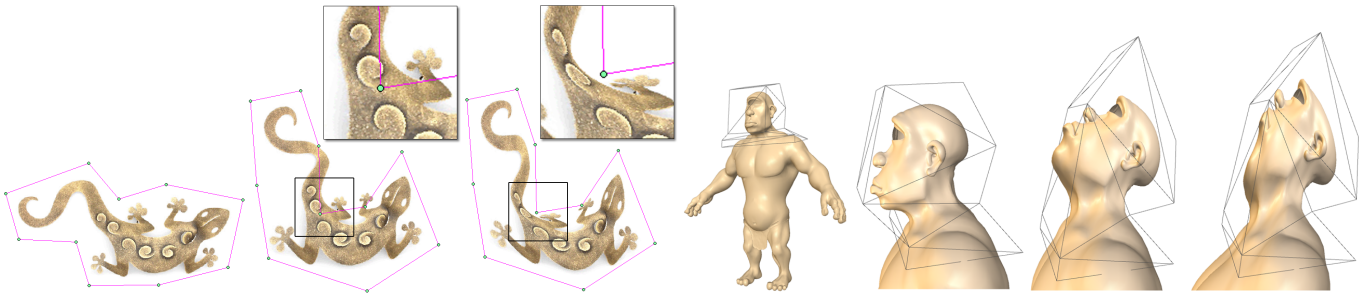
Green Coordinates

Yaron Lipman David Levin Daniel Cohen-Or
Tel-Aviv University



(a) (b) (c) (d) (e) (f)

Figure 1: (a-c) Cage-based 2D deformation of a Gecko. (b) Using Green Coordinates induces a pure conformal mapping. (c) The result of Harmonic Coordinates. Note the preservation of shape in the marked square. (d-f) Cage-based 3D articulation of an Ogre. (e) Using Green Coordinates in 3D admits a quasi-conformal deformation. In (f) the result using Mean Value Coordinates is presented. Note how Green Coordinates nicely preserve the shape of the Ogre's head.

Abstract

We introduce Green Coordinates for closed polyhedral cages. The coordinates are motivated by Green's third integral identity and respect both the vertices position and faces orientation of the cage. We show that Green Coordinates lead to space deformations with a shape-preserving property. In particular, in 2D they induce conformal mappings, and extend naturally to quasi-conformal mappings in 3D. In both cases we derive closed-form expressions for the coordinates, yielding a simple and fast algorithm for cage-based space deformation. We compare the performance of Green Coordinates with those of Mean Value Coordinates and Harmonic Coordinates and show that the advantage of the shape-preserving property is not achieved at the expense of speed or simplicity. We also show that the new coordinates extend the mapping in a natural analytic manner to the exterior of the cage, allowing the employment of partial cages.

1 Introduction

In recent years there is an increased interest in cages as practical means to manipulate 3D models [Floater 2003; Ju et al. 2005b; Joshi et al. 2007]. A cage is a low polygon-count polyhedron, which typically has a similar shape to the enclosed object. The points inside the cage are represented by affine sums of the cage's vertices multiplied by special weight functions called *coordinates*. Manipulating the cage induces a smooth space deformation of its interior. The main advantage of these cage-based space deformation techniques is their simplicity, flexibility and speed. Manipulating an enclosed object, for example a mesh surface, requires a rather small computational cost, since transforming a point requires merely a linear combination of the cage geometry using *precalculated* coordinates. Moreover, since each point is transformed independently, these techniques are indifferent to the surface representation and free of discretization errors.

However, current space-deformation techniques do not have good control over the preservation of shape and details, such as advanced surface-based deformation techniques. Throughout the paper we use the phrase *Shape-preserving deformations* as our main target. *Shape-preserving* deformations are smooth mappings such that their Jacobian matrices are close to rotations with isotropic scale. Notice that shape-preservation is reflecting local behavior of the transformation. That is, the shear component of the local transformation is small. Shape-preserving transformations are also referred to as *Quasi-conformal mappings*. While conformal mappings map infinitesimal balls into infinitesimal balls, with no shear at all, quasi-conformal mappings map infinitesimal balls into infinitesimal ellipsoids with bounded axis ratio.

Achieving shape-preserving space deformations defined by cage-based techniques seems unfeasible. The reason is that current cage methods express a point $\boldsymbol{\eta}$ inside a cage P as an *affine* sum of the cage vertices $\mathbb{V} = \{\boldsymbol{v}_i\}_{i \in I_\mathbb{V}} \subset \mathbb{R}^3$:

$$\boldsymbol{\eta} = F(\boldsymbol{\eta}; P) = \sum_{i \in I_\mathbb{V}} \varphi_i(\boldsymbol{\eta}) \boldsymbol{v}_i, \tag{1}$$

where $\varphi_i(\cdot)$ are referred to as "coordinates". Then the deformation defined by a deformed cage P' is defined by

$$\boldsymbol{\eta} \mapsto F(\boldsymbol{\eta}; P') = \sum_{i \in I_\mathbb{V}} \varphi_i(\boldsymbol{\eta}) \boldsymbol{v}'_i, \tag{2}$$

where $\mathbb{V}' = \{\boldsymbol{v}'_i\}_{i \in I_\mathbb{V}}$ are the deformed cage's vertices. These operators are *affine-invariant*. Consequently, when the cage undergoes an affine transformation, the operator reconstructs this affine transformation. Such affine transformations may include shear and anisotropic scale which violate the shape-preserving property. Moreover, the general form of the current cage-based operators

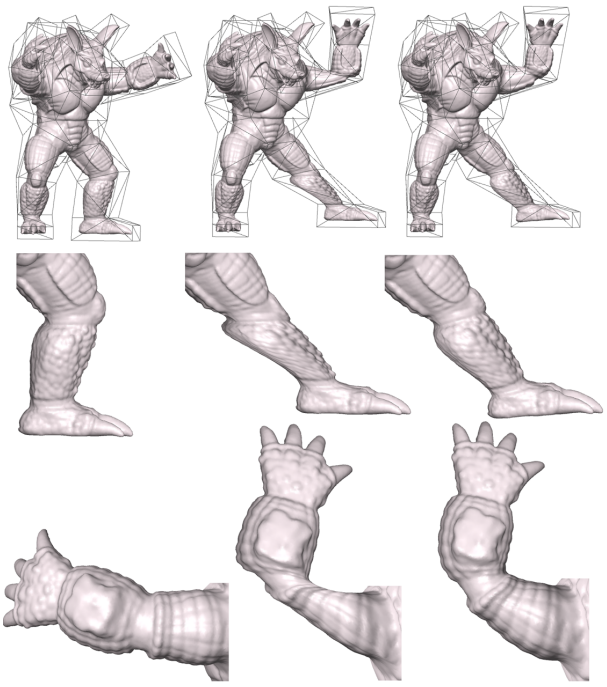


Figure 2: Cage-based articulation using Green Coordinates admits a quasi-conformal deformation in 3D (right column). The middle column shows the result of Mean Value Coordinates. Note how Green Coordinates preserve the details as well as the whole shape of the Armadillo's leg and arm.

(Eq. (1) and (2)) cannot produce shape-preserving mappings. This stems from the fact that respecting the requirement that the Jacobian consists of rotations and isotropic scaling, necessarily requires that the operator reflects a dependency between the different axes. However, in Eq. (2) each axis is treated independently of the others. For example, translating the x-axis coordinate of one cage's vertex has no affect on the y and z-axis coordinates whatsoever. The effect of the affine invariance property can be seen for example in Figure 3 where the details (bumps) maintain their original orientation under the translation of part of the cage.

Green Coordinates. Despite the above-mentioned limitation of cage-based operators, we show in the paper that it is still possible to define detail-preserving cage-based coordinates which retain all the advantages of the general cage-based operator. The coordinates that we present here introduce appropriate rotations into the space deformation to allow shape preservation. In Section 3 we show that these coordinates are derived from the theory of Green Functions [Nehari 1952; Kantorovich and Krylov 1964]. Therefore, we call them Green Coordinates (GC). This theory is applicable to piecewise-smooth boundaries in any dimension, and the resulting deformation operator does not require discretization. In $2D$ the operator is proved to induce a pure conformal mapping. Conformal mappings are the ideal shape-preserving deformations since they locally consist of rotations and isotropic scaling only, that is angle preserving, see Figure 4. In $3D$ the operator provides a natural generalization of these conformal maps, that is *quasi-conformal* maps. It should be noted that in 3D (and higher dimensions) no conformal mappings exist besides (composition of) similarity and inversion transformations [Blair 2000]. Quasi-conformal mapping is close to conformal in the sense that it allows a minimal amount of anisotropic scaling. We show the quasi-conformality empirically, that is, by checking that the distortion is bounded in $3D$. Furthermore, in both cases the operator has a closed-form analytic formula. By the term closed-form we mean that the coordinates can be calculated analytically from the cage positions without approximation and discretization of any kind.

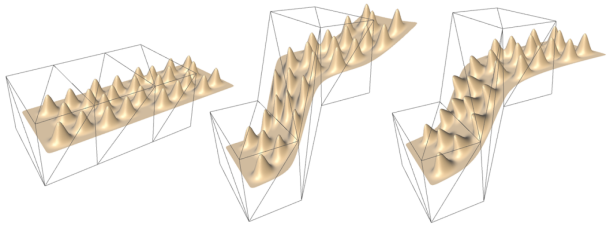


Figure 3: Detail preservation is exhibited using Green Coordinates (on the right), where the details adhere to the surface deformation and rotate accordingly. In the middle, the MVC result is depicted where the details maintain their original orientation and therefore shear.

