C^0 Generalized Coons Volumes over Arbitrary Polyhedra

KAIKAI QIN[†] and ZEQI GE[†], Hangzhou Dianzi University, China
PÉTER SALVI, Budapest University of Technology and Economics, Hungary
CHENHAO YING, Hangzhou Dianzi University, China
HUIBIAO WEN, Shandong University, China
KEPENG XU, Xidian University, China
SHIQING XIN, Shandong University, China
CHONGYANG DENG[*], Hangzhou Dianzi University, China

Transfinite interpolation is a fundamental theme in geometric modeling. The most well-known transfinite interpolation volume is the Coons volume. However, the topology of Coons volumes is restricted to hexahedra, which significantly limits their applications. In this paper, we generalize the Coons volume from hexahedral topology to arbitrary polyhedral topologies via generalized barycentric coordinates. We prove that the proposed generalized Coons volume possesses several desirable geometric properties and demonstrate its applications in computer graphics.

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

Additional Key Words and Phrases: Transfinite interpolation, Coons volumes, generalized barycentric coordinates

ACM Reference Format:
Kaikai Qin, Zeqi Ge, Péter Salvi, Chenhao Ying, Huibiao Wen, Kepeng Xu, Shiqing Xin, and Chongyang Deng. 2026. C^0 Generalized Coons Volumes over Arbitrary Polyhedra. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers (SIGGRAPH Conference Papers '26), July 19–23, 2026, Los Angeles, CA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3799902.3811087

1 Introduction

Constructing a volume that interpolates prescribed boundary surfaces is an important topic in computer graphics [Cherchi and Livesu 2023]. In many applications, such as solid texture mapping [Takayama et al. 2008], volume deformation [Su et al. 2019], and mesh generation [Fu et al. 2016; Guo et al. 2020; Li et al. 2021], it is often necessary to construct a mapping from a simple parametric domain (i.e., a polyhedron) to a complex shape.

In geometric modeling, a related and classical volumetric representation is the Coons volume, also known as *transfinite interpolation* [Hoschek et al. 1993]. The term "transfinite" implies that the interpolation is performed over a continuum of data (e.g., a curve or a surface), rather than just at a set of discrete data. Although Coons volumes can interpolate prescribed boundary surfaces, they have seen little use over the past four decades. One major drawback is that they have to be hexahedral, and each boundary surface patch must be quadrilateral. Recently, Qin et al. [2024b] proposed the C^0 generalized Coons (GC) patch, which can be defined over arbitrary polyhedra. However, they can only interpolate boundary curves rather than boundary surfaces.

Contribution. In this paper, we propose the C^0 generalized Coons (GC) volume based on *generalized barycentric coordinates*. The C^0 GC volume can be defined over arbitrary polyhedral domains, whether convex or concave, genus-zero or multi-connected, and interpolate given boundary surfaces, whether quadrilateral or polygonal. Furthermore, we demonstrate that the C^0 GC volume possesses several favorable properties, thereby showing its applications in computer graphics.[1]

1.1 Related work

Generalized barycentric coordinates (GBCs). Wachspress coordinates [Wachspress 1975] were first proposed as shape functions in finite element methods. However, they are defined only over convex polytopes [Warren 1996]. Floater [2003] proposed mean value coordinates, which can be defined over concave domains but may yield negative values. Ju et al. [2005] applied mean value coordinates to *cage-based deformation*, but negative coordinates often lead to counterintuitive artifacts. Therefore, various non-negative coordinates have been successively proposed, such as harmonic coordinates [Joshi et al. 2007], and others [Chang et al. 2023; Deng et al. 2020; Hormann and Sukumar 2008; Lipman et al. 2007].

Beyond the interpolating coordinates mentioned above, approximating coordinates, such as Green coordinates [Lipman et al. 2008] and Cauchy coordinates [Weber et al. 2009], have also been introduced. These coordinates exhibit (quasi-)conformal properties and can preserve local details during deformation. Moreover, some generalized barycentric coordinates have been extended to high-order

[*]Corresponding author.
[†]Equal contribution.

Authors' Contact Information: Kaikai Qin, qin-kaikai@hdu.edu.cn; Zeqi Ge, ge.zeqi@hdu.edu.cn, Hangzhou Dianzi University, Hangzhou, China; Péter Salvi, Budapest University of Technology and Economics, Budapest, Hungary, salvi@iit.bme.hu; Chenhao Ying, Hangzhou Dianzi University, Hangzhou, China, 245070123@hdu.edu.cn; Huibiao Wen, Shandong University, Qingdao, China, ericvein@163.com; Kepeng Xu, Xidian University, Xi'an, China, kepengxu11@gmail.com; Shiqing Xin, Shandong University, Qingdao, China, xinshiqing@sdu.edu.cn; Chongyang Deng, Hangzhou Dianzi University, Hangzhou, China, dcy@hdu.edu.cn.

SIGGRAPH Conference Papers '26, Los Angeles, CA, USA

ACM ISBN 979-8-4007-XXXX-X/2026/07
https://doi.org/10.1145/3799902.3811087

[1] "C^0" means that the proposed generalized Coons volume only interpolates the boundary positions, and does not interpolate the boundary derivatives.

structures to obtain more degrees of freedom and smoother deformation results, such as cubic mean value coordinates [Li et al. 2013], polynomial Green/Cauchy/biharmonic coordinates for 2D polygonal/curved cages [Lin and Chen 2024; Liu et al. 2025a,b; Michel et al. 2025; Michel and Thiery 2023], and 3D Bézier Green coordinates [Xiao and Chen 2025].

Transfinite interpolation approaches. The earliest and perhaps also the best-known transfinite interpolation method is the Coons patch [Coons 1967], which is defined over a rectangular domain and interpolates four given boundary curves. By utilizing Hermite basis functions, the Coons patch can achieve high-order (C^1/C^2) interpolation [Gregory 1974; Worsey 1984]. The original Coons patch has a quadrilateral topology. However, non-four-sided regions often occur in various scenarios; thus, polygonal patches are required [Várady et al. 2024]. During the past several decades, scholars have proposed various multi-sided transfinite interpolation patches, including triangular patches [Nielson 1979], Charrot–Gregory patches [Charrot and Gregory 1984], Kato's patch [Kato 1991], and ribbon-based patches [Salvi et al. 2014; Várady et al. 2011].

While in the past, the *Computer Aided Geometric Design* community has been mostly concerned with curves and surfaces, more recently, there has been an increasing interest in volumes [Elber 2023; Qin et al. 2024a]. Both hexahedral and tetrahedral Coons volumes have been extensively studied [Hoschek et al. 1993]. However, polyhedral transfinite volumes have rarely been addressed. Randrianarivony [2011] proposed an approach for constructing transfinite interpolation over convex polytopes in arbitrary dimensions. However, in the 3D case, the topology of the volumes is restricted to simple convex polyhedra in which all vertices have three incident faces. Qin et al. [2024b] proposed the C^0 generalized Coons (GC) patch, which generalizes the Coons patch to arbitrary topologies in arbitrary dimensions via generalized barycentric coordinates. However, the C^0 GC patch can only interpolate input boundary curves.

2 Background

2.1 The Coons volume

Given three pairs of quadrilateral boundary surfaces

$$\mathcal{S} = \left\{\mathbf{S}_{\mathbf{u},\mathbf{v},\mathbf{i}}(\xi,\eta), \mathbf{S}_{\mathbf{i},\mathbf{v},\mathbf{w}}(\xi,\eta), \mathbf{S}_{\mathbf{u},\mathbf{i},\mathbf{w}}(\xi,\eta)\right\} \mid (\xi,\eta) \in [0,1]^2\Big\}_{\mathbf{i}\in\{\mathbf{0},\mathbf{1}\}},$$

they form a curved hexahedron $\mathcal{H}$ with twelve boundary curves

$$C = \left\{\mathbf{C}_{\mathbf{u},\mathbf{i},\mathbf{j}}(t), \mathbf{C}_{\mathbf{j},\mathbf{v},\mathbf{i}}(t), \mathbf{C}_{\mathbf{i},\mathbf{j},\mathbf{w}}(t) \mid t \in [0,1]\right\}_{\mathbf{i},\mathbf{j}\in\{\mathbf{0},\mathbf{1}\}}$$

and eight corner points $\mathcal{P} = \left\{\mathbf{p}_{\mathbf{i},\mathbf{j},\mathbf{k}}\right\}_{\mathbf{i},\mathbf{j},\mathbf{k}\in\{\mathbf{0},\mathbf{1}\}}$ (see Fig. 1).

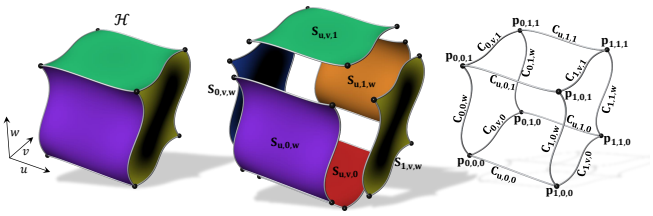


Fig. 1. The input six boundary surfaces.

