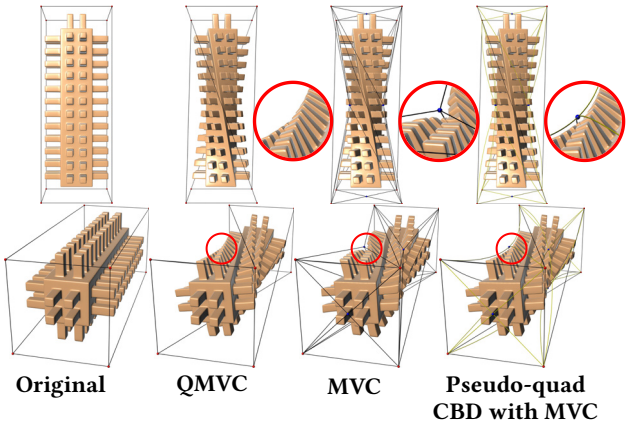

Fig. 20. **Spikybox**. From left to right: the original object; QMVC deformation; MVC deformation; and pseudo-quad cage-based deformation.

Sukumar 2008; Joshi et al. 2007; Lipman et al. 2007]. Therefore, this limitation will not hinder the use of our method. In addition, since the high-order methods rely on generalized barycentric coordinates, if the input coordinates (e.g., MEC) are not "geometry-aware," the deformation results will also be suboptimal (compared with HC) (see Fig. 22).

The second limitation is that our GC patches only achieve C^0 interpolation, which may not suit for *cage networks*. Although various parametric polygonal transfinite interpolation surfaces exist that achieve high-order interpolation, as stated earlier, they are unsuitable for deformation. It is worth investigating how to construct a transfinite interpolation scheme that achieves both high-order interpolation and satisfies the property of generalized barycentric reproduction. Additionally, some generalized barycentric coordinates