To achieve cage-based coordinates with shape-preserving property, we necessarily need a slightly different operator than the one defined by Eq. (1),(2). The new coordinates respect the orientation of the cage's faces and not only the positions of the vertices: Let the cage be an oriented simplicial surface (i.e., 2D polygon, 3D triangular mesh), that is $P = (\mathbb{V}, \mathbb{T})$, where $\mathbb{V} = \{\boldsymbol{v}_i\}_{i \in I_\mathbb{V}} \subset \mathbb{R}^d$ are the vertices and $\mathbb{T} = \{t_j\}_{j \in I_\mathbb{T}}$ are the simplicial face elements, namely edges in case of polygons in 2D, triangles in case of triangular meshes in 3D. Let us further denote by $\boldsymbol{n}(t_j)$ the outward normal to the oriented simplicial face t_j ($\|\boldsymbol{n}(t_j)\| = 1$). Our general framework for defining the coordinates is derived by representing each interior point $\boldsymbol{\eta}$ as the linear combination

$$\boldsymbol{\eta} = F(\boldsymbol{\eta}; P) = \sum_{i \in I_\mathbb{V}} \phi_i(\boldsymbol{\eta}) \boldsymbol{v}_i + \sum_{j \in I_\mathbb{T}} \psi_j(\boldsymbol{\eta}) \boldsymbol{n}(t_j). \qquad (3)$$

Thus, the deformation induced by a deformed cage P' is defined by

$$\boldsymbol{\eta} \mapsto F(\boldsymbol{\eta}; P') = \sum_{i \in I_\mathbb{V}} \phi_i(\boldsymbol{\eta}) \boldsymbol{v}'_i + \sum_{j \in I_\mathbb{T}} \psi_j(\boldsymbol{\eta}) s_j \, \boldsymbol{n}(t'_j), \qquad (4)$$

where $\boldsymbol{v}'_i$ and t'_j denote the vertices and faces of P', respectively. The scaling factors $\{s_j\}_{j \in I_\mathbb{T}}$ are essential for achieving important properties such as scale invariance. The definition of the scalars $\{s_j\}$ is explained later on. In particular, in 2D, it is simply $s_j = \|t'_j\| / \|t_j\|$, where $\|t_j\|$ is the length of t_j. We remark here that although we aim at least distorting mappings, some applications may require more flexibility, e.g., the user may want to simply stretch an object, or part of it. Such features can also be achieved by an adequate choice of the factors $\{s_j\}$ (see Section 3 for details).

The new method, similarly to previous cage-based methods, allows fast interactive deformation that only require to compute linear sums (Eq. (4)) with the *precalculated* coordinates. In Section 3 we present a way of defining the coordinate functions ϕ, ψ so that the operator F has the desired shape-preserving property and a closed-form formula.

Let us precede by few examples: In Figure 1 (a-c) we perform 2D deformation, comparing Harmonic Coordinates (HC) [Joshi et al. 2007], and Green Coordinates (GC). In this example we articulate the tail of the gecko by manipulating the cage. As can be observed, the conformality of the deformation produced by Green Coordinates better preserves the shape. The Harmonic coordinates, on the other hand, are affine-invariant and as such may contain shears and

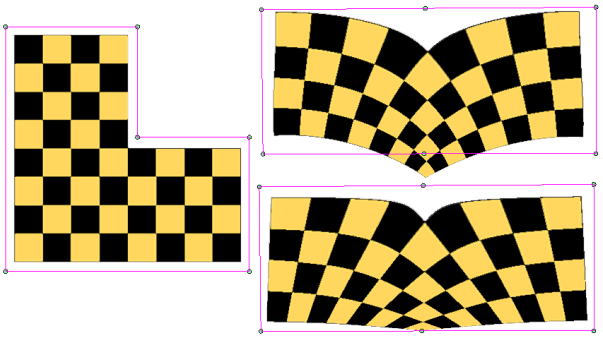


Figure 4: 'L'-shaped checkerboard is deformed. Left: The original checkerboard pattern and cage. Top-right: GC result. Bottom-right: the HC result. Note that in order to guarantee that the mapping is conformal, the map extends beyond the deformed cage.

non-uniform scalings. However, the HC deformation better adheres the cage than the GC deformation. In a sense, the shape preservation property becomes possible due to relaxation of the interpolation requirement. As can be observed in this figure, not insisting on interpolating the cage's boundaries allows the deformation to preserve the shape. Moreover, the shape-preserving property also helps preventing local foldovers (see Figure 13).

In Figure 1 (d-f) we perform similar comparison, now with Mean Value Coordinates (MVC) [Ju et al. 2005b] in 3D where we articulate the Ogre model. Note the preservation of the shape of the ogre's head, in particular his chin, mouth and forehead. Another example is shown in Figure 2, where the Armadillo's hand and leg are articulated. Note, that in these cases (not highly concave cages) employing the Harmonic Coordinates will yield similar results to the Mean Value Coordinates.

2 Background

Space deformation techniques were introduced by Sederberg and Parry [1986] and further extended by others [Coquillart 1990; MacCracken and Joy 1996; Kobayashi and Ootsubo 2003]. The basic space deformation technique defines a lattice with a rather small number of control points that encloses the subject model. Manipulating the control points smoothly deforms the space enclosed in the lattice, and the embedded geometry deforms accordingly. As indicated in [Joshi et al. 2007], the rigid spatial topological structure of the FFD latices makes the deformation less flexible. This motivated a search for a more general control polyhedron to enclose the model in a tighter fashion and have a better match of degrees of freedom to the subject model.

Floater [2003] has introduced the Mean Value Coordinates (MVC) for 2D polygons as a closed-form scheme for smoothly interpolating data on general polygons. Later [Ju et al. 2005b; Floater et al. 2005; Langer et al. 2006] have further generalized the Mean Value Coordinates to 3D. Ju et al. [2005b] presented a surface deformation technique based on these coordinates. The MVC have been subject to more theoretical investigation and have proved to be well-defined in the whole plane and infinitely smooth except at the vertices [Hormann and Floater 2006]. Joshi et al. [2007] introduced different cage-based coordinates called Harmonic Coordinates. These coordinates are non-negative and do not possess a local extrema. These properties lead to more intuitive control in the deformation process, mainly of highly concave cages, compared to the original MVC. However, Harmonic Coordinates do not possess closed-form formulas as MVC. Later, Lipman et al.[2007] presented alternative coordinates which are also non-negative. As we discussed in the introduction, all these methods are affine-invariant and not shape-preserving. Another alternative to compute space deformations is employing scattered-data interpolation methods like RBF [Kojekine et al. 2002; Botsch and Kobbelt 2005]. However, in these methods also each axis is treated independently and hence shape preservation is generally not possible. For a more complete discussion of previous work we note that previously to Floater's 2D Mean Value Coordinates there was a considerable amount of work done generalizing the barycentric coordinates to general polygons and polyhedra [Wachpress 1975; Pinkall and Polthier 1993; Warren 1996; Meyer et al. 2002; Ju et al. 2005a].

A different family of deformation techniques applies the deformation directly to the surface [Sorkine et al. 2004; Yu et al. 2004; Lipman et al. 2005; Zhou et al. 2005; Botsch et al. 2006; Huang et al. 2006; Sorkine and Alexa 2007; Au et al. 2007; Shi et al. 2007] These methods are based on measuring some deformation energy directly over the surface, or representing the surface with some tailored structures, and then optimizing them under some user constraints to yield the desired deformation. These "direct" approaches achieve high quality shape-preserving deformation. However, these methods require solving large, often non-linear, systems of equations, which may suffer from discretization errors. The GC technique that we present achieves similar shape-preservation quality as these direct methods with the advantage of closed-form expressions for the mapping operator.

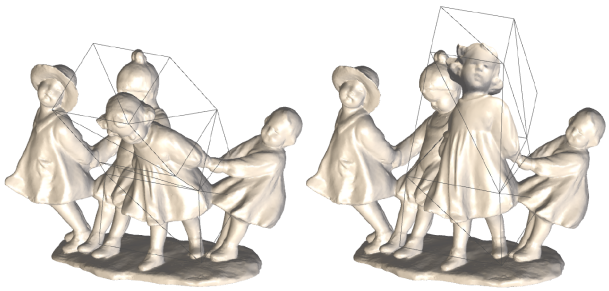


Figure 5: The original model (on the left) is modified by a partial cage to straighten the girl. Note the preservation of the dress details and the smooth extension of the deformation to the exterior of the cage.

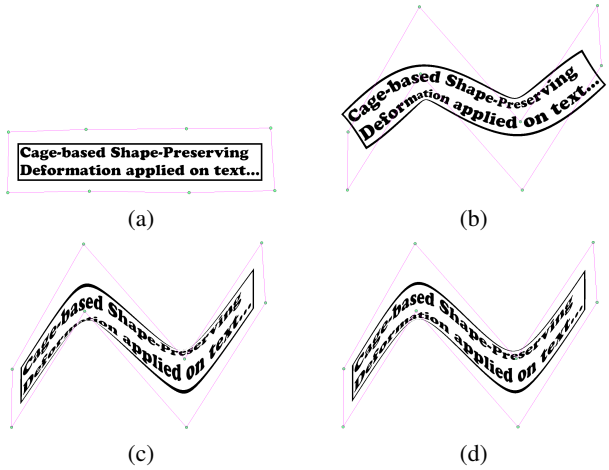


Figure 6: Deformation of a text with a coarse cage (a). The results of the Green, Mean Value and Harmonic Coordinates are displayed in (b),(c) and (d), respectively.