Then the Coons volume $\mathbf{V}_{Coons}(u,v,w)$ that interpolates all given boundary surfaces is defined as

$$\begin{aligned}\mathbf{V}_{Coons}(u,v,w) = &\left(\mathbf{V}_{\mathbf{u},\mathbf{v}}(u,v,w) + \mathbf{V}_{\mathbf{v},\mathbf{w}}(u,v,w) + \mathbf{V}_{\mathbf{u},\mathbf{w}}(u,v,w)\right)\\ &-\left(\mathbf{V}_{\mathbf{u}}(u,v,w) + \mathbf{V}_{\mathbf{v}}(u,v,w) + \mathbf{V}_{\mathbf{w}}(u,v,w)\right) + \mathbf{V}_{\mathbf{p}}(u,v,w),\end{aligned} \tag{1}$$

where

$$\begin{aligned}\mathbf{V}_{\mathbf{u},\mathbf{v}}(u,v,w) &= (1-w)\mathbf{S}_{\mathbf{u},\mathbf{v},\mathbf{0}}(u,v) + w\mathbf{S}_{\mathbf{u},\mathbf{v},\mathbf{1}}(u,v),\\ \mathbf{V}_{\mathbf{v},\mathbf{w}}(u,v,w) &= (1-u)\mathbf{S}_{\mathbf{0},\mathbf{v},\mathbf{w}}(v,w) + u\mathbf{S}_{\mathbf{1},\mathbf{v},\mathbf{w}}(v,w),\\ \mathbf{V}_{\mathbf{u},\mathbf{w}}(u,v,w) &= (1-v)\mathbf{S}_{\mathbf{u},\mathbf{0},\mathbf{w}}(u,w) + v\mathbf{S}_{\mathbf{u},\mathbf{1},\mathbf{w}}(u,w),\end{aligned} \tag{2}$$

$$\begin{aligned}\mathbf{V}_{\mathbf{u}}(u,v,w) &= (1-w)\left[(1-v)\mathbf{C}_{\mathbf{u},\mathbf{0},\mathbf{0}}(u) + v\mathbf{C}_{\mathbf{u},\mathbf{1},\mathbf{0}}(u)\right]\\ &\quad + w\left[(1-v)\mathbf{C}_{\mathbf{u},\mathbf{0},\mathbf{1}}(u) + v\mathbf{C}_{\mathbf{u},\mathbf{1},\mathbf{1}}(u)\right],\\ \mathbf{V}_{\mathbf{v}}(u,v,w) &= (1-u)\left[(1-w)\mathbf{C}_{\mathbf{0},\mathbf{v},\mathbf{0}}(v) + w\mathbf{C}_{\mathbf{0},\mathbf{v},\mathbf{1}}(v)\right]\\ &\quad + u\left[(1-w)\mathbf{C}_{\mathbf{1},\mathbf{v},\mathbf{0}}(v) + w\mathbf{C}_{\mathbf{1},\mathbf{v},\mathbf{1}}(v)\right],\\ \mathbf{V}_{\mathbf{w}}(u,v,w) &= (1-v)\left[(1-u)\mathbf{C}_{\mathbf{0},\mathbf{0},\mathbf{w}}(w) + u\mathbf{C}_{\mathbf{1},\mathbf{0},\mathbf{w}}(w)\right]\\ &\quad + v\left[(1-u)\mathbf{C}_{\mathbf{0},\mathbf{1},\mathbf{w}}(w) + u\mathbf{C}_{\mathbf{1},\mathbf{1},\mathbf{w}}(w)\right],\end{aligned} \tag{3}$$

$$\begin{aligned}\mathbf{V}_{\mathbf{p}}(u,v,w) = &(1-u)(1-v)(1-w)\mathbf{p}_{\mathbf{0},\mathbf{0},\mathbf{0}} + u(1-v)(1-w)\mathbf{p}_{\mathbf{1},\mathbf{0},\mathbf{0}}\\ &+ uv(1-w)\mathbf{p}_{\mathbf{1},\mathbf{1},\mathbf{0}} + (1-u)v(1-w)\mathbf{p}_{\mathbf{0},\mathbf{1},\mathbf{0}}\\ &+ (1-u)(1-v)w\mathbf{p}_{\mathbf{0},\mathbf{0},\mathbf{1}} + u(1-v)w\mathbf{p}_{\mathbf{1},\mathbf{0},\mathbf{1}}\\ &+ uvw\mathbf{p}_{\mathbf{1},\mathbf{1},\mathbf{1}} + (1-u)vw\mathbf{p}_{\mathbf{0},\mathbf{1},\mathbf{1}},\end{aligned} \tag{4}$$

and $(u,v,w) \in [0,1]^3$. Figure 2 shows the construction process of the Coons volume $\mathbf{V}_{Coons}(u,v,w)$.

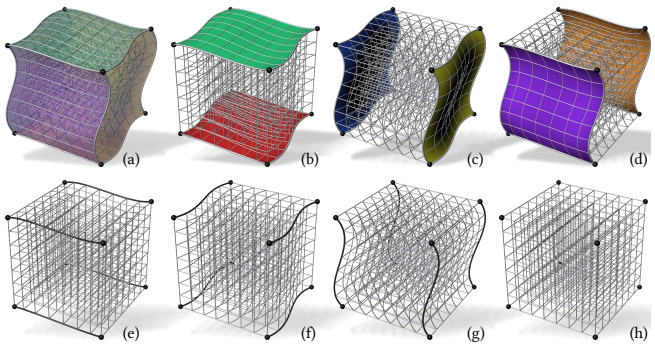


Fig. 2. The volumetric Boolean-sum construction of a Coons volume. (a) The resultant Coons volume. (b) $\mathbf{V}_{\mathbf{u},\mathbf{v}}(u,v,w)$. (c) $\mathbf{V}_{\mathbf{v},\mathbf{w}}(u,v,w)$. (d) $\mathbf{V}_{\mathbf{u},\mathbf{w}}(u,v,w)$. (e) $\mathbf{V}_{\mathbf{u}}(u,v,w)$. (f) $\mathbf{V}_{\mathbf{v}}(u,v,w)$. (g) $\mathbf{V}_{\mathbf{w}}(u,v,w)$. (h) $\mathbf{V}_{\mathbf{p}}(u,v,w)$.

2.2 C^0 generalized Coons patches defined over polyhedra

The C^0 GC patch [Qin et al. 2024a] can be defined over higher-dimensional domains with non-manifold topologies, as long as the adopted generalized barycentric coordinates are well-defined. An n dimensional C^0 GC patch is a mapping from $\mathbb{R}^n$ to $\mathbb{R}^m$ ($m \geq n$). In this paper, we focus on 3D volumes defined over polyhedral domains (i.e., $m = n = 3$), where a C^0 GC "patch" is actually a curved volume.

Given some boundary curves $C = \left\{\mathbf{C}_{i,j}(t) \mid t \in [0,1]\right\}$, they form a curved wireframe $\boldsymbol{\psi}$ with corner points $\mathcal{P} = \left\{\mathbf{p}_k\right\}$, i.e., $\mathbf{C}_{i,j}(0) = \mathbf{p}_i$ and $\mathbf{C}_{i,j}(1) = \mathbf{p}_j$. Ω is a polyhedron that is homeomorphic to $\boldsymbol{\psi}$ and

has vertices $\mathcal{V} = \{\mathbf{v}_k\}$ and edges $\mathcal{E} = \{\mathbf{e}_{i,j}(t) = (1-t)\mathbf{v}_i + t\mathbf{v}_j \mid t \in [0,1]\}$ (see Fig. 3).

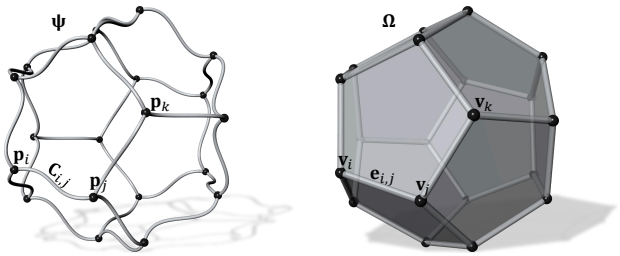


Fig. 3. The input boundary curves and the corresponding domain polyhedron.