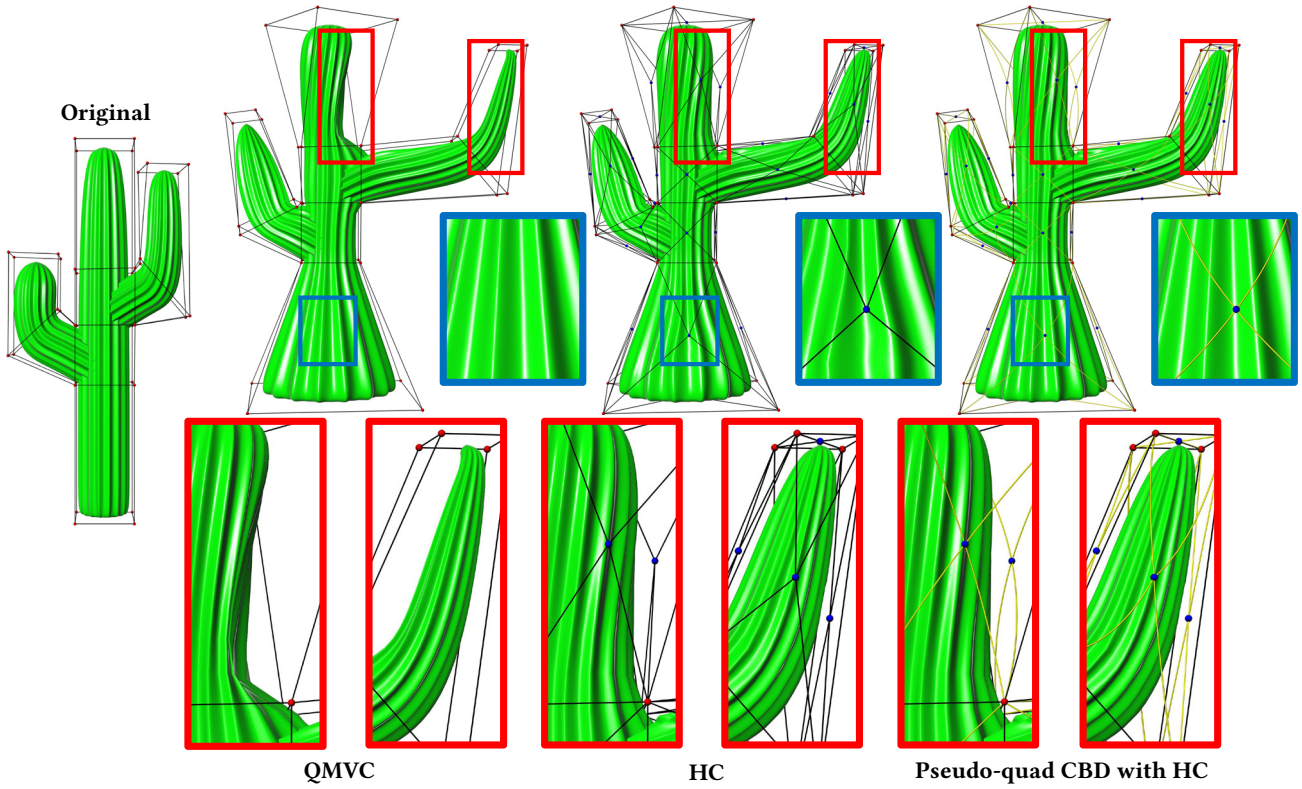

Fig. 21. **Cactus**. From left to right: the original object; QMVC deformation; HC deformation; and pseudo-quad cage-based deformation.

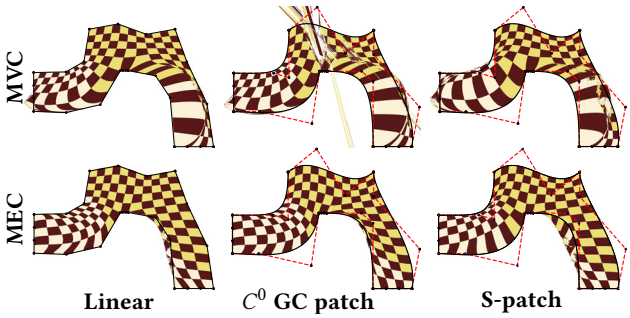

Fig. 22. Deforming the original pant-shape polygon in Fig. 1 by different coordinates. Top row: MVC. Bottom row: MEC. From left to right: linear CBD; deformation using C^0 GC patch; and deformation using selective degree elevation of the S-patch. Since the original cage is concave, MVC produces negative values. Although MEC [Hormann and Sukumar 2008] ensures non-negativity, its prior function is not "geometry-aware" [Ströter et al. 2024]. Whether the deformation is linear or high-order, the results obtained using MVC and MEC are inferior compared to those using HC.

are capable of achieving high-order interpolation, such as biharmonic coordinates [Weber et al. 2012] and moving least squares coordinates [Manson and Schaefer 2010]. We speculate that these generalized barycentric coordinates might apply to our method.

Moreover, we currently do not have a method to ensure injectivity during the high-order CBD process (see Fig. 15). It is necessary to study the regularity of C^0 GC patches to address this issue, which is undoubtedly very challenging because it involves composite functions of generalized barycentric coordinates. It is worth noting that [Randrianarivony 2007] studied the regularity of planar Coons maps with polynomial boundaries.

7 CONCLUSION

We have introduced the C^0 GC patch, which unifies and generalizes the Coons patch and the triangular Coons patch. The C^0 GC patch is defined in 2D domains and 3D domains or higher-dimensional domains with arbitrary polytopes, including non-manifold topologies. Besides, the C^0 GC patch satisfies the property of generalized barycentric reproduction. By combining the degree elevation of Bézier curves and the knot insertion of B-spline curves, we have successfully applied the C^0 GC patch to space deformation, enabling high-order CBD. Compared to linear CBD, high-order CBD offers more freedom and acquires smoother results. Moreover, except for the non-negativity constraint, the C^0 GC patch is independent of the choice of generalized barycentric coordinates. Our method may promote the application of the CBD technique and inspire further research on generalized barycentric coordinates.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their constructive suggestions and comments. This work was supported by the National Natural Science Foundation of China (NSFC) under the project number 61872121.