3 Derivation of Green Coordinates

In this section we derive the Green Coordinates in $\mathbb{R}^d$. As argued in the introduction, shape-preservation cannot be achieved by affine combinations of the cage's vertices alone, and we suggest to consider combinations of vertices and normals of the form (3), where the exact relation is coded in the coordinate functions $\{\phi_i\}$ and $\{\psi_j\}$ and the scalars $\{s_j\}$.

Our derivation of these coordinate functions is based upon the theory of Green functions and upon Green's third integral identity: Let $u(\boldsymbol{\xi}), \boldsymbol{\xi} = (\xi_1, ..., \xi_d)$ be a harmonic function in a domain $D \subset \mathbb{R}^d$ enclosed by a piecewise-smooth boundary ∂D. A function u is called harmonic if it is a solution to Laplace equation, i.e.,

$$\Delta u = \frac{\partial^2 u}{\partial {\xi_1}^2} + \frac{\partial^2 u}{\partial {\xi_2}^2} + ... + \frac{\partial^2 u}{\partial {\xi_d}^2} = 0. \tag{5}$$

Further, let $G(\cdot, \cdot)$ be the *fundamental solution* of the Laplace equation in $\mathbb{R}^d$, that is

$$\Delta_{\boldsymbol{\xi}} G(\boldsymbol{\xi}, \boldsymbol{\eta}) = \delta(\boldsymbol{\xi} - \boldsymbol{\eta}), \tag{6}$$

where $\delta(\cdot)$ is the Dirac delta function. Then, for any $\boldsymbol{\eta} \in D^{in} := interior(D)$, $u(\boldsymbol{\eta})$ can be expressed by its boundary values and boundary normal derivatives via Green's third identity:

$$u(\boldsymbol{\eta}) = \int_{\partial D} \left(u(\boldsymbol{\xi}) \frac{\partial G(\boldsymbol{\xi}, \boldsymbol{\eta})}{\partial \boldsymbol{n}(\boldsymbol{\xi})} - G(\boldsymbol{\xi}, \boldsymbol{\eta}) \frac{\partial u(\boldsymbol{\xi})}{\partial \boldsymbol{n}(\boldsymbol{\xi})} \right) d\sigma_{\boldsymbol{\xi}}, \tag{7}$$

where $\boldsymbol{n}$ is the oriented outward normal to ∂D and $d\sigma_{\boldsymbol{\xi}}$ is the area element on ∂D.

The solution of (6) in $\mathbb{R}^d$ without boundary conditions results in the *fundamental solutions* of the Laplace equation in $\mathbb{R}^d$, which have the following expressions:

$$G(\boldsymbol{\xi}, \boldsymbol{\eta}) = \begin{cases} \frac{1}{(2-d)\omega_d} \|\boldsymbol{\xi} - \boldsymbol{\eta}\|^{2-d} & d \geq 3 \\ \frac{1}{2\pi} \log \|\boldsymbol{\xi} - \boldsymbol{\eta}\| & d = 2 \end{cases}, \tag{8}$$

where ω_d is the area of a unit sphere in $\mathbb{R}^d$.

Now let us take the domain D to be the domain enclosed by our cage P, and let us use the coordinate functions $\boldsymbol{\eta} = (\eta_1, ..., \eta_d)$, which are linear functions, in the role of the harmonic function u in (7), that is $u(\boldsymbol{\eta}) = \boldsymbol{\eta}$:

$$\boldsymbol{\eta} = \int_{\partial D} \left(\boldsymbol{\xi} \frac{\partial G(\boldsymbol{\xi}, \boldsymbol{\eta})}{\partial \boldsymbol{n}(\boldsymbol{\xi})} - G(\boldsymbol{\xi}, \boldsymbol{\eta}) \frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{n}(\boldsymbol{\xi})} \right) d\sigma_{\boldsymbol{\xi}}. \tag{9}$$

Noting that the outward normal $\boldsymbol{n}(\boldsymbol{\xi})$ is constant on each face t_j we have $\partial \boldsymbol{\xi} / \partial \boldsymbol{n}(\boldsymbol{\xi}) = \partial \boldsymbol{\xi} / \partial \boldsymbol{n}(t_j) = \boldsymbol{n}(t_j)$, where $\boldsymbol{n}(t_j)$ denotes the outward normal of face t_j. Let us write the integral (9) as a sum of integrals over the cage's faces t_j (omitting the arguments in G and $\boldsymbol{n}$ for brevity):

$$\boldsymbol{\eta} = \sum_{j \in I_{\mathbb{T}}} \left(\int_{t_j} \boldsymbol{\xi} \frac{\partial G}{\partial \boldsymbol{n}} d\sigma_{\boldsymbol{\xi}} - \int_{t_j} G\, \boldsymbol{n}(t_j) d\sigma_{\boldsymbol{\xi}} \right), \quad \boldsymbol{\eta} \in D^{in}. \tag{10}$$

Denote by $\mathrm{N}\{\boldsymbol{v}_i\}$ the union of all faces in the 1-ring neighborhood of vertex $\boldsymbol{v}_i$, and let the function Γ_i be the piecewise-linear hat function defined on $\mathrm{N}\{\boldsymbol{v}_i\}$, which is one at $\boldsymbol{v}_i$, zero at all other vertices in the 1-ring and linear on each face. Then writing $\boldsymbol{\xi}$ as the (unique) barycentric combination in the simplicial face t_j, $\boldsymbol{\xi} = \sum_{k=1}^{d} \Gamma_k(\boldsymbol{\xi}) \boldsymbol{v}_k$, where $\boldsymbol{v}_k$ are the vertices of the face t_j, we get from (10)

$$\boldsymbol{\eta} = \sum_{j \in I_{\mathbb{T}}} \sum_{\boldsymbol{v}_k \in \mathbb{V}(t_j)} \boldsymbol{v}_k \left(\int_{t_j} \Gamma_k(\boldsymbol{\xi}) \frac{\partial G}{\partial \boldsymbol{n}} d\sigma_{\boldsymbol{\xi}} \right) - \sum_{j \in I_{\mathbb{T}}} \boldsymbol{n}(t_j) \left(\int_{t_j} G\, d\sigma_{\boldsymbol{\xi}} \right),$$

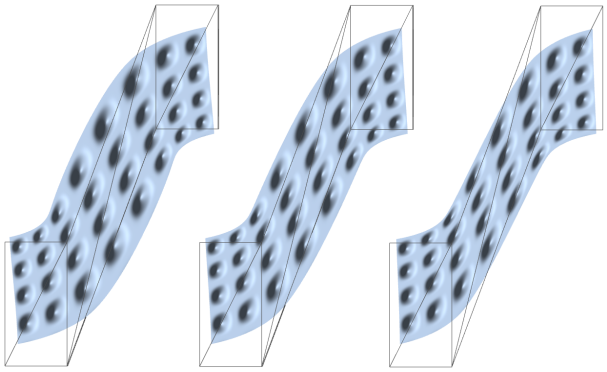


Figure 7: Different $\{s_j\}$ scaling to accommodate non-uniform stretch: Left, s_j by Eq. (14). In the middle $0.5(s_j + 1)$, and on the right $s_j = 1$.

where $\mathbb{V}(t_j)$ denotes the vertices of the face t_j. The last equation can be rearranged to get

$$\boldsymbol{\eta} = \sum_{i \in I_{\mathbb{V}}} \phi_i(\boldsymbol{\eta}) \boldsymbol{v}_i + \sum_{j \in I_{\mathbb{T}}} \psi_j(\boldsymbol{\eta}) \boldsymbol{n}(t_j), \quad \boldsymbol{\eta} \in D^{in}, \tag{11}$$

where the coordinate functions ϕ_i and ψ_j are

$$\begin{aligned} \phi_i(\boldsymbol{\eta}) &= \int_{\boldsymbol{\xi} \in \mathrm{N}\{\boldsymbol{v}_i\}} \Gamma_i(\boldsymbol{\xi}) \frac{\partial G(\boldsymbol{\xi}, \boldsymbol{\eta})}{\partial \boldsymbol{n}(\boldsymbol{\xi})} d\sigma_{\boldsymbol{\xi}} \quad i \in I_{\mathbb{V}} \\ \psi_j(\boldsymbol{\eta}) &= -\int_{\boldsymbol{\xi} \in t_j} G(\boldsymbol{\xi}, \boldsymbol{\eta}) d\sigma_{\boldsymbol{\xi}} \qquad j \in I_{\mathbb{T}}, \end{aligned} \tag{12}$$

To complete the construction of the mapping $\boldsymbol{\eta} \mapsto F(\boldsymbol{\eta}; P')$ defined by (4) we still need to define the scaling factors $\{s_j\}$. The definition of these factors is derived by the following properties, desirable for shape-preserving deformations:

1. *Linear reproduction*: $\boldsymbol{\eta} = F(\boldsymbol{\eta}; P)$, for $\boldsymbol{\eta} \in P^{in}$.
2. *Translation invariance*: $\sum_{i \in I_{\mathbb{V}}} \phi_i(\boldsymbol{\eta}) = 1$, for $\boldsymbol{\eta} \in P^{in}$.
3. *Rotation and scale invariance*: For an affine transformation which consists of a rotation with possible isotropic scale T, $T\boldsymbol{\eta} = F(\boldsymbol{\eta}; TP)$.
4. *Shape preservation*: For $d = 2$, the mapping $\boldsymbol{\eta} \mapsto F(\boldsymbol{\eta}; P')$ is conformal, for $d = 3$, this mapping is quasi-conformal.
5. *Smoothness*: $\{\phi_i(\boldsymbol{\eta})\}, \{\psi_j(\boldsymbol{\eta})\}$ are harmonic functions in P^{in}. Hence, they are C^∞ for $\boldsymbol{\eta} \in P^{in}$.