Then the C^0 generalized Coons patch [Qin et al. 2024b] $\mathrm{S}_{GC}(\mathbf{x})$ that is defined over Ω and interpolates all given boundary curves is

$$\mathrm{S}_{GC}(\mathbf{x}) = \underbrace{\sum_{\mathbf{e}_{i,j}\in\mathcal{E}} (\lambda_i + \lambda_j)\mathbf{C}_{i,j}\left(\frac{\lambda_j}{\lambda_i+\lambda_j}\right)}_{\text{curve term}} - \underbrace{\sum_{\mathbf{v}_k\in\mathcal{V}} (d_k - 1)\lambda_k \mathbf{p}_k}_{\text{point term}}, \quad (5)$$

where $\mathbf{x} \in \Omega$, $\{\lambda_k\}$ are *non-negative* generalized barycentric coordinates of $\mathbf{x}$ with respect to the vertices $\mathcal{V}$, and d_k is the degree of the vertex $\mathbf{v}_k$. Qin et al. [2024b] pointed out that the above equation has a simple but intuitive explanation: each corner point $\mathbf{p}_k$ lies on d_k curves, so it is calculated d_k times and needs to be subtracted $d_k - 1$ times. In concrete terms, from the properties of barycentric coordinates we know that for a point on edge $\mathbf{e}_{i,j}$ we have $\lambda_i + \lambda_j = 1$ and also $\lambda_k = 0$ for all $k \notin \{i, j\}$, so the curve term of Eq. (5) reduces to $\mathbf{C}_{i,j}(\lambda_j) + (d_i - 1)\lambda_i\mathbf{p}_i + (d_j - 1)\lambda_j\mathbf{p}_j$. The point term cancels all but the first term, thereby ensuring the boundary interpolation property. Figure 4 presents a toy example of a 3D C^0 GC patch.

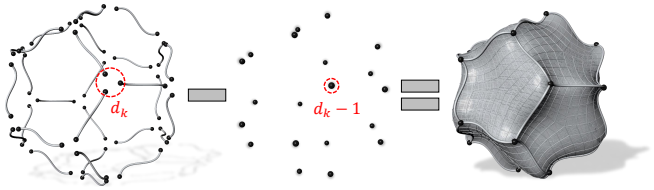


Fig. 4. The curve-based Boolean-sum construction of a 3D C^0 GC patch [Qin et al. 2024b].

Why non-negative GBCs? When negative coordinates occur, it may happen that λ_i and λ_j have opposite signs but similar magnitudes. Then $\left|\frac{\lambda_j}{\lambda_i+\lambda_j}\right|$ can become extremely large, and $\frac{\lambda_j}{\lambda_i+\lambda_j}$ no longer lies in $[0, 1]$. In contrast, if the barycentric coordinates are non-negative, the patch evaluation is not affected even when both λ_i and λ_j tend to zero. Although $\frac{\lambda_j}{\lambda_i+\lambda_j}$ has no limit there, its value remains bounded within $[0, 1]$, and $(\lambda_i + \lambda_j)$ approaches zero, so the corresponding term vanishes.

3 C^0 generalized Coons volumes

The original Coons volume is defined over a canonical parametric domain, which makes it difficult to extend to general polyhedral topologies. Here we follow the strategy of Qin et al. [2024a] and reparameterize it using generalized barycentric coordinates, thereby generalizing it to a general polyhedral domain.

3.1 Reparameterization by trilinear coordinates

We generalize the Coons volume using trilinear coordinates on a convex hexahedron with planar quadrilaterals. Let Ω be such a hexahedron with eight vertices $\mathcal{V} = \{\mathbf{v}_k\}_{k=1}^8$. For each point $\mathbf{x} \in \Omega$, there exist unique $(u, v, w) \in [0, 1]^3$ such that

$$\mathbf{x} = \sum_{k=1}^{8} \lambda_k \mathbf{v}_k, \quad (6)$$

where

$$\begin{aligned} \lambda_1 &= (1-u)(1-v)(1-w), & \lambda_2 &= u(1-v)(1-w), \\ \lambda_3 &= uv(1-w), & \lambda_4 &= (1-u)v(1-w), \\ \lambda_5 &= (1-u)(1-v)w, & \lambda_6 &= u(1-v)w, \\ \lambda_7 &= uvw, & \lambda_8 &= (1-u)vw, \end{aligned} \quad (7)$$

are the trilinear coordinates of point $\mathbf{x}$ with respect to vertices $\{\mathbf{v}_k\}_{k=1}^8$ (see Fig. 5).

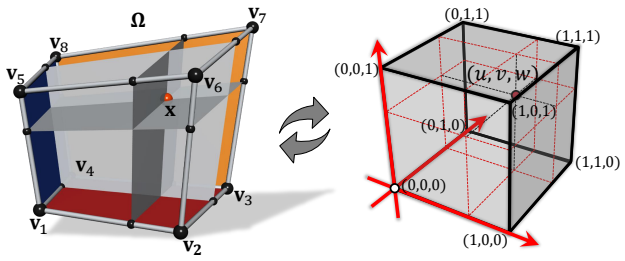


Fig. 5. The domain hexahedron is mapped into a unit cube by trilinear coordinates.

For convenience, here we use some notations to index the edges $\mathcal{E} = \{\mathbf{e}_{\{i,j\}}\}$ and faces $\mathcal{F} = \{\mathbf{f}_{\{i,j,k,l\}}\}$ of the hexahedron Ω, where $\mathbf{e}_{\{i,j\}}$ indicates the edge is composed of two vertices $\mathbf{v}_i$ and $\mathbf{v}_j$, i.e., $\mathbf{e}_{\{i,j\}} = \mathbf{e}_{i,j}(t) = \mathbf{e}_{j,i}(1-t)$, and $\mathbf{f}_{\{i,j,k,l\}}$ implies that the corresponding four vertices span the face. Henceforth, these boundary surfaces, curves, and corner points are all represented using new notations, i.e., $\mathcal{S} = \{\mathrm{S}_{\{i,j,k,l\}}\}$, $\mathcal{C} = \{\mathbf{C}_{\{i,j\}}\}$, and $\mathcal{P} = \{\mathbf{p}_k\}_{k=1}^8$. For example, $\mathrm{S}_{\{1,4,3,2\}} = \mathrm{S}_{\mathbf{u},\mathbf{v},\mathbf{0}}$, $\mathbf{C}_{\{1,2\}} = \mathbf{C}_{\mathbf{u},\mathbf{0},\mathbf{0}}$, and $\mathbf{p}_1 = \mathbf{p}_{\mathbf{0},\mathbf{0},\mathbf{0}}$ (see Fig. 6).

The parameters (u, v, w) of a Coons volume $\mathbf{V}_{Coons}(u, v, w)$ can be represented by the trilinear coordinates. Then we can rewrite the Coons volume formula by trilinear coordinates. For simplicity's sake, we consider only three terms from Eqs. (2)–(4): $(1-w)\mathrm{S}_{\mathbf{u},\mathbf{v},\mathbf{0}}(u, v)$, $(1-w)(1-v)\mathbf{C}_{\mathbf{u},\mathbf{0},\mathbf{0}}(u)$, and $(1-u)(1-v)(1-w)\mathbf{p}_{\mathbf{0},\mathbf{0},\mathbf{0}}$. First, it is obvious that

$$(1-u)(1-v)(1-w)\mathbf{p}_{\mathbf{0},\mathbf{0},\mathbf{0}} = \lambda_1\mathbf{p}_1, \quad (8)$$

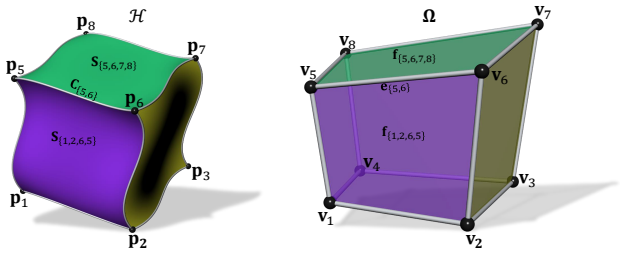


Fig. 6. The input boundary surfaces and the corresponding domain hexahedron.

$$
\begin{aligned}
(1-w)(1-v)\mathbf{C}_{\mathbf{u},0,0}(u) &= (\lambda_1+\lambda_2)\mathbf{C}_{\{1,2\}}\left(\frac{\lambda_1}{\lambda_1+\lambda_2},\frac{\lambda_2}{\lambda_1+\lambda_2}\right)\\
&= (\lambda_1+\lambda_2)\mathbf{C}_{1,2}\left(\frac{\lambda_2}{\lambda_1+\lambda_2}\right) = (\lambda_1+\lambda_2)\mathbf{C}_{2,1}\left(\frac{\lambda_1}{\lambda_1+\lambda_2}\right).
\end{aligned}
\tag{9}
$$

It can be seen that the above equations are consistent in form with the formula for the C^0 GC patch (Eq. 5).