REFERENCES

Robert E. Barnhill. 1977. Representation and Approximation of Surfaces. In *Mathematical Software*, John R. Rice (Ed.). Academic Press, 69–120. https://doi.org/10.1016/B978-0-12-587260-7.50008-X

M. Botsch, L. Kobbelt, P. Pauly, M.and Alliez, and B. Levy. 2010. *Polygon Mesh Processing* (1st ed.). A K Peters/CRC Press, New York. https://doi.org/10.1201/b10688

Mario Botsch, Mark Pauly, Christian Rossl, Stephan Bischoff, and Leif Kobbelt. 2006. Geometric modeling based on triangle meshes. In *ACM SIGGRAPH 2006 Courses* (Boston, Massachusetts) *(SIGGRAPH '06)*. Association for Computing Machinery, New York, NY, USA, 1–es. https://doi.org/10.1145/1185657.1185839

Max Budninskiy, Beibei Liu, Yiying Tong, and Mathieu Desbrun. 2016. Power coordinates: a geometric construction of barycentric coordinates on convex polytopes. *ACM Trans. Graph.* 35, 6, Article 241 (dec 2016), 11 pages. https://doi.org/10.1145/2980179.2982441

Q. Chang, C. Deng, and K. Hormann. 2023. Maximum Likelihood Coordinates. *Computer Graphics Forum* 42, 5 (2023), e14908. https://doi.org/10.1111/cgf.14908 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.14908

Peter Charrot and John A. Gregory. 1984. A pentagonal surface patch for computer aided geometric design. *Computer Aided Geometric Design* 1, 1 (1984), 87–94. https://doi.org/10.1016/0167-8396(84)90006-2

S. A. Coons. 1967. *SURFACES FOR COMPUTER-AIDED DESIGN OF SPACE FORMS*. Technical Report. USA.

Sabine Coquillart. 1990. Extended free-form deformation: a sculpturing tool for 3D geometric modeling. In *Proceedings of the 17th Annual Conference on Computer Graphics and Interactive Techniques* (Dallas, TX, USA) *(SIGGRAPH '90)*. Association for Computing Machinery, New York, NY, USA, 187–196. https://doi.org/10.1145/97879.97900

Chongyang Deng, Qingjun Chang, and Kai Hormann. 2020. Iterative coordinates. *Computer Aided Geometric Design* 79 (2020), 101861. https://doi.org/10.1016/j.cagd.2020.101861

Ana Dodik, Oded Stein, Vincent Sitzmann, and Justin Solomon. 2023. Variational Barycentric Coordinates. *ACM Trans. Graph.* 42, 6, Article 255 (dec 2023), 16 pages. https://doi.org/10.1145/3618403

Michael S. Floater. 2015. Generalized barycentric coordinates and applications. *Acta Numerica* 24 (2015), 161–214. https://doi.org/10.1017/S0962492914000129

Michael S. Floater, Géza Kós, and Martin Reimers. 2005. Mean value coordinates in 3D. *Computer Aided Geometric Design* 22, 7 (2005), 623–631. https://doi.org/10.1016/j.cagd.2005.06.004 Geometric Modelling and Differential Geometry.

JA Gregory. 1986. N-sided surface patches. *The Mathematics of Surfeces* (1986), 217–232.

John A. Gregory. 1974. SMOOTH INTERPOLATION WITHOUT TWIST CONSTRAINTS. In *Computer Aided Geometric Design*, ROBERT E. BARNHILL and RICHARD F. RIESENFELD (Eds.). Academic Press, 71–87. https://doi.org/10.1016/B978-0-12-079050-0.50009-6

John A. Gregory and Jörg M. Hahn. 1989. A C2 polygonal surface patch. *Computer Aided Geometric Design* 6, 1 (1989), 69–75. https://doi.org/10.1016/0167-8396(89)90007-1

Josef Griessmair and Werner Purgathofer. 1989. Deformation of Solids with Trivariate B-Splines. In *EG 1989-Technical Papers*. Eurographics Association. https://doi.org/10.2312/egtp.19891010

Kai Hormann and Michael S. Floater. 2006. Mean value coordinates for arbitrary planar polygons. *ACM Trans. Graph.* 25, 4 (oct 2006), 1424–1441. https://doi.org/10.1145/1183287.1183295

K. Hormann and N. Sukumar. 2008. Maximum entropy coordinates for arbitrary polytopes. In *Proceedings of the Symposium on Geometry Processing* (Copenhagen, Denmark) *(SGP '08)*. Eurographics Association, Goslar, DEU, 1513–1520.