Linear reproduction is the basic relation (11) we started with, we just need to take $s_j = 1$ if $t'_j = t_j$. This choice is also suitable for the second property, together with the relation $\sum_{i \in I_{\mathbb{V}}} \phi_i(\boldsymbol{\eta}) = 1$ followed by applying (7) to the function $u(\boldsymbol{\eta}) \equiv 1$. To ensure the third property we take $s_j = \|T\|_2$, and thus $T\boldsymbol{n}(t_j) = s_j \boldsymbol{n}(t'_j)$. The face t_j, together with the point $\boldsymbol{v}_{j_1} + \boldsymbol{n}(t_j)$, where $\boldsymbol{v}_{j_1}$ is a vertex in t_j, define a simplex S_j in $\mathbb{R}^d$, and similarly t'_j and $\boldsymbol{v}'_{j_1} + s_j \boldsymbol{n}(t'_j)$ define a simplex S'_j. In the case of a similarity (rotation and uniform scaling) map T we have $T(S_j) = S'_j$. In the general case we would like to define s_j so that the linear mapping taking S_j onto S'_j is least-distorting. In other words, s_j should represent the *stretch* the face t_j undergoes as the cage is deformed. In 2D ($d = 2$) this stretch is well defined, simply take

$$s_j = \|t'_j\| / \|t_j\|, \tag{13}$$

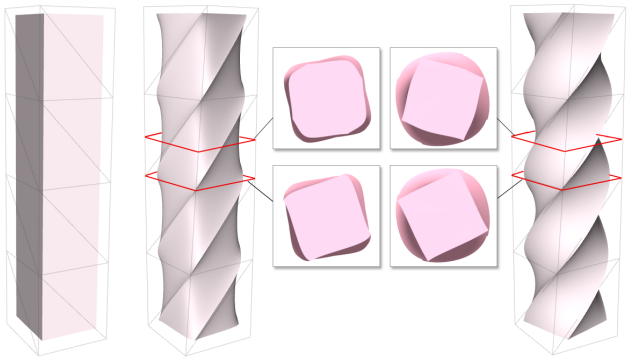


Figure 8: Twisting a bar using MVC (left) and GC (right). Note the two cuts displayed from top view.

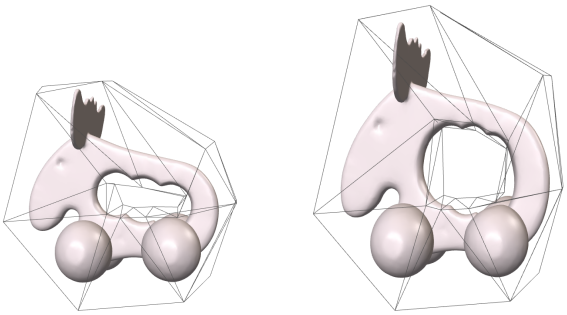


Figure 9: Deformation with non simply-connected cage (torus).

i.e., the exact stretch of the edge t_j. In higher dimensions, however, the stretch is not so evident and it cannot be described by a single scalar. Nevertheless, we find the following definition natural: In 3D, let σ_1, σ_2 be the singular values of the linear map taking t_j to t'_j. Then, to have a least-distorting map taking S_j onto S'_j we should define s_j as some average of σ_1 and σ_2. The choice that provided us with the desired quasi-conformality property is $s_j = \sqrt{\frac{\sigma_1^2+\sigma_2^2}{2}}$. Using computations presented in [Pinkall and Polthier 1993] for linear transformations between triangles in $\mathbb{R}^3$, one (t_j) with edges defined by the vectors $\boldsymbol{u}, \boldsymbol{v}$ and the other (t'_j) by the corresponding vectors $\boldsymbol{u}', \boldsymbol{v}'$, it turns out that

$$s_j = \frac{\sqrt{\|\boldsymbol{u}'\|^2\|\boldsymbol{v}\|^2 - 2(\boldsymbol{u}' \cdot \boldsymbol{v}')(\boldsymbol{u} \cdot \boldsymbol{v}) + \|\boldsymbol{v}'\|^2\|\boldsymbol{u}\|^2}}{\sqrt{8}area(t_j)}. \qquad (14)$$

Note that this final definition encapsulates and generalizes all of the above cases. As demonstrated by the examples throughout the paper, the above definition of the factors s_j leads to 'least-distorting' deformations. However, in some cases, one may be interested in a distortion, such as stretching the object non-uniformly. Such effects may still be achieved by replacing the definitions (13) and (14) by the simple choice $s_j = 1$. Intermediate effects may be obtained by sliding the values of s_j between these two options (see Figure 7).

The fifth property holds for any choice of $\{s_j\}$, and is due to the fact that for $\boldsymbol{\eta} \in P^{in}$ $\{\phi_i\}$ and $\{\psi_j\}$ can be differentiated an infinite number of times under the integral sign. Furthermore, since the function $G(\cdot,\cdot)$ is symmetric and harmonic, it implies that $\{\phi_i\}, \{\psi_j\}$ are also harmonic functions. Finally, regarding the fourth property, in the case of $d = 2$, the mapping $\boldsymbol{\eta} \mapsto F(\boldsymbol{\eta}; P')$ is pure conformal. The proof is rather long and technical and provided fully in [Lipman and Levin 2008]. Generally, the proof is based on

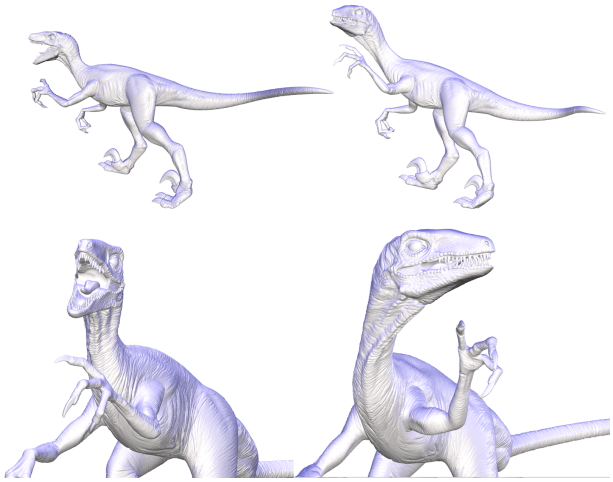
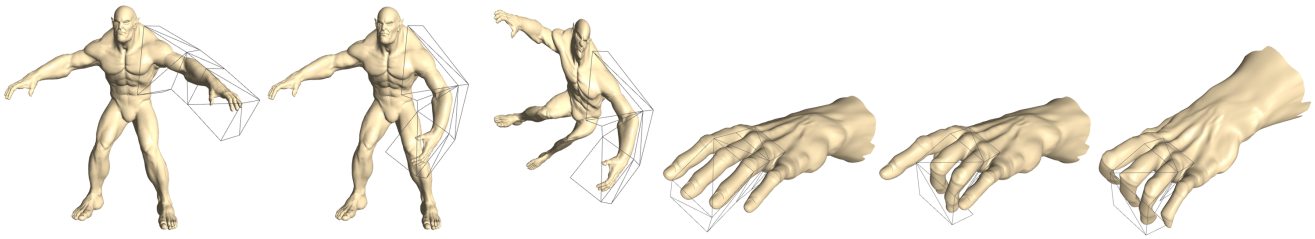


Figure 10: Deforming the Raptor model ($2000K$ triangles). Left - original model, right - GC deformation.

two simple ingredients: First, confirming that the coordinate functions $\{\phi_i\}$ and $\{\psi_j\}$ are in some sense conjugate harmonic. Second, observing that the gradient field of a harmonic function defines a conformal map.

Quasi-Conformality In $\mathbb{R}^d$, $d \geq 3$ we cannot expect to get pure conformal mappings. Instead, we wish to minimize as much as possible the shear component of the transformation. More formally, we define the *distortion* of a map $F : D \mapsto F(D) \subset \mathbb{R}^d$ at each point $\boldsymbol{\eta} \in D$ by $\frac{\sigma_{max}(\boldsymbol{\eta})}{\sigma_{min}(\boldsymbol{\eta})}$, where $\sigma_{min}(\boldsymbol{\eta})$ and $\sigma_{max}(\boldsymbol{\eta})$ are the minimal and maximal singular values of the differential of F (Jacobian matrix) at the point $\boldsymbol{\eta}$, respectively. Note that the choice of the scaling factors $\{s_j\}$ is also based upon this principle. A map with a bounded maximal distortion is called quasi-conformal. Figure 13 compares the deformation F induced by Green Coordinates, Mean Value Coordinates and Harmonic Coordinates, using two orthogonal planes with a circles pattern. This figure also shows the histogram of the distortions of each of the maps, defined in the interior of the cage. Note that the maximal distortion of GC mapping in these examples does not exceed the value of 3.2, while the maximal distortion of MVC and HC mappings has exceeded the value of 100. Note that the Y-axis is shown in a logarithmic scale. Applying other transformations to the same cage, we noticed that, in exception of degenerate cases, the deformations induced by Green Coordinates have a maximal distortion bounded by a constant ≤ 6. In contrast, the deformations induced by Mean Value Coordinates and Harmonic Coordinates present unbounded total distortion which is linearly proportional to the amount of distortion of the deformed cage. Figure 8 demonstrates 2π twisting of a bar model (each cage level is rotated by $\pi/2$). Note the two cuts depicted from top view: The GC preserves the square silhouette better than MVC.

Closed-form formulas for $2D$ and $3D$ Interestingly, closed-form formulas can be derived for the dimensions $d = 2, 3$ which are the cases considered in this paper. The derivation of the formulas is rather technical, so to keep the fluency of the reading we have attached only the final pseudocodes for calculating the 2D and 3D coordinates for $\boldsymbol{\eta} \in P^{in}$, see Algorithms 1 and 2 in Appendix A. The detailed derivations are listed in [Lipman and Levin 2008].