Next, we consider the last term $(1-w)\mathbf{S}_{\mathbf{u},\mathbf{v},0}(u,v)$. We would like to rewrite this term in a form similar to the two terms above; however, this is non-trivial. Since $\mathbf{S}_{\mathbf{u},\mathbf{v},0}(\xi,\eta)$ is a bi-parametric patch defined over the unit square, it can be reparameterized via bilinear coordinates of a convex quadrilateral. To maintain consistency in form, we choose $\mathbf{f}_{\{1,4,3,2\}}$ as the domain quadrilateral. Then $\mathbf{S}_{\mathbf{u},\mathbf{v},0}(\xi,\eta)$ can be expressed as

$$
\begin{aligned}
\mathbf{S}_{\mathbf{u},\mathbf{v},0}(\xi,\eta) &= \mathbf{S}_{\mathbf{u},\mathbf{v},0}(\xi(\hat{\lambda}_1,\hat{\lambda}_4,\hat{\lambda}_3,\hat{\lambda}_2),\eta(\hat{\lambda}_1,\hat{\lambda}_4,\hat{\lambda}_3,\hat{\lambda}_2))\\
&= \mathbf{S}_{\mathbf{u},\mathbf{v},0}(\hat{\lambda}_1,\hat{\lambda}_4,\hat{\lambda}_3,\hat{\lambda}_2),
\end{aligned}
\tag{10}
$$

where $\hat{\lambda}_1,\hat{\lambda}_4,\hat{\lambda}_3,\hat{\lambda}_2$ are bilinear coordinates of an arbitrary point inside $\mathbf{f}_{1,4,3,2}$. Here, we introduce four face local parameters

$$
\left\{\frac{\lambda_i}{\sum_{j\in\{1,4,3,2\}}\lambda_j}\right\}_{i\in\{1,4,3,2\}},
$$

which are similar to those of the generalized Bézier volume [Qin et al. 2024a]. When the point lies on $\mathbf{f}_{\{1,4,3,2\}}$, the following equation holds.

$$
\frac{\lambda_i}{\sum_{j\in\{1,4,3,2\}}\lambda_j} = \hat{\lambda}_i, \quad i\in\{1,4,3,2\}. \tag{11}
$$

In fact, due to the construction of trilinear coordinates (Eq. 7), the above four face local parameters are essentially the bilinear coordinates of the point $\mathbf{x}$ with respect to a cross-sectional quadrilateral, whose four vertices are $(1-w)\left[\mathbf{v}_1,\mathbf{v}_4,\mathbf{v}_3,\mathbf{v}_2\right] + w\left[\mathbf{v}_5,\mathbf{v}_8,\mathbf{v}_7,\mathbf{v}_6\right]$. Thus, the last term can be written in the following form.

$$
\begin{aligned}
&(1-w)\mathbf{S}_{\mathbf{u},\mathbf{v},0}(u,v) =\\
&\left(\sum_{j\in\{1,4,3,2\}}\lambda_j\right)\mathbf{S}_{\{1,4,3,2\}}\left(\left\{\frac{\lambda_i}{\sum_{j\in\{1,4,3,2\}}\lambda_j}\right\}_{i\in\{1,4,3,2\}}\right).
\end{aligned}
\tag{12}
$$

Analogously, the remaining terms in Eqs. (2)–(4) can all be expressed by trilinear coordinates in similar forms. Substituting these terms back into Eq. (1), we obtain the following new expression of the Coons volume.

$$
\begin{aligned}
\mathbf{V}_{Coons}(\mathbf{x}) = &\underbrace{\sum_{\mathbf{f}_{\{i,j,k,l\}}\in\mathcal{F}}\left[\left(\sum_{p\in\{i,j,k,l\}}\lambda_p\right)\mathbf{S}_{\{i,j,k,l\}}\left(\left\{\frac{\lambda_q}{\sum_{p\in\{i,j,k,l\}}\lambda_p}\right\}_{q\in\{i,j,k,l\}}\right)\right]}_{\text{surface term}}\\
&\underbrace{-\sum_{\mathbf{e}_{\{i,j\}}\in\mathcal{E}}(\lambda_i+\lambda_j)\mathbf{C}_{\{i,j\}}\left(\frac{\lambda_i}{\lambda_i+\lambda_j},\frac{\lambda_j}{\lambda_i+\lambda_j}\right)}_{\text{curve term}} + \underbrace{\sum_{\mathbf{v}_k\in\mathcal{V}}\lambda_k\mathbf{p}_k}_{\text{point term}}.
\end{aligned}
\tag{13}
$$

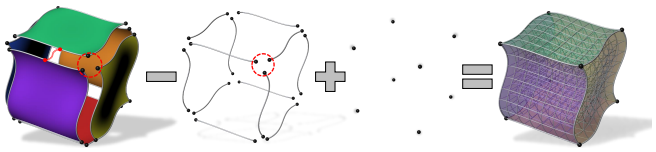


Fig. 7. The new surface-based Boolean-sum construction of the hexahedral Coons volume.

The above equation has an intuitive explanation similar to that of the C^0 GC patch [Qin et al. 2024b]. In the surface term, each curve is calculated twice, and each corner point is calculated three times. Therefore, each curve needs to be subtracted once, while each corner point needs to be subtracted twice. In the curve term, however, each curve is subtracted once, while each corner point is subtracted three times. Therefore, each corner point must be added once in the point term (see Fig. 7).

3.2 Polyhedral generalization

The above new expression of the Coons volume can be generalized to general polyhedral topologies via generalized barycentric coordinates defined over polyhedra in a straightforward way. Given some boundary surfaces $\mathcal{S} = \{\mathbf{S}_\mathbf{I}\}$ ($\mathbf{I} = \{i_1, i_2, ..., i_n\}$), they form a curved polyhedron $\boldsymbol{\psi}$ with boundary curves $\mathcal{C} = \{\mathbf{C}_{\{i,j\}}\}$ and corner points $\mathcal{P} = \{\mathbf{p}_k\}$. Ω is a polyhedron that is homeomorphic to $\boldsymbol{\psi}$ and has vertices $\mathcal{V} = \{\mathbf{v}_k\}$, edges $\mathcal{E} = \{\mathbf{e}_{\{i,j\}}\}$, and faces $\mathcal{F} = \{\mathbf{f}_\mathbf{I}\}$. Note that each $\mathbf{S}_\mathbf{I}$ is a multi-sided surface with an arbitrary number of sides n, whose parameter domain is the corresponding n-sided polygon $\mathbf{f}_\mathbf{I}$ with vertices $\{\mathbf{v}_{i_1}, \mathbf{v}_{i_2}, ..., \mathbf{v}_{i_n}\}$ (see Fig. 8).

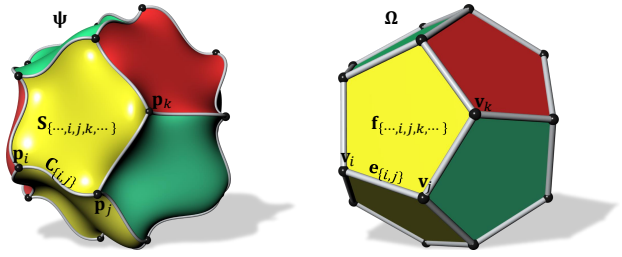


Fig. 8. The input boundary surfaces and the corresponding domain polyhedron.

Then the C^0 generalized Coons volume is defined as

$$\mathbf{V}_{GC}(\mathbf{x}) = \underbrace{\sum_{\mathbf{f_I}\in\mathcal{F}}\left[\left(\sum_{j\in\mathbf{I}}\lambda_j\right)\mathbf{S_I}\left(\left\{\frac{\lambda_i}{\sum_{j\in\mathbf{I}}\lambda_j}\right\}_{i\in\mathbf{I}}\right)\right]}_{\text{surface term}} - \underbrace{\sum_{\mathbf{e}_{\{i,j\}}\in\mathcal{E}}(\lambda_i+\lambda_j)\mathbf{C}_{\{i,j\}}\left(\frac{\lambda_i}{\lambda_i+\lambda_j},\frac{\lambda_j}{\lambda_i+\lambda_j}\right)}_{\text{curve term}} + \underbrace{\sum_{\mathbf{v}_k\in\mathcal{V}}\lambda_k\mathbf{p}_k}_{\text{point term}}, \tag{14}$$

where $\mathbf{x}\in\Omega$ and $\{\lambda_k\}$ are *non-negative* generalized barycentric coordinates of point $\mathbf{x}$ with respect to the vertices $\mathcal{V}$. Figure 9 shows the construction process of the C^0 GC volume.

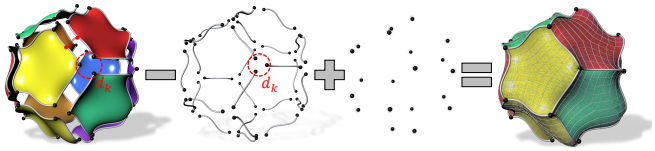


Fig. 9. The surface-based Boolean-sum construction of our C^0 GC volume.

3.3 Properties

Theorem 1. **(Reduction to the Coons volume)** *When the domain polyhedron is a convex hexahedron with planar quadrilaterals, the C^0 GC volume with trilinear coordinates is reduced to a Coons volume.*

Proof. Trivial. According to Section 3.1, the conclusion is obvious. □

Theorem 2. **(Interpolation)** *The C^0 GC volume interpolates all given boundary surfaces.*

Theorem 3. **(Reduction to the C^0 GC patch)** *When all the given boundary surfaces are 2D C^0 GC patches, i.e.,*

$$\mathbf{S_I}(\hat{\mathbf{x}}) = \sum_{\mathbf{e}_{\{i,j\}}\subset\mathbf{f_I}}(\hat{\lambda}_i+\hat{\lambda}_j)\mathbf{C}_{\{i,j\}}\left(\frac{\hat{\lambda}_i}{\hat{\lambda}_i+\hat{\lambda}_j},\frac{\hat{\lambda}_j}{\hat{\lambda}_i+\hat{\lambda}_j}\right) - \sum_{k\in\mathbf{I}}\hat{\lambda}_k\mathbf{p}_k, \tag{15}$$

where $\hat{\mathbf{x}}\in\mathbf{f_I}$ and $\{\hat{\lambda}_i\}$ are 2D generalized barycentric coordinates of point $\hat{\mathbf{x}}$ with respect to the vertices of $\mathbf{f_I}$. Then the C^0 GC volume is reduced to a 3D C^0 GC patch.