Pushkar Joshi, Mark Meyer, Tony DeRose, Brian Green, and Tom Sanocki. 2007. Harmonic coordinates for character articulation. *ACM Trans. Graph.* 26, 3 (jul 2007), 71–es. https://doi.org/10.1145/1276377.1276466

Tao Ju, Scott Schaefer, and Joe Warren. 2005. Mean value coordinates for closed triangular meshes. *ACM Trans. Graph.* 24, 3 (jul 2005), 561–566. https://doi.org/10.1145/1073204.1073229

K. Kato. 1991. Generation of N-sided surface patches with holes. *Computer-Aided Design* 23, 10 (1991), 676–683. https://doi.org/10.1016/0010-4485(91)90020-W

Henry J. Lamousin and Warren N. Waggenspack Jr. 1994. NURBS-Based Free-Form Deformations. *IEEE Comput. Graph. Appl.* 14, 6 (nov 1994), 59–65. https://doi.org/10.1109/38.329096

Torsten Langer, Alexander Belyaev, and Hans-Peter Seidel. 2008. Mean value Bézier maps. In *Proceedings of the 5th International Conference on Advances in Geometric Modeling and Processing* (Hangzhou, China) *(GMP'08)*. Springer-Verlag, Berlin, Heidelberg, 231–243.

Xian-Ying Li, Tao Ju, and Shi-Min Hu. 2013. Cubic mean value coordinates. *ACM Trans. Graph.* 32, 4, Article 126 (jul 2013), 10 pages. https://doi.org/10.1145/2461912.2461917

Yaron Lipman, Johannes Kopf, Daniel Cohen-Or, and David Levin. 2007. GPU-assisted positive mean value coordinates for mesh deformations. In *Proceedings of the Fifth Eurographics Symposium on Geometry Processing* (Barcelona, Spain) *(SGP '07)*. Eurographics Association, Goslar, DEU, 117–123.

Yaron Lipman, David Levin, and Daniel Cohen-Or. 2008. Green Coordinates. In *ACM SIGGRAPH 2008 Papers* (Los Angeles, California) *(SIGGRAPH '08)*. Association for Computing Machinery, New York, NY, USA, Article 78, 10 pages. https://doi.org/10.1145/1399504.1360677

Charles T. Loop and Tony D. DeRose. 1989. A multisided generalization of Bézier surfaces. *ACM Trans. Graph.* 8, 3 (jul 1989), 204–234. https://doi.org/10.1145/77055.77059

Ron MacCracken and Kenneth I. Joy. 1996. Free-form deformations with lattices of arbitrary topology. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '96)*. Association for Computing Machinery, New York, NY, USA, 181–188. https://doi.org/10.1145/237170.237247

Josiah Manson and Scott Schaefer. 2010. Moving Least Squares Coordinates. *Computer Graphics Forum* 29, 5 (2010), 1517–1524. https://doi.org/10.1111/j.1467-8659.2010.01760.x arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-8659.2010.01760.x

J.A. Marshall. 1975. *Some Applications of Blending Function Techniques to Finite Element Methods*. Ph. D. Dissertation. University of Dundee.

Élie Michel and Jean-Marc Thiery. 2023. Polynomial 2D Green Coordinates for Polygonal Cages. In *ACM SIGGRAPH 2023 Conference Proceedings* (Los Angeles, CA, USA) *(SIGGRAPH '23)*. Association for Computing Machinery, New York, NY, USA, Article 23, 9 pages. https://doi.org/10.1145/3588432.3591499

Gregory M. Nielson. 1979. The side-vertex method for interpolation in triangles. *Journal of Approximation Theory* 25, 4 (1979), 318–336. https://doi.org/10.1016/0021-9045(79)90020-0

Kaikai Qin, Yajuan Li, and Chongyang Deng. 2023. Blending Bézier patch for multi-sided surface modeling. *Computer Aided Geometric Design* 105 (2023), 102222. https://doi.org/10.1016/j.cagd.2023.102222

Kaikai Qin, Yajuan Li, and Chongyang Deng. 2024. Generalized Bézier volumes over simple convex polyhedra. *Computer Aided Geometric Design* 111 (2024), 102338. https://doi.org/10.1016/j.cagd.2024.102338

Maharavo Randrianarivony. 2007. Sufficient and necessary conditions for the regularity of planar Coons maps. https://api.semanticscholar.org/CorpusID:14057030

Maharavo Randrianarivony. 2011. On transfinite interpolations with respect to convex domains. *Computer Aided Geometric Design* 28, 2 (2011), 135–149. https://doi.org/10.1016/j.cagd.2010.10.003

João Pedro Duro Reis and Jiří Kosinka. 2018. Injective hierarchical free-form deformations using THB-splines. *Computer-Aided Design* 100 (2018), 30–38. https://doi.org/10.1016/j.cad.2018.02.005