(a) (b)

Figure 11: Deformation using partial 3D cages. Note the local influence of the GC deformation (middle in (a) and (b)), compared to the global influence of the MVC deformation (right in (a) and (b)).

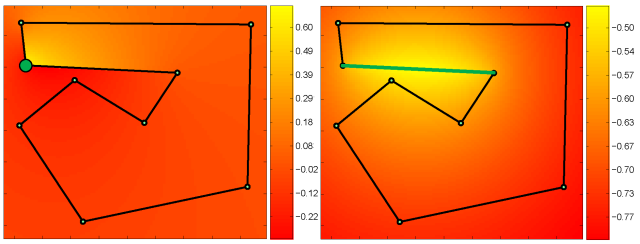


Figure 12: An illustration of the values of ϕ_i (left) for one vertex (marked in bold green point), and ψ_j (right) for one edge (marked in bold green line) in 2D.

4 Extending to the cage's exterior

The Green Coordinates defined by Eq. (3),(4) and (12) are smooth in the interior of the cage P. However, each coordinate $\phi_i(\boldsymbol{\eta})$ has jump discontinuities along the edges (simplicial faces) meeting at $\boldsymbol{v}_i$, see Figure 12. A natural question is whether the coordinates can be smoothly extended to the exterior of P. In $2D$ the Green Coordinates induce conformal transformations of the interior of P, and the above question is addressing the analytic continuation of these conformal transformations through the boundaries of P. An important application of such an extension is the deformation of a certain region of an object by a partial cage only, for example see Figure 5. A proper extension to the exterior of the partial cage would have smooth transition to the rest of the object and a diminishing influence, leaving the rest of the object in place.

In this section we derive the *unique* analytic continuation of the coordinates outside the cage, and show that it requires only a rather slight modification to the closed-form formulas at hand. Let us remark that the use of the term *analytic continuation* is twofold: In case $d = 2$ we refer to the classical meaning of extending the conformal (or analytic) complex maps. While in the case $d \geq 3$ we mean extending the map in a *real-analytic* manner. That is, we show that the coordinate functions are real-analytic, which means they can be locally represented as a power series, and then their extension is unique in their (connected) domain.

Extension through a face. Let us start by describing how the coordinates can be extended through some face $t_\ell \in \mathbb{T}$, $\ell \in I_\mathbb{T}$ of the cage. Let $i_1, ..., i_d \in I_\mathbb{V}$ be the indices of the vertices which consist of the face t_ℓ. First, the mapping $\boldsymbol{\eta} \mapsto F(\boldsymbol{\eta}; P')$ is conformal also in the exterior of the cage, which we denote by P^{ext}. However, $F(\boldsymbol{\eta}; P) = 0$ for $\boldsymbol{\eta} \in P^{ext}$ (this can be seen by similar argumentation to Section 3). Therefore, the important linear reproduction (property 1 in Section 3) does not hold outside the cage. In addition, the coefficients $\phi_{i_k}(\cdot), k = 1, ..., d$ are not continuous across the face t_ℓ. In view of this, in order to extend the coordinates (12) smoothly through t_ℓ we take the following path.

From properties 1 and 2 listed in Section 3 we have that the coordinates $\phi_{i_1}(\boldsymbol{\eta}), ..., \phi_{i_d}(\boldsymbol{\eta}), \psi_\ell(\boldsymbol{\eta})$ where $\boldsymbol{\eta} \in P^{in}$ satisfy

$$\sum_{k=1}^{d} \phi_{i_k}(\boldsymbol{\eta})\boldsymbol{v}_{i_k} + \psi_\ell(\boldsymbol{\eta})\boldsymbol{n}(t_\ell) = \boldsymbol{\eta} - \sum_{\substack{i \neq i_k \\ k = 1..d}} \phi_i(\boldsymbol{\eta})\boldsymbol{v}_i - \sum_{j \neq \ell} \psi_j(\boldsymbol{\eta})\boldsymbol{n}(t_j), \tag{15}$$

and

$$\sum_{k=1}^{d} \phi_{i_k}(\boldsymbol{\eta}) = 1 - \sum_{\substack{i \neq i_k \\ k = 1..d}} \phi_i(\boldsymbol{\eta}). \tag{16}$$

This yields a linear system for the coefficients $\phi_{i_k}(\boldsymbol{\eta}), k = 1..d$ and $\psi_\ell(\boldsymbol{\eta})$. It can be shown that this system is invertible for any $\boldsymbol{\eta} \in \mathbb{R}^d$ (see [Lipman and Levin 2008]). Thus, we may view this system as an alternative way to define $\phi_{i_1}(\boldsymbol{\eta}), ..., \phi_{i_d}(\boldsymbol{\eta}), \psi_\ell(\boldsymbol{\eta})$. Therefore, it is natural to extend the coordinates across the face t_ℓ by keeping the original definition for all the coordinates except $\phi_{i_k}(\boldsymbol{\eta}), k = 1..d$ and $\psi_\ell(\boldsymbol{\eta})$ and for the later coordinates, in both sides of t_ℓ, by the system of linear equations (15),(16). To distinguish the newly defined coordinates from the original ones defined in (12) we denote the new ones as $\tilde{\phi}_{i_k}(\boldsymbol{\eta})$ and $\tilde{\psi}_\ell(\boldsymbol{\eta})$. Note that $\tilde{\phi}_i(\boldsymbol{\eta}) = \phi_i(\boldsymbol{\eta})$ and $\tilde{\psi}_j(\boldsymbol{\eta}) = \psi_j(\boldsymbol{\eta})$ inside the cage.

Simplifying the system (15),(16) using the facts that for $\boldsymbol{\eta} \in P^{ext}$ we have $F(\boldsymbol{\eta}; P) = 0$ and $\sum_i \phi_i(\boldsymbol{\eta}) = 0$, we obtain

$$\begin{aligned} \tilde{\phi}_{i_k}(\boldsymbol{\eta}) &= \phi_{i_k}(\boldsymbol{\eta}) + \alpha_k(\boldsymbol{\eta}) \quad k = 1, .., d \\ \tilde{\psi}_\ell(\boldsymbol{\eta}) &= \psi_\ell(\boldsymbol{\eta}) + \beta(\boldsymbol{\eta}) \,, \end{aligned} \tag{17}$$

where $\{\alpha_k(\boldsymbol{\eta})\}$ and $\beta(\boldsymbol{\eta})$ vanish for $\boldsymbol{\eta} \in P^{in}$ and for $\boldsymbol{\eta} \in P^{ext}$ they satisfy

$$\begin{aligned} \sum_{k=1}^{d} \alpha_k(\boldsymbol{\eta})\boldsymbol{v}_{i_k} + \beta(\boldsymbol{\eta})\boldsymbol{n}(t_\ell) &= \boldsymbol{\eta} \\ \sum_{k=1}^{d} \alpha_k(\boldsymbol{\eta}) &= 1. \end{aligned} \tag{18}$$

Furthermore, for a point $\boldsymbol{\eta}$ on the exact boundary of P we get the same equations where the right hand sides are multiplied by $1/2$.

System (18) defines $\{\alpha_k(\boldsymbol{\eta})\}$ and $\beta(\boldsymbol{\eta})$ as the unique affine coordinates of the point $\boldsymbol{\eta}$ in the simplex defined by the vertices $\{\boldsymbol{v}_{i_k}\}$ of the face t_ℓ plus the vertex $\boldsymbol{v}_{i_1} + \boldsymbol{n}(t_\ell)$: $\boldsymbol{\eta} = L(\boldsymbol{\eta}; P, \ell)$ where

$$\begin{aligned} L(\boldsymbol{\eta}; P, \ell) &= (\alpha_1(\boldsymbol{\eta}) - \beta(\boldsymbol{\eta}))\, \boldsymbol{v}_{i_1} \\ &+ \sum_{k=2}^{d} \alpha_k(\boldsymbol{\eta})\boldsymbol{v}_{i_k} + \beta(\boldsymbol{\eta})\,(\boldsymbol{v}_{i_1} + \boldsymbol{n}(t_\ell))\,. \end{aligned} \tag{19}$$