The proofs of Theorems 2 and 3 can be found in the supplementary material.

Corollary 1. **(Generalized barycentric reproduction)** *If all the given boundary surfaces are defined by 2D generalized barycentric interpolation, i.e.,*

$$\mathbf{S_I}(\hat{\mathbf{x}}) = \sum_{i\in\mathbf{I}}\hat{\lambda}_i\mathbf{p}_i, \tag{16}$$

where $\hat{\mathbf{x}}\in\mathbf{f_I}$ and $\{\hat{\lambda}_i\}$ are 2D generalized barycentric coordinates of point $\hat{\mathbf{x}}$ with respect to the vertices of $\mathbf{f_I}$. Then the C^0 GC volume reproduces 3D generalized barycentric interpolation, i.e.,

$$\mathbf{V}_{GC}(\mathbf{x}) = \sum_{\mathbf{v}_k\in\mathcal{V}}\lambda_k\mathbf{p}_k. \tag{17}$$

Proof. Trivial. This is because Eq. (16) is a special case of Eq. (15) and the C^0 GC patch satisfies the generalized barycentric reproduction property [Qin et al. 2024b]. □

4 Results

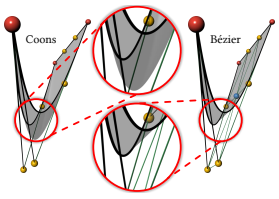
4.1 Sphere interpolation over Platonic solids

The sphere is perhaps one of the most fundamental geometric shapes, widely used in both organic and human-made objects. An interesting topic is the accurate representation and approximation of sphere surfaces using parametric patches [Dedoncker et al. 2018; Grošelj and Šadl Praprotnik 2022; Vavpetič and Žagar 2022]. Here, we employ the proposed C^0 GC volumes to represent a sphere solid by interpolating its boundary surfaces (see Fig. 13). The five Platonic solids are utilized as the parameter domain polyhedra. For the hexahedron and dodecahedron, we employed Wachspress coordinates [Wachspress 1975; Warren 1996], which are defined for simple convex polyhedra (in which all vertices have three incident faces), while for the octahedron and icosahedron, we used mean value coordinates [Floater et al. 2005; Ju et al. 2005], which are defined on triangular meshes. For comparison, the C^0 GC patches with the same input boundary curves and domain polyhedra are also presented. It should be noted that since the C^0 GC patch can only interpolate the boundary curves, it is incapable of achieving an accurate representation of a sphere.

4.2 3D high-order cage-based deformation

Cage-based deformation [Ju et al. 2005] is a simple but powerful method for shape editing. Early work focused on linear cage-based deformation, where both the source and target cages are linear, while recent work has begun to focus on high-order cage-based deformation. 2D high-order cage coordinates [Li et al. 2013; Lin and Chen 2024; Liu et al. 2025a,b; Michel et al. 2025; Michel and Thiery 2023] have been extensively studied. However, only a few works [Xiao and Chen 2025] have addressed 3D high-order cage coordinates.

Qin et al. [2024b] pointed out that the C^0 GC patch satisfies the generalized barycentric reproduction property, making it suitable for 3D high-order cage-based deformation. Our C^0 GC volume also possesses this property and can therefore be applied as well. The deformation process comprises two main steps: 1) cage degree elevation, and 2) manipulating control points (see Fig. 10). Note that while the C^0 GC patch only performs degree elevation on cage edges, our method applies it to cage faces.



Comparison with the C^0 GC patch. Figure 16 provides a comparison between our C^0 GC volume and the C^0 GC patch. Harmonic coordinates [Joshi et al. 2007] for triangular cages were adopted for evaluation. The C^0 GC patch maps each linear edge to a cubic Bézier curve, while each linear triangle is automatically mapped to a triangular Coons patch [Marshall 1975; Nielson 1979]. In contrast, our C^0 GC volume maps each linear triangle to a cubic triangular Bézier patch. Since Coons

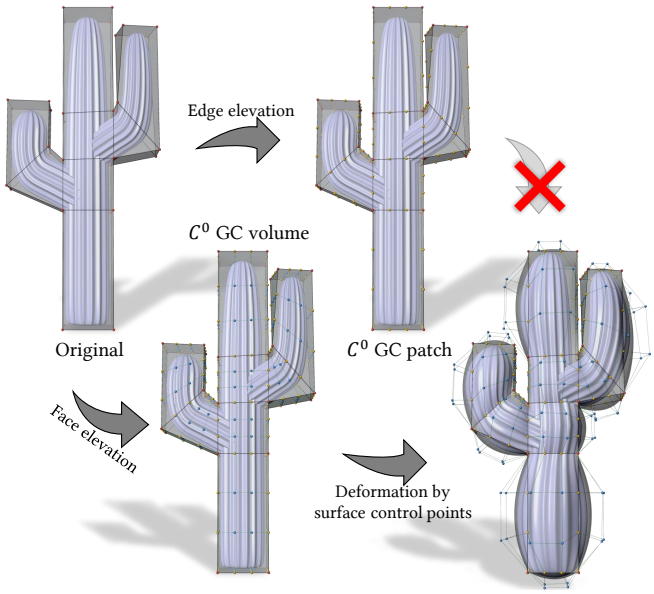


Fig. 10. The top-right image shows the reproduction result of the C^0 GC patch [Qin et al. 2024b] using only cubic Bézier curves. The bottom-left image shows the reproduction result of our C^0 GC volume using bicubic Bézier patches. Both results are identical to the original Cactus model. When the edges of the cage are fixed, the C^0 GC volume still possesses additional degrees of freedom, namely the internal control points of the patches, to achieve deformation, whereas the C^0 GC patch does not.

patches lack internal control, they may produce undesired high-curvature shapes in the interior of the surface. In contrast, Bézier patches can avoid this issue by utilizing internal control points (see the inset). Thus, when used for deformation, the C^0 GC patch is prone to causing unnecessary distortions (green circle in Fig. 16), sharp artifacts (rectangles in Fig. 16), or large extrusion beyond the cage boundaries (red circle in Fig. 16). Conversely, the C^0 GC volume substantially reduces artifacts, thus significantly improving the deformation results.

Tri cage vs. Triquad cage. As noted in prior work [Thiery and Boubekeur 2022; Thiery et al. 2018], quad or triquad meshes are more prevalent than pure triangle meshes in industry. Quadrilaterals are not only more intuitive for artists in cage modeling but also better at aligning with models' anisotropic features. Thiery and their co-authors established that in linear cage deformation, using a pure triangle cage often results in unpredictable distortions, whereas employing a suitable triquad cage can avoid such artifacts. We find that this issue not only remains but is significantly amplified under high-order cage-based deformation. Figure 14 compares the deformation of an octopus model using a tri cage and a triquad cage. In this example (as well as in Figs. 10 & 15), we also employ harmonic coordinates [Joshi et al. 2007] for volume evaluation. Following the strategy of Thiery et al. [2018], we treat each (non-planar) quad face as a bilinear patch. Correspondingly, each linear triangle or quadrilateral is deformed into a cubic triangular or tensor-product Bézier patch, respectively. As shown in the figure, the deformation result obtained using the tri cage exhibits not only unexpected distortions (green rectangles) but also extrusion beyond the cage boundary (red rectangles). Conversely, using a triquad cage effectively eliminates these artifacts.

Comparison with state-of-the-art. Figure 15 presents a comparison with Bézier Green coordinates [Xiao and Chen 2025], which are a high-order version of 3D Green coordinates [Lipman et al. 2008]. The figure shows that while the Bézier Green coordinates deformation better preserves local details, it fails to maintain alignment with the control cage (red arrow). In contrast, although the C^0 GC volume introduces shearing artifacts (red circle), it aligns perfectly with the cage. This is due to the well-known fact that conformality and interpolation are inherently conflicting properties. We also provide the linear cage deformation result obtained using a refined cage, which contains roughly nine times more vertices and faces than the original coarse cage. Nevertheless, the result of the linear method is not competitive with that of the high-order method, as it is visually non-smooth.

4.3 Interpolation over PolyCubes

PolyCubes [Tarini et al. 2004] are special 3-manifold polyhedra whose faces are aligned with the coordinate planes. They have been widely used in seamless texture mapping [Tarini et al. 2004], spline fitting [Wang et al. 2008], iso-geometric analysis [Al Akhras et al. 2017], and quadrilateral/hexahedral mesh generation [Bommes et al. 2013; Guo et al. 2020]. PolyCube-Maps are special surface mappings that parameterize an input 2-manifold triangle mesh onto the surface of a PolyCube.