Péter Salvi. 2020. A multi-sided generalization of the C^0 Coons patch. In *Proceedings of the Workshop on the Advances of Information Technology*. BME, 110–111. arXiv:2002.11347

Péter Salvi. 2024. Intuitive interior control for multi-sided patches with arbitrary boundaries. *Computer-Aided Design and Applications* 21, 1 (2024), 143–154. https://doi.org/10.14733/cadaps.2024.143-154

Péter Salvi, Márton Vaitkus, and Tamás Várady. 2023. Constrained modeling of multi-sided patches. *Computers & Graphics* 114 (2023), 86–95. https://doi.org/10.1016/j.cag.2023.05.020

Péter Salvi, Tamás Várady, and Alyn Rockwood. 2014. Ribbon-based transfinite surfaces. *Computer Aided Geometric Design* 31, 9 (2014), 613–630. https://doi.org/10.1016/j.cagd.2014.06.006

Scott Schaefer. 2017. *Generalized Barycentric Coordinates in Computer Graphics and Computational Mechanics* (1st ed.). CRC Press, Boca Raton, Chapter Multi-Sided Patches via Barycentric Coordinates, 135–146.

Thomas W. Sederberg and Scott R. Parry. 1986. Free-form deformation of solid geometric models. *SIGGRAPH Comput. Graph.* 20, 4 (aug 1986), 151–160. https://doi.org/10.1145/15886.15903

J. Smith and S. Schaefer. 2015. Selective Degree Elevation for Multi-Sided Bézier Patches. *Computer Graphics Forum* 34, 2 (2015), 609–615. https://doi.org/10.1111/cgf.12588 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.12588

Daniel Ströter, Jean-Marc Thiery, Kai Hormann, Jiong Chen, Qingjun Chang, Sebastian Besler, Johannes Sebastian Mueller-Roemer, Tamy Boubekeur, André Stork, and Dieter W. Fellner. 2024. A Survey on Cage-based Deformation of 3D Models. *Computer Graphics Forum* (2024). https://doi.org/10.1111/cgf.15060

Jean-Marc Thiery, Pooran Memari, and Tamy Boubekeur. 2018. Mean value coordinates for quad cages in 3D. *ACM Trans. Graph.* 37, 6, Article 229 (dec 2018), 14 pages. https://doi.org/10.1145/3272127.3275063

Tamás Várady, Péter Salvi, and Alyn Rockwood. 2012. Transfinite surface interpolation with interior control. *Graph. Models* 74, 6 (nov 2012), 311–320. https://doi.org/10.1016/j.gmod.2012.03.003

Tamás Várady, Alyn Rockwood, and Péter Salvi. 2011. Transfinite surface interpolation over irregular n-sided domains. *Computer-Aided Design* 43, 11 (2011), 1330–1340. https://doi.org/10.1016/j.cad.2011.08.028 Solid and Physical Modeling 2011.

Tamás Várady, Péter Salvi, and György Karikó. 2016. A Multi-sided Bézier Patch with a Simple Control Structure. *Computer Graphics Forum* 35, 2 (2016), 307–317. https://doi.org/10.1111/cgf.12833 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.12833

Tamás Várady, Péter Salvi, and Márton Vaitkus. 2024. Genuine multi-sided parametric surface patches – A survey. *Computer Aided Geometric Design* 110 (2024), 102286. https://doi.org/10.1016/j.cagd.2024.102286

Eugene L. Wachspress. 1975. *A Rational Finite Element Basis.*

Ofir Weber, Roi Poranne, and Craig Gotsman. 2012. Biharmonic Coordinates. *Comput. Graph. Forum* 31, 8 (dec 2012), 2409–2422. https://doi.org/10.1111/j.1467-8659.2012.03130.x

Juyong Zhang, Bailin Deng, Zishun Liu, Giuseppe Patanè, Sofien Bouaziz, Kai Hormann, and Ligang Liu. 2014. Local barycentric coordinates. *ACM Trans. Graph.* 33, 6, Article 188 (nov 2014), 12 pages. https://doi.org/10.1145/2661229.2661255

Yuzhe Zhang, Jianmin Zheng, and Yiyu Cai. 2020. Proxy-driven free-form deformation by topology-adjustable control lattice. *Computers & Graphics* 89 (2020), 167–177. https://doi.org/10.1016/j.cag.2020.05.013