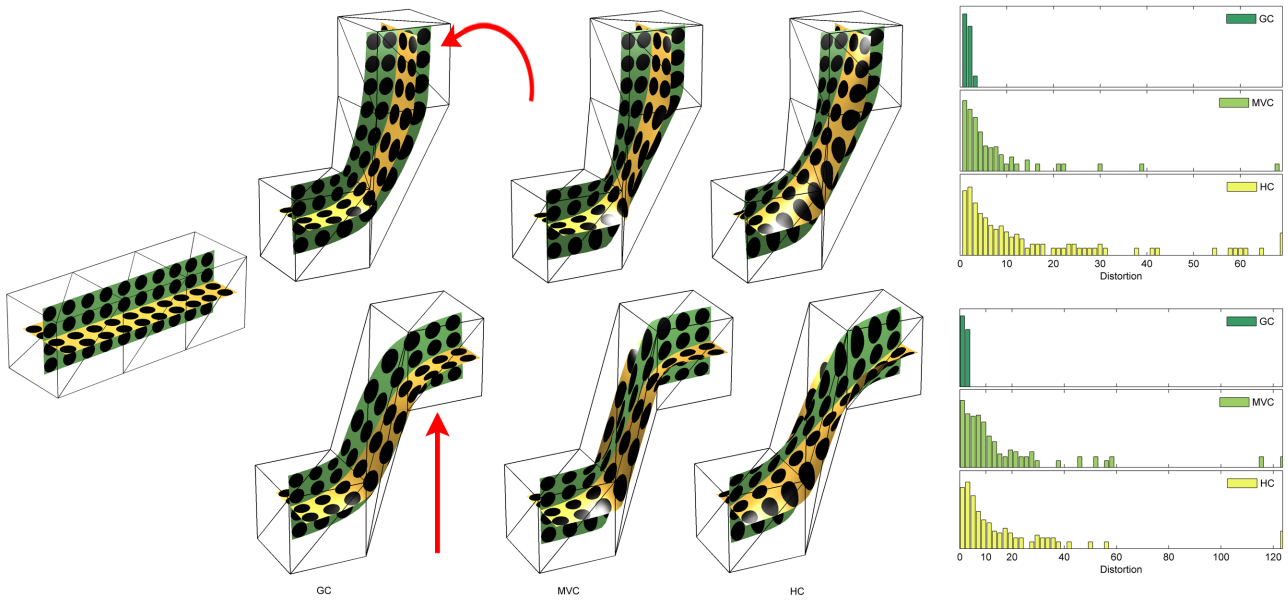


Figure 13: Comparison of GC , MVC and HC. Two intersecting planes with circles pattern enclosed by a simple cage (left) are deformed twice: Each row demonstrates a different cage manipulation, indicated by an arrow. Note that MVC and HC might cause some shear, significant stretching and foldovers. On the right: The histogram of the distortion values of each map in logarithmic scale (see Section 3).

Furthermore, note that $\{\alpha_k(\boldsymbol{\eta})\}$ are the unique barycentric coordinates of the projection of $\boldsymbol{\eta}$ onto the hyperplane defined by the simplicial face t_ℓ. Hence, they also have a closed-form expressions. The above derivation implies that this simple correction (17) to the coordinates $\phi_{i_k}(\boldsymbol{\eta}), k = 1..d$ and $\psi_\ell(\boldsymbol{\eta})$ in the exterior of P provides the unique analytic continuation through the face t_ℓ:

Theorem 4.1. *The mapping*

$$\tilde{F}(\boldsymbol{\eta}; P') = \sum_{i \in I_{\mathbb{V}}} \tilde{\phi}_i(\boldsymbol{\eta}) \boldsymbol{v}'_i + \sum_{j \in I_{\mathbb{T}}} \tilde{\psi}_j(\boldsymbol{\eta}) s_j \boldsymbol{n}(t'_j) \qquad (20)$$

in the $2D$ case is the unique complex-analytic extension of the mapping $\boldsymbol{\eta} \mapsto F(\boldsymbol{\eta}; P')$ through the edge t_ℓ. In 3D, $\tilde{\phi}_i$,$\tilde{\psi}_j$ are the unique real-analytic (and harmonic) extensions of the coordinate functions ϕ_i, ψ_j through the face t_ℓ.

Note that the mapping outside the cage can be written as

$$\tilde{F}(\boldsymbol{\eta}; P') = F(\boldsymbol{\eta}; P') + L(\boldsymbol{\eta}; P', \ell) \quad \text{for } \boldsymbol{\eta} \in P^{ext}, \qquad (21)$$

where $L(\boldsymbol{\eta}; P', l)$ is defined by replacing $\boldsymbol{v}_{i_k}$ and $\boldsymbol{n}(t_\ell)$ in (19) by their transformed versions $\boldsymbol{v}'_{i_k}$ and $s_\ell \boldsymbol{n}(t'_\ell)$ from P'.

Proof: For the 2D case, we note that two holomorphic functions that coincide on a line, are the unique analytic continuation of each other. Hence, it is enough to show that the mapping (20) is conformal inside and outside the cage, and that it is continuous on the edge t_ℓ. The conformality inside and outside the cage is proven in [Lipman and Levin 2008]. The continuity across the edge t_ℓ can be understood from the fact that the new coordinates $\tilde{\phi}_i$,$\tilde{\psi}_j$ (inside and outside) are solutions of the non-singular system of equations (15),(16) which has C^∞ smooth coefficients.

In the 3D case, we note that the new coordinate functions are harmonic both inside and outside the cage (we are only adding an affine function outside, see (21)). As explained above, the new coordinates are smooth across the face t_ℓ and therefore the new coordinates are harmonic through the face t_ℓ. We note that harmonic functions are real-analytic, and real-analytic functions in connected domains which coincide on an open set coincide everywhere [Sheldon Axler 2001]. Therefore, we have that since the new coordinates $\tilde{\phi}_i$,$\tilde{\psi}_j$ coincide with ϕ_i,ψ_j inside the cage, they are actually the unique real-analytic (and harmonic) extension through the face t_ℓ.

Deformation with partial cages. The above procedure of the coordinate extension allows the employment of partial cages. The construction of cages around the entire model may not always be simple, while fitting partial cages around the region of interest is rather simple. Canonical simple shaped cages can then be used as tools for local deformation. Figure 11 shows an example of a simple cage fitted twice: once to the whole arm of the character and once to two fingers only.

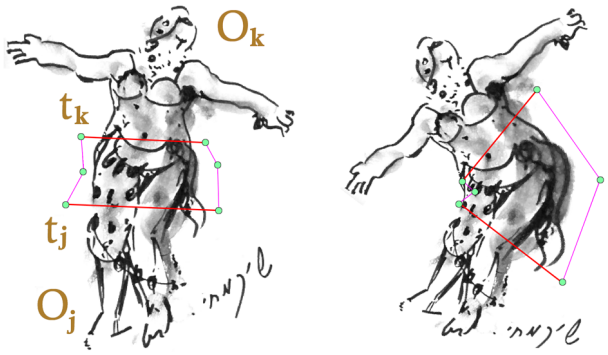
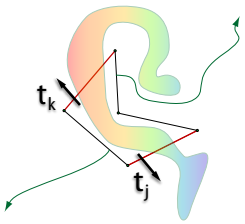


Figure 14: 2D deformation using a partial cage. The Green Coordinates are extended through two faces t_j, t_k (colored red).

It is possible to extend the coordinates through every face, by adding to the transformation $F(\boldsymbol{\eta}; P')$ outside the cage, the affine transformation $L(\boldsymbol{\eta}; P', l)$. As proved in Theorem 4.1 this extension is unique. Therefore, in the common case where different cage

faces undergo different affine transformations, it is not possible to extend the deformation analytically over the *whole* space. Yet, it is important to note that for our purpose it is enough to define an extension which is *smooth on the object* to be deformed. The simplest option would be to extend the coordinates through one face, which we call the *exit face*. We know that our transformation would always be smooth though this face. Now, if the exit face is transformed by a similarity transformation (rigid transformation possibly with uniform scaling) then our definition of the scaling factor s_ℓ assures us that $L(\boldsymbol{\eta}; P', \ell)$ reconstruct this (spatial) similarity affine transformation. We also observe that the deformation will also be smooth through all the other faces which undergo the same similarity transformation $L(\boldsymbol{\eta}; P', \ell)$ as the exit face. Let us call this condition the *similarity condition*. For a smooth deformation it is enough to ensure that the object does not intersect faces which are not satisfying the similarity condition.

In the case of extending the coordinates through two exit faces t_j, t_k the exterior of the cage may be cut into two disjoint connected parts, one which include t_j and one which include t_k, and define the extension in each part accordingly. The deformation will always be smooth through both these exit faces, and over any object which does not intersect the cut or any of the faces not satisfying the similarity condition, as in the following figure:



The same principle holds for any number of extensions (E_j) through exit faces (t_j). That is, the exterior of the cage is decomposed into disjoint connected parts O_j, such that $t_j \subset O_j$, and thus each O_j would be subject to a different (corresponding) extension E_j. The deformation will be smooth in each part O_j through the exit faces t_j and all other faces which satisfy the similarity condition in O_j. Figure 14 shows the extension through two edges (t_j, t_k) colored red.

For different results of partial cage deformations, see Figures 3,5,11. An interesting point which appears in these examples (especially in Figure 11), is that although the Mean Value Coordinates are well-defined and smooth everywhere outside the cage [Hormann and Floater 2006], their influence is not decaying outside the cage, and the effect of partial cage manipulation is not local. Note that Harmonic Coordinates are not defined outside the cage.

5 Implementation and Results

The Green Coordinates and their associated calculations and application were developed in C++ and MATLAB. Although the coordinates are defined for arbitrary dimension, we have implemented them only for the two and three dimension cases. Let $\Lambda = \{\boldsymbol{\eta}\} \subset \mathbb{R}^d$ denote the subject of the deformation, where Λ can be an arbitrary collection of points in any dimension $\mathbb{R}^d$. The cage, denoted by P, is a closed polygonal line in 2D, triangular mesh in 3D or a simplicial surface in $\mathbb{R}^d$. It should be emphasized that P is not necessarily simply connected (see Figure 9). To ease its manipulation, the cage should be as coarse as possible, while still having a reasonable amount of degrees of freedom and flexibility to perform the desired deformation. In our experiments, we learned that even a very coarse cage leads to deformations which are more plausible than those achieved with affine-invariant methods (see for example Figures 6 and 15).

Given a cage, the coordinate functions $\phi_i(\boldsymbol{\eta}), \psi_j(\boldsymbol{\eta})$, $i \in I_{\mathbb{V}}, j \in I_{\mathbb{T}}, \boldsymbol{\eta} \in \Lambda$ are calculated in a closed-form manner as a function of the cage geometry. The pseudocodes for $\boldsymbol{\eta} \in P^{in}$ are presented in Algorithms 1 and 2 for the $2D$ and $3D$ cases, respectively. In Table 1 we give some timings for the coordinate evaluation (preprocess) and deformation, calculated using single thread on a Core 2^{TM} 2.4GHz, 3.5GB RAM machine.