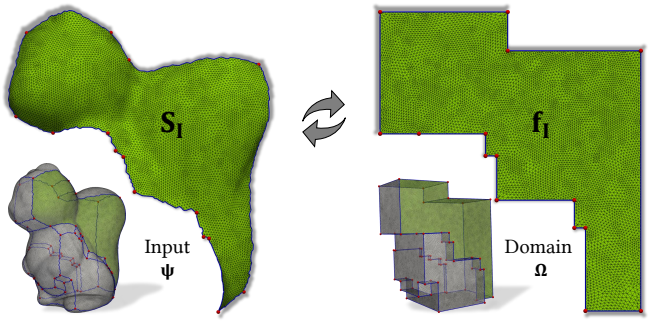


Fig. 11. **PolyCube-Map**: the input triangle mesh (left) is parameterized onto the boundary of a PolyCube (right).

We treat the input triangle mesh as the boundary surface patches and the corresponding PolyCube as the parametric domain, thereby constructing an interpolating C^0 GC volume. Each surface patch is regarded as a C^0 continuous piecewise linear function defined on a polygonal face of the PolyCube (see Fig. 11). Here, we adopt the method proposed by Yang et al. [2019] to compute the PolyCube-Map. Harmonic coordinates [Joshi et al. 2007] are still chosen as the parameters. Since all the faces of a PolyCube are planar polygons, the computation of harmonic coordinates is similar to that for a triangle mesh.

Table 1. Running times of different approaches (in seconds). #Q: number of query points. #V/#E/#F: number of cage (domain polyhedron) vertices/edges/faces. GCP/GCV: the C^0 generalized Coons patches/volumes.

Model	#Q	#V	#E	#F	GCP	GCV
Icosahedron	80711	12	30	20	0.927s	1.467s
Dodecahedron	89793	20	30	12	1.226s	1.522s
Horse	48485	51	147	98	2.905s	5.676s
Shark	14976	24	66	44	0.428s	0.814s
Cactus	98820	36	68	34	2.876s	5.814s

Figure 17 presents a gallery of the interpolating C^0 GC volumes over PolyCubes. As can be seen from the figure, although the topologies are highly complex—these examples are concave and even contain holes—our method still works well. As expected, the resulting C^0 GC volumes interpolate all the prescribed boundary surfaces, regardless of their complexity.

4.4 Running times

We report in Table 1 the timings for the computations of the C^0 GC patches and volumes. As expected, the runtime of C^0 GC volumes is roughly twice that of C^0 GC patches. This is because computing C^0 GC patches includes only the curve and point terms (Eq. 5), whereas computing C^0 GC volumes additionally includes the surface term (Eq. 14). Additionally, we would like to emphasize that our current implementation is not performance-tuned. Since our method has an explicit formula, the running times could be significantly reduced by leveraging parallelization.

The above running times do not include the computation of the barycentric coordinates. For concave polyhedra, since we use harmonic coordinates, which do not have a closed-form expression, we have to compute them numerically via discretization. In our examples, the precomputation of harmonic coordinates is quite time-consuming, typically taking more than ten minutes per model.[2]

All experiments are conducted on a laptop equipped with an Intel Core i7-10750H at 2.60 GHz. The implementations are carried out in `C++` along with `TetGen` [Si 2015] and `libigl` [Jacobson et al. 2018] libraries on Windows. More implementation details can be found in the supplementary material.

4.5 Discussion

Figure 18 shows the scaled Jacobians of the constructed C^0 GC volumes. As interpolation quality is not explicitly considered, the resulting C^0 GC volumes may exhibit fold-over, manifested as negative Jacobians (highlighted in red in Fig. 18). *The shape of the parametric domain may play an important role in the quality of the interpolation volume.*

Prior work by Várady et al. [2011, 2024] on transfinite surfaces points out that the parametric domain polygon should mimic the shape of the input curve configuration as closely as possible, in order to reduce distortion near the corners. Figure 12(a) presents a typical example. For a concave curve configuration, using a square as the parametric domain results in folding near the concave corner, whereas using a concave quadrilateral as the parametric domain avoids this issue.

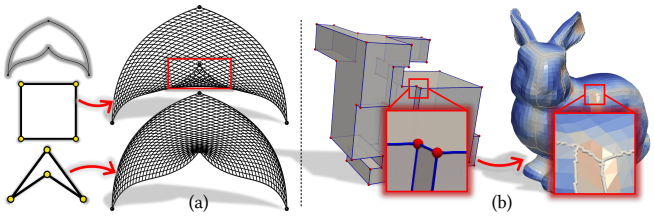


Fig. 12. Influence of parametric domain shape.

The situation is similar for transfinite interpolation volumes. In spherical interpolation examples, using a tetrahedron as the parametric domain results in negative Jacobians along the boundary curves. In contrast, using other regular polyhedra that better mimic the spherical shape as the parametric domain significantly reduces this problem (see Fig. 18). Likewise, negative Jacobians often appear near the boundary curves and corners in PolyCube interpolation, due to PolyCubes not adequately mimicking the shapes of input surface configurations. Figure 12(b) shows a typical scenario: although the local region on the back of the input bunny model is relatively flat, the parametrized PolyCube surface contains two corner points where adjacent faces are perpendicular to each other.

How to construct an appropriate parametric domain polyhedron is a question worth investigating. Cage generation [Calderon and Boubekeur 2017; Sacht et al. 2015] could be regarded as work related to this topic.

5 Conclusion

We have introduced the C^0 generalized Coons (GC) volume, which is a polyhedral generalization of the well-known Coons volume. By using non-negative generalized barycentric coordinates as parameters, the C^0 GC volume can be defined over arbitrary polyhedral domains and interpolate any given boundary surfaces. We have also proven that the C^0 GC volume possesses favorable properties such as interpolation, reduction to the Coons volume and C^0 GC patch, and generalized barycentric reproduction. Furthermore, we have demonstrated applications of the C^0 GC volume in computer graphics, such as sphere interpolation, 3D high-order cage-based deformation, and volumetric PolyCube-Map.

The Coons approach is still alive! We hope that the present work will contribute to the theoretical study of Coons approaches and promote their applications in computer graphics.

Limitations & Future work. Firstly, C^0 GC volumes are restricted to 3-manifold polyhedra. Extending C^0 GC volumes to higher dimensional non-manifold domains is theoretically important but nontrivial, and we leave it for future work. Secondly, the current GC volume construction provides only C^0 interpolation; however, Hermite transfinite interpolation over arbitrary polyhedra is still a "terra incognita". Finally, since interpolation quality is not explicitly considered, the resulting C^0 GC volumes may suffer from fold-over,

[2]To obtain highly accurate results, we used high-resolution tetrahedral meshes to compute harmonic coordinates, resulting in a high computational time. For example, for the Horse model, we solved it on a tetrahedral mesh with more than 391k vertices.

which may limit their applications. Improving interpolation quality—both theoretically and in practice—remains an important topic for further study.

Acknowledgments

The authors would like to thank the anonymous reviewers for their constructive suggestions and comments. Thanks also to Dr. Dong Xiao, Shibo Liu and Qi Zhang from University of Science and Technology of China, for their generous sharing of models and high-order cages. Kaikai Qin, Zeqi Ge, Chenhao Ying and Chongyang Deng were supported by the National Natural Science Foundation of China (NSFC) under the grant number 61872121. Péter Salvi was supported by National Research Development and Innovation Office (NKFI) under the award number 145970. Huibiao Wen and Shiqing Xin were supported by the National Natural Science Foundation of China (NSFC) under the grant number 62272277.

References

H. Al Akhras, T. Elguedj, A. Gravouil, and M. Rochette. 2017. Towards an automatic isogeometric analysis suitable trivariate models generation—Application to geometric parametric analysis. *Computer Methods in Applied Mechanics and Engineering* 316 (2017), 623–645. Special Issue on Isogeometric Analysis: Progress and Challenges.

David Bommes, Bruno Lévy, Nico Pietroni, Enrico Puppo, Claudio Silva, Marco Tarini, and Denis Zorin. 2013. Quad-Mesh Generation and Processing: A Survey. *Computer Graphics Forum* 32, 6 (2013), 51–76.

Stéphane Calderon and Tamy Boubekeur. 2017. Bounding proxies for shape approximation. *ACM Trans. Graph.* 36, 4, Article 57 (July 2017), 13 pages.

Q. Chang, C. Deng, and K. Hormann. 2023. Maximum Likelihood Coordinates. *Computer Graphics Forum* 42, 5 (2023), e14908.

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.

G. Cherchi and M. Livesu. 2023. VOLMAP: a Large Scale Benchmark for Volume Mappings to Simple Base Domains. *Computer Graphics Forum* 42, 5 (2023), e14915.

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

Sander Dedoncker, Laurens Coox, Florian Maurin, Francesco Greco, and Wim Desmet. 2018. Bézier tilings of the sphere and their applications in benchmarking multipatch isogeometric methods. *Computer Methods in Applied Mechanics and Engineering* 332 (2018), 255–279.

Chongyang Deng, Qingjun Chang, and Kai Hormann. 2020. Iterative coordinates. *Computer Aided Geometric Design* 79 (2020), 101861.