Example	verts	tris	Eval (sec)	Deform (sec)
Hand (11) Hand cage	14K 16	28K 28	2.3	0.010
Man (11) Man cage	32K 16	64K 28	5.3	0.024
Elk (9) Elk cage	20K 35	41K 70	8.5	0.036
Ogre (1) Ogre cage	62K 35	124K 70	9.98	0.044
Budha (15) Budha cage	543K 16	1087K 28	89.1	0.395
Armadillo (2) Armadillo cage	173K 110	345K 216	215.5	0.930 (0.204)
Raptor (10) Raptor cage	1000K 16	2000K 28	160.9	0.720

Table 1: Green coordinates evaluation and deformation times.

During the online session, the deformation of the subject is applied in real-time rate. The user manipulates the cage's geometry, $P \rightarrow P'$, and the deformed geometry of the subject $\Lambda \rightarrow F(\Lambda; P')$ is immediately reconstructed via the linear sum Eq. (4). In cases when the user manipulates only a small part of the cage (as the Armadillo example in Figure (2), see also the table above) it is possible to further accelerate the speed of the deformation by taking advantage of the fact that manipulation involves only a subset of the cage's vertices. Thus, it is more efficient to calculate the difference of the locations of each transformed point. Consequently, if P', P'' are source and target cages, then the deformation difference $\boldsymbol{\eta}'' - \boldsymbol{\eta}' = F(\boldsymbol{\eta}; P'') - F(\boldsymbol{\eta}; P')$ is a function of only the modified faces and vertices. For example, using this method for deforming the Armadillo model reduces the deformation time from 0.930 seconds (see the table above) to 0.204 seconds. The simplicity of this online calculation (i.e., linear combinations of *constant* precalculated coordinates), allows interactive deformation of huge models. For example, the deformation of the Buddha model, which consists of $1087K$ triangles, is deformed at interactive frame rates (see the accompanying video), and the Raptor model (Figure 10), which consists of $2000K$ triangles.

6 Conclusions

We have introduced the Green Coordinates for cage-based deformations. The new coordinates provide shape-preserving mappings from the space $\mathbb{R}^d$ into itself. For the $d = 2, 3$ cases, we extracted closed-form formulas to simplify their computation. It is proved that in the 2D case the deformations are conformal, and we show that they extend to quasi-conformal in 3D. Furthermore, it is shown that the coordinates can be analytically extended to the exterior of the cage allowing the usage of partial cages.

As we showed in the paper, the deformation is not interpolatory. This can be considered as a limitation in applications that require interpolation of the cage's boundary. However, a cage is defined quite loosely around the shape, and the cage is a rather convenient deformation tool for articulating shapes. Another issue, is the deformation's computational complexity in comparison to other free-from methods such as MVC or HC. Since GC are defined on the faces of the cage as well as the vertices of the cage, the number of terms in the linear sum used to calculate the deformation (4) con-

tains about three times the number of terms appearing at the vertex based methods (2).

We would like to stress that the definition of conformal mappings has been extensively investigated and it typically involves complex constructions and approximate numerical solutions. Here, the target domain is not prescribed or given as constraints. The target domain is defined on-the-fly to resemble the geometry of a target cage. We find it surprising that these conformal and quasi-conformal mappings come in such simple, closed-form formulas. We thus believe that there are further applications for Green Coordinates beyond deformations. Another interesting direction for future work is to use the added degrees of freedom in Eq. (4) to make the mapping onto the deformed cage. Another practical direction for future research is employing GPU techniques, such as vertex shaders, to further accelerate the on-line deformation.

7 Acknowledgments

We would like to thank Kiaran Ritchie for the Ogre model, Scott Schaefer for the Armadillo's cage and Adi Levin for his valuable comments on an early version of the paper. We also thank the anonymous reviewers for their helpful comments and suggestions. The Raptor, Dancing Children, Elk (Figure 9) models where taken from the AIM@SHAPE shape repository. The man and arm models are courtesy of Autodesk®. This work was supported in part by grants from the Israeli Ministry of Science and the Israel Science Foundation.

References

AU, O. K.-C., FU, H., TAI, C.-L., AND COHEN-OR, D. 2007. Handle-aware isolines for scalable shape editing. *ACM Trans. Graph. 26*, 3, 83.

BLAIR, D. E. 2000. *Inversion Theory and Conformal Mapping*. American Mathematical Society.

BOTSCH, M., AND KOBBELT, L. 2005. Real-time shape editing using radial basis functions. *Computer Graphics Forum 24*, 3, 611 – 621.

BOTSCH, M., PAULY, M., GROSS, M., AND KOBBELT, L. 2006. Primo: coupled prisms for intuitive surface modeling. In *SGP '06*, 11–20.

COQUILLART, S. 1990. Extended free-form deformation: a sculpturing tool for 3d geometric modeling. *Proceedings of SIGGRAPH '90*, 187–196.

FLOATER, M. S., KOS, G., AND REIMERS, M. 2005. Mean value coordinates in 3d. *Computer Aided Geometric Design 22, 7*, 623–631.

FLOATER, M. S. 2003. Mean value coordinates. *Computer Aided Geometric Design 20, 1*, 19–27.

HORMANN, K., AND FLOATER, M. S. 2006. Mean value coordinates for arbitrary planar polygons. *ACM Transactions on Graphics 25, 4*, 1424–1441.

HUANG, J., SHI, X., LIU, X., ZHOU, K., WEI, L.-Y., TENG, S.-H., BAO, H., GUO, B., AND SHUM, H.-Y. 2006. Subspace gradient domain mesh deformation. *ACM Trans. Graph. 25*, 3, 1126–1134.

JOSHI, P., MEYER, M., DEROSE, T., GREEN, B., AND SANOCKI, T. 2007. Harmonic coordinates for character articulation. *Transactions on Graphics 26, 3 (Proc. SIGGRAPH)*.

JU, T., SCHAEFER, S., WARREN, J., AND DESBRUN, M. 2005. A geometric construction of coordinates for convex polyhedra using polar duals. In *SGP '05*, 181.

JU, T., SCHAEFER, S., AND WARREN, J. 2005. Mean value coordinates for closed triangular meshes. vol. 24, 3 (Proc. SIGGRAPH), 561–566.

KANTOROVICH, L. V., AND KRYLOV, V. I. 1964. *Approximate methods of higher analysis*. Interscience Publishers, INC.

KOBAYASHI, K. G., AND OOTSUBO, K. 2003. t-ffd: free-form deformation by using triangular mesh. In *SM '03*, 226–234.

KOJEKINE, N., SAVCHENKO, V., SENIN, M., AND HAGIWARA, I., 2002. Real-time 3d deformations by means of compactly supported radial basis functions.

LANGER, T., BELYAEV, A., AND SEIDEL, H.-P. 2006. Spherical Barycentric Coordinates . 81–88.

LIPMAN, Y., AND LEVIN, D. 2008. On the derivation of green coordinates. *Technical Report*.

LIPMAN, Y., SORKINE, O., LEVIN, D., AND COHEN-OR, D. 2005. Linear rotation-invariant coordinates for meshes. In *Proceedings of ACM SIGGRAPH 2005*, ACM Press, 479–487.

LIPMAN, Y., KOPF, J., COHEN-OR, D., AND LEVIN, D. 2007. Gpu-assisted positive mean value coordinates for mesh deformations. In *SGP '07*, 117–123.

MACCRACKEN, R., AND JOY, K. I. 1996. Free-form deformations with lattices of arbitrary topology. In *SIGGRAPH '96*, 181–188.

MEYER, M., BARR, A., LEE, H., AND DESBRUN, M. 2002. Generalized barycentric coordinates on irregular polygons. *J. Graph. Tools 7*, 1, 13–22.

NEHARI, Z. 1952. *Conformal mapping*. McGraw-Hill.

PINKALL, U., AND POLTHIER, K. 1993. Computing discrete minimal surfaces and their conjugates. *Experimental Mathematics*.

SEDERBERG, T. W., AND PARRY, S. R. 1986. Free-form deformation of solid geometric models. *SIGGRAPH '86*, 151–160.

SHELDON AXLER, PAUL BOURDON, R. W. 2001. *Harmonic Function Theory*. Springer.

SHI, X., ZHOU, K., TONG, Y., DESBRUN, M., BAO, H., AND GUO, B. 2007. Mesh puppetry: cascading optimization of mesh deformation with inverse kinematics. *ACM Trans. Graph. 26*, 3, 81.

SORKINE, O., AND ALEXA, M. 2007. As-rigid-as-possible surface modeling. In *SGP '07: Proceedings of the fifth Eurographics symposium on Geometry processing*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 109–116.

SORKINE, O., LIPMAN, Y., COHEN-OR, D., ALEXA, M., RÖSSL, C., AND SEIDEL, H.-P. 2004. Laplacian surface editing. In *SGP '04*, 179–188.

WACHPRESS, E. 1975. A rational finite element basis. *manuscript*.

WARREN, J. 1996. Barycentric coordinates for convex polytopes. *Advances in Computational Mathematics 6*, 2, 97–108.

YU, Y., ZHOU, K., XU, D., SHI, X., BAO, H., GUO, B., AND SHUM, H.-Y. 2004. Mesh editing with poisson-based gradient field manipulation. *ACM Trans. Graph. 23*, 3, 644–651.