Gershon Elber. 2023. A Review of a B-spline based Volumetric Representation: Design, Analysis and Fabrication of Porous and/or Heterogeneous Geometries. *Computer-Aided Design* 163 (2023), 103587.

Michael S. Floater. 2003. Mean value coordinates. *Computer Aided Geometric Design* 20, 1 (2003), 19–27.

Michael S. Floater, Géza Kós, and Martin Reimers. 2005. Mean value coordinates in 3D. *Computer Aided Geometric Design* 22, 7 (2005), 623–631. Geometric Modelling and Differential Geometry.

Xiao-Ming Fu, Chong-Yang Bai, and Yang Liu. 2016. Efficient Volumetric PolyCube-Map Construction. *Computer Graphics Forum* 35, 7 (2016), 97–106.

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.

Jan Grošelj and Ada Šadl Praprotnik. 2022. Exact sphere representations over Platonic solids based on rational multisided Bézier patches. *Computer Aided Geometric Design* 98 (2022), 102148.

Hao-Xiang Guo, Xiaohan Liu, Dong-Ming Yan, and Yang Liu. 2020. Cut-enhanced PolyCube-maps for feature-aware all-hex meshing. *ACM Trans. Graph.* 39, 4, Article 106 (Aug. 2020), 14 pages.

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.

Josef Hoschek, Dieter Lasser, and Larry L. Schumaker. 1993. *Fundamentals of Computer Aided Geometric Design.* A. K. Peters, Ltd., USA.

Alec Jacobson, Daniele Panozzo, et al. 2018. libigl: A simple C++ geometry processing library. https://libigl.github.io/.

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.

Tao Ju, Scott Schaefer, and Joe Warren. 2005. Mean value coordinates for closed triangular meshes. *ACM Trans. Graph.* 24, 3 (July 2005), 561–566.

K. Kato. 1991. Generation of N-sided surface patches with holes. *Computer-Aided Design* 23, 10 (1991), 676–683.

Lingxiao Li, Paul Zhang, Dmitriy Smirnov, S. Mazdak Abulnaga, and Justin Solomon. 2021. Interactive all-hex meshing via cuboid decomposition. *ACM Trans. Graph.* 40, 6, Article 256 (Dec. 2021), 17 pages.

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.

Zhehui Lin and Renjie Chen. 2024. Polynomial Cauchy Coordinates for Curved Cages. In *SIGGRAPH Asia 2024 Conference Papers* (Tokyo, Japan) *(SA '24)*. Association for Computing Machinery, New York, NY, USA, Article 67, 8 pages.

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. *ACM Trans. Graph.* 27, 3 (Aug. 2008), 1–10.

Shibo Liu, Tielin Dai, Ligang Liu, and Xiao-Ming Fu. 2025a. Polynomial 2D Biharmonic Coordinates for High-order Cages. *ACM Trans. Graph.* 44, 4, Article 77 (July 2025), 10 pages.

Shibo Liu, Ligang Liu, and Xiao-Ming Fu. 2025b. Closed-form Cauchy Coordinates and Their Derivatives for 2D High-order Cages. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers (SA Conference Papers '25)*. Association for Computing Machinery, New York, NY, USA, Article 188, 11 pages.

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

Élie Michel, Alec Jacobson, Siddhartha Chaudhuri, and Jean-Marc Thiery. 2025. Variational Green and Biharmonic Coordinates for 2D Polynomial Cages. *ACM Trans. Graph.* 44, 4, Article 76 (July 2025), 20 pages.

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.

Gregory M. Nielson. 1979. The side-vertex method for interpolation in triangles. *Journal of Approximation Theory* 25, 4 (1979), 318–336.

Kaikai Qin, Yajuan Li, and Chongyang Deng. 2024a. Generalized Bézier volumes over simple convex polyhedra. *Comput. Aided Geom. Des.* 111, C (June 2024), 24 pages.

Kaikai Qin, Yunhao Zhou, Chenhao Ying, Yajuan Li, and Chongyang Deng. 2024b. C^0 Generalized Coons Patches for High-order Cage-based Deformation. *ACM Trans. Graph.* 43, 6, Article 220 (Nov. 2024), 15 pages.

Maharavo Randrianarivony. 2011. On transfinite interpolations with respect to convex domains. *Computer Aided Geometric Design* 28, 2 (2011), 135–149.

Leonardo Sacht, Etienne Vouga, and Alec Jacobson. 2015. Nested cages. *ACM Trans. Graph.* 34, 6, Article 170 (Nov. 2015), 14 pages.

Péter Salvi, Tamás Várady, and Alyn Rockwood. 2014. Ribbon-based transfinite surfaces. *Computer Aided Geometric Design* 31, 9 (2014), 613–630.

Hang Si. 2015. TetGen, a Delaunay-Based Quality Tetrahedral Mesh Generator. *ACM Trans. Math. Softw.* 41, 2, Article 11 (Feb. 2015), 36 pages.

Jian-Ping Su, Xiao-Ming Fu, and Ligang Liu. 2019. Practical Foldover-Free Volumetric Mapping Construction. *Computer Graphics Forum* 38, 7 (2019), 287–297.

Kenshi Takayama, Makoto Okabe, Takashi Ijiri, and Takeo Igarashi. 2008. Lapped solid textures: filling a model with anisotropic textures. *ACM Trans. Graph.* 27, 3 (Aug. 2008), 1–9.

Marco Tarini, Kai Hormann, Paolo Cignoni, and Claudio Montani. 2004. PolyCube-Maps. *ACM Trans. Graph.* 23, 3 (Aug. 2004), 853–860.

Jean-Marc Thiery and Tamy Boubekeur. 2022. Green Coordinates for Triquad Cages in 3D. In *SIGGRAPH Asia 2022 Conference Papers* (Daegu, Republic of Korea) *(SA '22)*. Association for Computing Machinery, New York, NY, USA, Article 38, 8 pages.

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.

Aleš Vavpetič and Emil Žagar. 2022. Geometric approximation of the sphere by triangular polynomial spline patches. *Computer Aided Geometric Design* 92 (2022), 102061.

Tamás Várady, Alyn Rockwood, and Péter Salvi. 2011. Transfinite surface interpolation over irregular n-sided domains. *Computer-Aided Design* 43, 11 (2011), 1330–1340. Solid and Physical Modeling 2011.

Tamás Várady, Péter Salvi, and Márton Vaitkus. 2024. Genuine multi-sided parametric surface patches – A survey. *Computer Aided Geometric Design* 110 (2024), 102286.

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

Hongyu Wang, Ying He, Xin Li, Xianfeng Gu, and Hong Qin. 2008. Polycube splines. *Computer-Aided Design* 40, 6 (2008), 721–733. Selected Papers from the ACM Solid and Physical Modeling Symposium 2007.

Joe Warren. 1996. Barycentric coordinates for convex polytopes. *Advances in Computational Mathematics* 6, 1 (1996), 97–108.

Ofir Weber, Mirela Ben-Chen, and Craig Gotsman. 2009. Complex Barycentric Coordinates with Applications to Planar Shape DeformationAbstract. *Computer Graphics Forum* 28, 2 (2009), 587–597.
A.J. Worsey. 1984. A modified C2 Coons' patch. *Computer Aided Geometric Design* 1, 4 (1984), 357–360.
Dong Xiao and Renjie Chen. 2025. Flexible 3D Cage-based Deformation via Green Coordinates on Bézier Patches *(SIGGRAPH Conference Papers '25)*. Association for Computing Machinery, New York, NY, USA, Article 58, 10 pages.
Yang Yang, Xiao-Ming Fu, and Ligang Liu. 2019. Computing Surface PolyCube-Maps by Constrained Voxelization. *Computer Graphics Forum* 38, 7 (2019), 299–309.

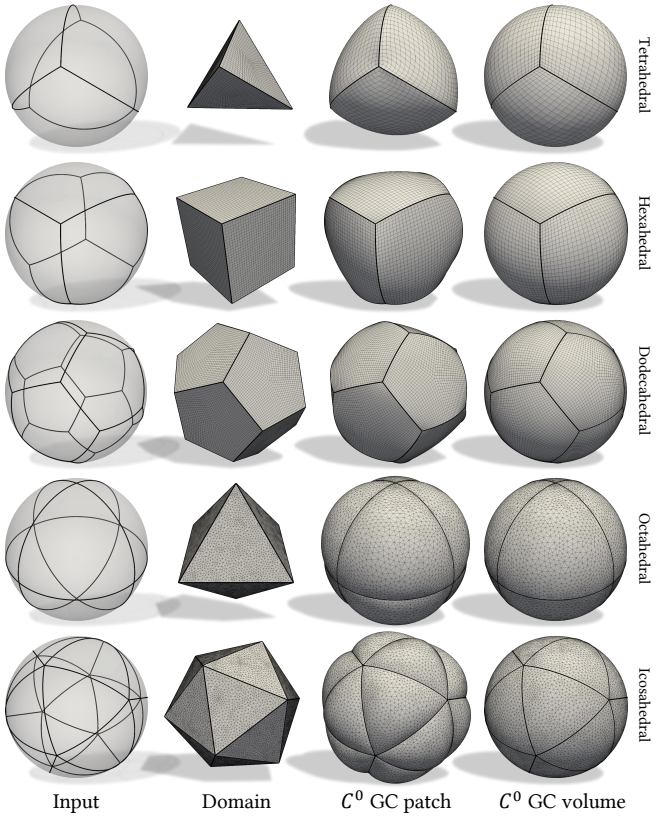


Fig. 13. **Sphere interpolation over Platonic solids**. From left to right: input boundary curves/surfaces on a sphere; the associated domain polyhedra; the C^0 GC patches [Qin et al. 2024b]; and our C^0 GC volumes.