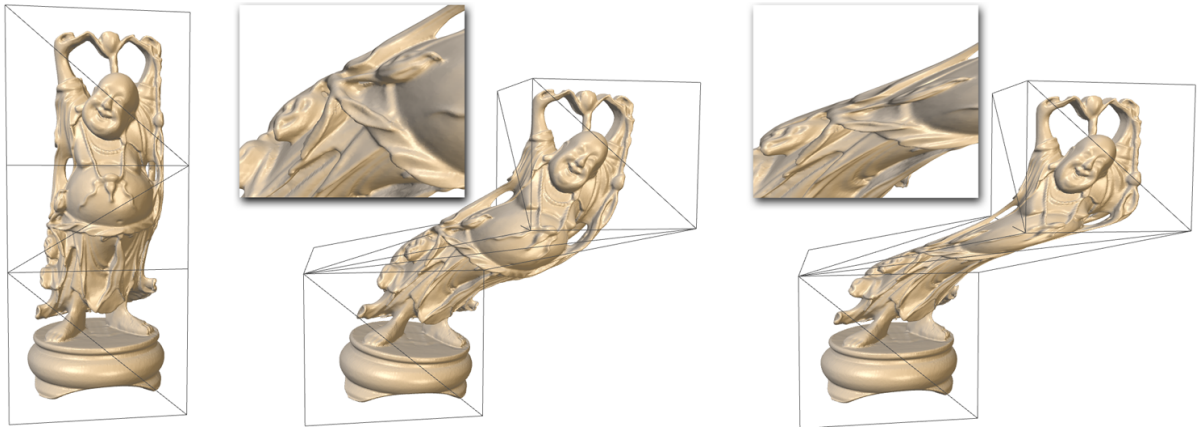


Figure 15: Deformation of a large model ($1087K$ triangles) in real-time is shown in the middle (See the accompanying video), and on the right is the result using MVC.

ZHOU, K., HUANG, J., SNYDER, J., LIU, X., BAO, H., GUO, B., AND SHUM, H.-Y. 2005. Large mesh deformation using the volumetric graph laplacian. *ACM Trans. Graph. 24*, 3, 496–503.

Appendix A

In this appendix we lay out the pseudocodes for calculating Green Coordinates in 2D and 3D for interior points $\Lambda \subset P^{in}$. We note that for exterior or boundary points one should add to these coordinates the $\{\alpha_k\}$ and β as explained in Section 4. Note that α_k and β also posses a simple closed-form formula employing the regular barycentric coordinates in triangles (3D) or edges (2D).

Input: cage $P = (\mathbb{V}, \mathbb{T})$, set of points $\Lambda = \{\boldsymbol{\eta}\}$
Output: $2D$ GC $\phi_i(\boldsymbol{\eta}), \psi_j(\boldsymbol{\eta}), i \in I_{\mathbb{V}}, j \in I_{\mathbb{T}}, \boldsymbol{\eta} \in \Lambda$
/* Initialization */
set all $\phi_i = 0$ and $\psi_j = 0$
/* Coordinate computation */
foreach *point* $\boldsymbol{\eta} \in \Lambda$ **do**
  foreach *edge* $j \in I_{\mathbb{T}}$ *with vertices* $\boldsymbol{v}_{j_1}, \boldsymbol{v}_{j_2}$ **do**
    $\boldsymbol{a} := \boldsymbol{v}_{j_2} - \boldsymbol{v}_{j_1}$; $\boldsymbol{b} := \boldsymbol{v}_{j_1} - \boldsymbol{\eta}$
    $Q := \boldsymbol{a} \cdot \boldsymbol{a}$; $S := \boldsymbol{b} \cdot \boldsymbol{b}$; $R := 2\boldsymbol{a} \cdot \boldsymbol{b}$
    $BA := \boldsymbol{b} \cdot \|\boldsymbol{a}\| \boldsymbol{n}(t_j)$; $SRT := \sqrt{4SQ - R^2}$
    $L0 := \log(S)$; $L1 := \log(S + Q + R)$
    $A0 := \frac{tan^{-1}(R/SRT)}{SRT}$
    $A1 := \frac{tan^{-1}((2Q+R)/SRT)}{SRT}$
    $A10 := A1 - A0$; $L10 := L1 - L0$
    $\psi_j(\boldsymbol{\eta}) :=$
    $-\|\boldsymbol{a}\|/(4\pi) \left[\left(4S - \frac{R^2}{Q} \right) A10 + \frac{R}{2Q} L10 + L1 - 2 \right]$
    $\phi_{j_2}(\boldsymbol{\eta}) := \phi_{j_2}(\boldsymbol{\eta}) - \frac{BA}{2\pi} \left[\frac{L10}{2Q} - A10 \frac{R}{Q} \right]$
    $\phi_{j_1}(\boldsymbol{\eta}) := \phi_{j_1}(\boldsymbol{\eta}) + \frac{BA}{2\pi} \left[\frac{L10}{2Q} - A10 \left(2 + \frac{R}{Q} \right) \right]$
  end
end

Algorithm 1: $2D$ Green Coordinates algorithm.

Input: cage $P = (\mathbb{V}, \mathbb{T})$, set of points $\Lambda = \{\boldsymbol{\eta}\}$
Output: $3D$ GC $\phi_i(\boldsymbol{\eta}), \psi_j(\boldsymbol{\eta}), i \in I_{\mathbb{V}}, j \in I_{\mathbb{T}}, \boldsymbol{\eta} \in \Lambda$
/* Initialization */
set all $\phi_i = 0$ and $\psi_j = 0$
/* Coordinate computation */
foreach *point* $\boldsymbol{\eta} \in \Lambda$ **do**
  foreach *face* $j \in I_{\mathbb{T}}$ *with vertices* $\boldsymbol{v}_{j_1}, \boldsymbol{v}_{j_2}, \boldsymbol{v}_{j_3}$ **do**
    foreach $\ell = 1, 2, 3$ **do**
      $\boldsymbol{v}_{j_\ell} := \boldsymbol{v}_{j_\ell} - \boldsymbol{\eta}$
    $\boldsymbol{p} := (\boldsymbol{v}_{j_1} \cdot \boldsymbol{n}(t_j)) \boldsymbol{n}(t_j)$
    foreach $\ell = 1, 2, 3$ **do**
      $s_\ell :=$
      $sign\left(\left((\boldsymbol{v}_{j_\ell} - \boldsymbol{p}) \times (\boldsymbol{v}_{j_{\ell+1}} - \boldsymbol{p}) \right) \cdot \boldsymbol{n}(t_j) \right)$
      $I_\ell := \textbf{GCTriInt}(\boldsymbol{p}, \boldsymbol{v}_{j_\ell}, \boldsymbol{v}_{j_{\ell+1}}, 0)$
      $II_\ell := \textbf{GCTriInt}(0, \boldsymbol{v}_{j_{\ell+1}}, \boldsymbol{v}_{j_\ell}, 0)$
      $\boldsymbol{q}_\ell := \boldsymbol{v}_{j_{\ell+1}} \times \boldsymbol{v}_{j_\ell}$
      $\boldsymbol{N}_\ell := \boldsymbol{q}_\ell / \|\boldsymbol{q}_\ell\|$
    $I := -\left| \sum_{k=1}^{3} s_k I_k \right|$
    $\psi_j(\boldsymbol{\eta}) := -I$
    $\boldsymbol{w} := \boldsymbol{n}(t_j) I + \sum_{k=1}^{3} \boldsymbol{N}_k II_k$
    if $\|\boldsymbol{w}\| > \epsilon$ **then**
      foreach $\ell = 1, 2, 3$ **do**
        $\phi_{j_\ell}(\boldsymbol{\eta}) := \phi_{j_\ell}(\boldsymbol{\eta}) + \frac{\boldsymbol{N}_{\ell+1} \cdot \boldsymbol{w}}{\boldsymbol{N}_{\ell+1} \cdot \boldsymbol{v}_{j_\ell}}$
  end
end

Procedure **GCTriInt**($\boldsymbol{p}$,$\boldsymbol{v}_1$,$\boldsymbol{v}_2$,$\boldsymbol{\eta}$)
$\alpha := \cos^{-1}\left(\frac{(\boldsymbol{v}_2 - \boldsymbol{v}_1) \cdot (\boldsymbol{p} - \boldsymbol{v}_1)}{\|\boldsymbol{v}_2 - \boldsymbol{v}_1\| \|\boldsymbol{p} - \boldsymbol{v}_1\|} \right)$
$\beta := \cos^{-1}\left(\frac{(\boldsymbol{v}_1 - \boldsymbol{p}) \cdot (\boldsymbol{v}_2 - \boldsymbol{p})}{\|\boldsymbol{v}_1 - \boldsymbol{p}\| \|\boldsymbol{v}_2 - \boldsymbol{p}\|} \right)$
$\lambda := \|\boldsymbol{p} - \boldsymbol{v}_1\|^2 \sin(\alpha)^2$
$c := \|\boldsymbol{p} - \boldsymbol{\eta}\|^2$
foreach $\theta = \pi - \alpha, \pi - \alpha - \beta$ **do**
  $S := \sin(\theta)$; $C := \cos(\theta)$
  $I_\theta := \frac{-sign(S)}{2} \left[2\sqrt{c} \tan^{-1}\left(\frac{\sqrt{c}C}{\sqrt{\lambda + S^2 c}} \right) + \sqrt{\lambda} \log\left(\frac{2\sqrt{\lambda}S^2}{(1-C)^2} \left(1 - \frac{2cC}{c(1+C) + \lambda + \sqrt{\lambda^2 + \lambda c S^2}} \right) \right) \right]$
return $\frac{-1}{4\pi} \left| I_{\pi-\alpha} - I_{\pi-\alpha-\beta} - \sqrt{c}\beta \right|$

Algorithm 2: $3D$ Green Coordinates algorithm.