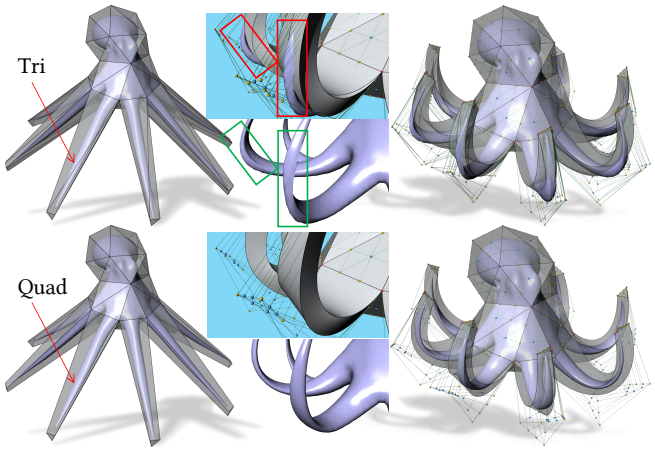


Fig. 14. **Tri cage** (top) vs. **Triquad cage** (bottom). Left: the original Octopus model with different cages. Right: deformation results. Middle: zoomed-in view.

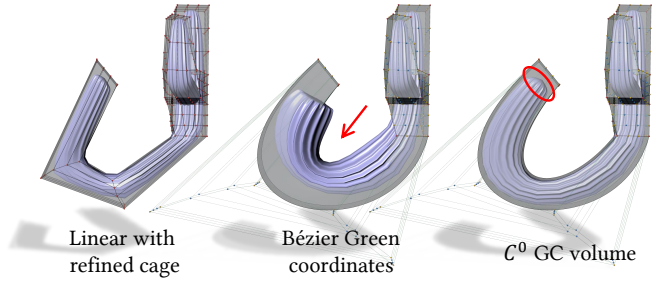


Fig. 15. **Quasi-conformal** vs. **Interpolatory**. Comparisons of our method with the linear method [Joshi et al. 2007] using refined cage and Bézier Green coordinates [Xiao and Chen 2025]. The original Cactus model with cage is shown in Fig. 10.

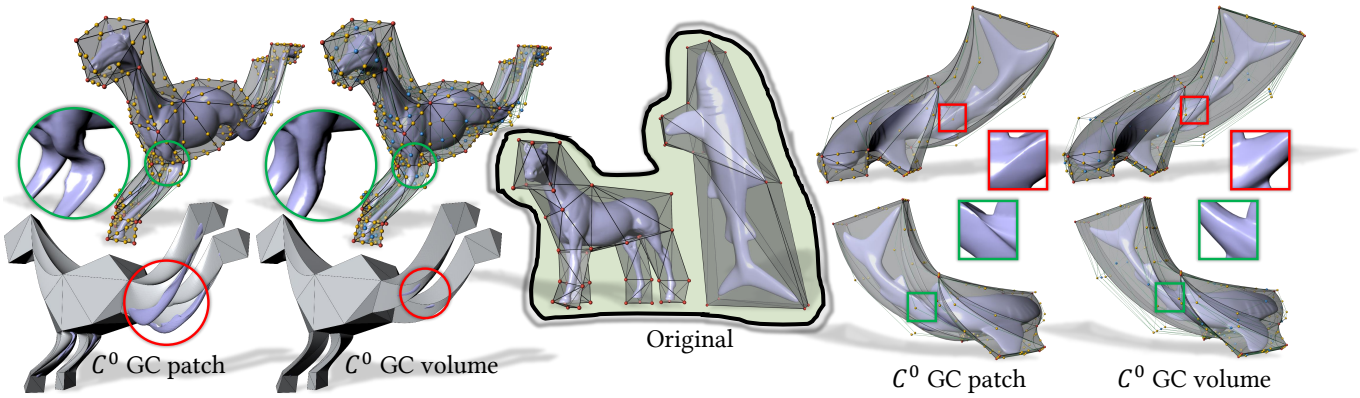


Fig. 16. **Curve-based** vs. **Surface-based**. Comparison of our method with C^0 GC patch [Qin et al. 2024b].

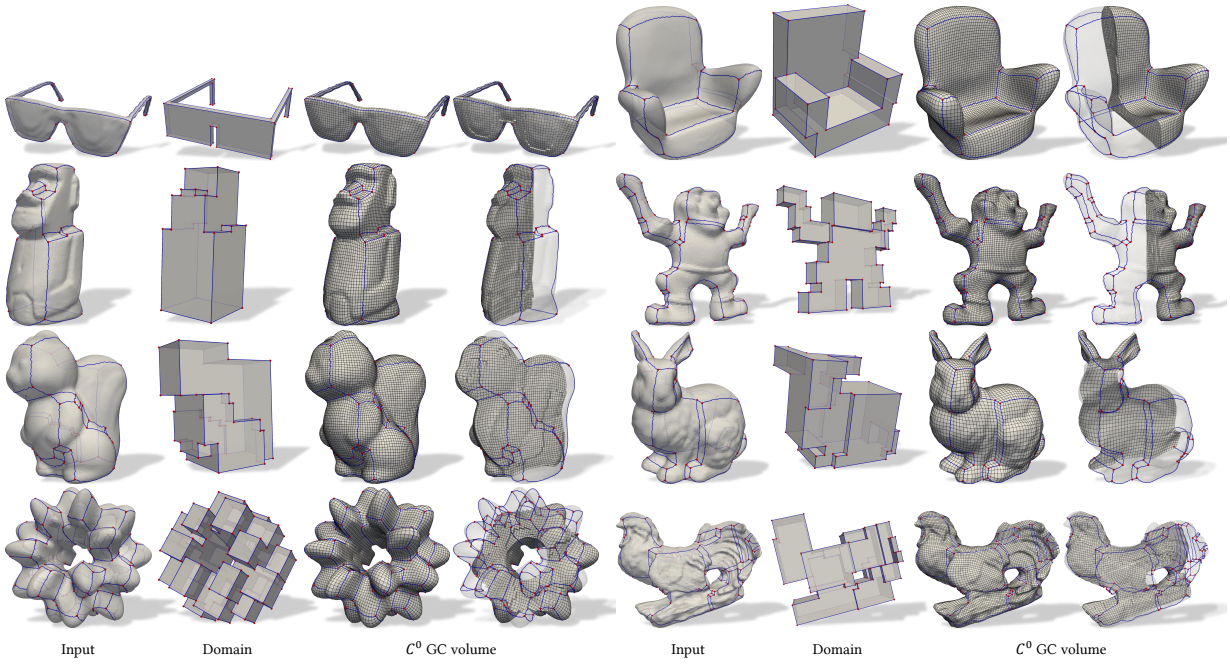


Fig. 17. **Interpolation over PolyCubes**. From left to right: the input boundary surfaces (triangle meshes); the corresponding domain polyhedra (PolyCubes); and the interpolating C^0 GC volumes.

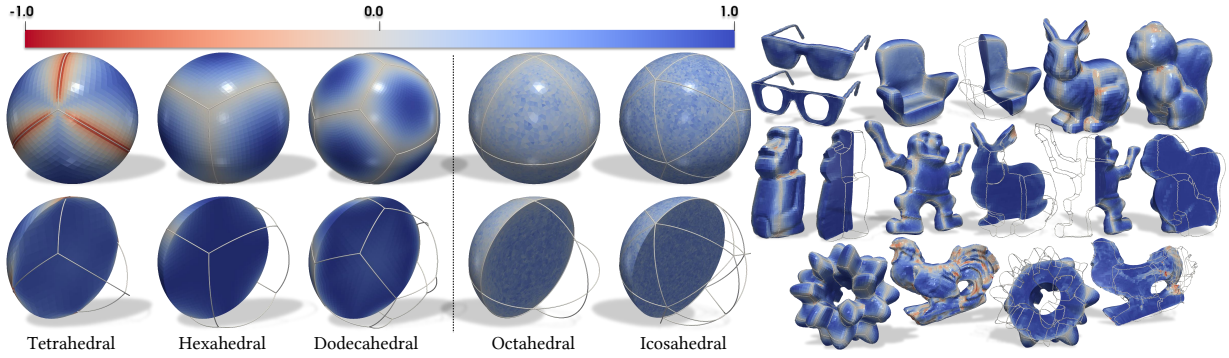


Fig. 18. **Scaled Jacobians** of the C^0 GC volumes. Red indicates negative Jacobians.