

3D Model Deformations With Arbitrary Control Points

M.Àngels Cerveró, Àlvar Vinacua, Pere Brunet

*Departament de Ciències de la Computació
Universitat Politècnica de Catalunya
Campus Nord
08034 Barcelona*

Abstract

Cage-based space deformations are often used to edit and animate images and geometric models. The deformations of the cage are easily transferred to the model by recomputing fixed convex combinations of the vertices of the cage, the control points. In current cage-based schemes the configuration of edges and facets between these control points affects the resulting deformations. In this paper we present a family of similar schemes that includes some of the current techniques, but also new schemes that depend only on the positions of the control points. We prove that these methods afford a solution under fairly general conditions and result in an easy and flexible way to deform objects using freely placed control points, with the necessary conditions of positivity and continuity.

Keywords: Deformations, Cage-based, Interactive mesh deformation

1. Introduction

Techniques to deform three dimensional models are important in computer graphics. They can be used as modelling tools, to animate models, or within simulations. Additionally, some applications may require the deformation to satisfy other restrictions, like clamped portions of the model, or volume preservation.

A large number of methods currently in use and in the literature follow the Cage paradigm, whereby the model is surrounded by a coarse polyhedral cage, and the vertices of that cage—the control vertices—are used as handles to control the deformations. To describe how the space inside (and around) the cage deforms as the vertices of the cage move, some form of generalized barycentric coordinates with respect to the control vertices is used. These schemes give each point a set of coordinates that depend on the relative position of the point itself and the control points. Given such a coordinate system, when the cage is deformed, it is just a matter of computing the new positions of the points with the given coordinates to retrieve the deformed model. If these coordinates are smooth, the induced deformations will also be smooth. The advantage, of course, is that the user (or the simulation or optimization code) must only concern with a small number of handles (the control points) as opposed to a very large number of points (the vertices of the model). This paradigm is simple, elegant and efficiently deforms the models. However, some cage-free deformation techniques have been introduced recently. They provide more flexibility in the choice of deformation handles—which

may not be connected—and provide powerful tools to make the deformation process more versatile and intuitive.

In our research, we are especially interested in the deformation of soft tissues in medical or biological models. These models represent organs and tissues which are soft and lack an internal rigid structure. They are elastic but incompressible. In these cases an obvious guiding structure to help in devising a cage seldom exists, making schemes that do not rely on connectivity more natural to use.

In this paper we propose some new methods to compute a set of generalized barycentric coordinates which are *cage-free* and depend only on the positions of the deformation handles. The main contributions we present here are:

- The definition of a formal framework, the Celestial Coordinates, in which many of the existing schemes can be described.
- Two new Celestial Coordinate schemes that depend only on the positions of the control points, and not on their connectivity.

Section 2 discusses the previous work in this area. Then, Section 3 defines the Celestial Coordinates family and Sections 4 and 5 derive two new systems that belong to this family. Finally, Sections 6, 7 and 8 present results to evaluate these new schemes and our plans for future work along these lines.

2. Previous Work

There is a lot of bibliography proposing different types of Generalised Barycentric Coordinate (GBC) systems so the deformations of the control points have the desired properties of smoothness, locality and real-time responsiveness.

Email addresses: macervero@lsi.upc.edu (M.Àngels Cerveró),
alvar@cs.upc.edu (Àlvar Vinacua), pere@lsi.upc.edu (Pere Brunet)

The best known examples are the classical *Barycentric Coordinates*, defined by Möbius in 1827, which restrict the cage to be a simplex. More recently, *Mean-Value Coordinates* (MVC) [1] have been extended to 3D [2, 3]. They generalise barycentric coordinates to the kernel (the set of points that see all vertices) of star-shaped polyhedral cages. These coordinates are guaranteed to be positive in this kernel and C^∞ inside and outside of it. However, they are only C^0 on its boundary.

Harmonic Coordinates (HC) [4] do not have a closed formulation because they depend on the specific problem to solve. HC are C^∞ inside the cage and C^0 on its boundary. They have a more local effect than the MVC.

Positive Mean Value Coordinates (PMVC) [5] were introduced to ensure positive coordinates all over the cage, not only in its kernel. To fulfil this requirement they must relax the constraints of smoothness and continuity through the supporting planes of the boundary of this cage.

The *Green Coordinates* (GC) [6] are a new approach to perform shape-preserving deformations that require the normals of the faces of the cage to compute the coordinates. They are C^∞ inside and outside the cage but discontinuous at its boundary.

Although all these schemes provide efficient deformations, MVC may distort local details and PMVC and GC have discontinuities at the boundaries.

Li et al. [7] present a deformation technique using GC. The supporting cage is replaced by an umbrella shaped cell. This umbrella is automatically constructed over a point of the model specified by the user and updated during the deformation step. This method simulates shape deformation schemes in terms of the flexibility of the control handles. It also performs local shape-preserving deformations in real-time. Although the construction of the umbrella is completely transparent to the user, this technique is still completely dependent on the topology of the pseudo-cage.

Garcia et al. [8] present a multi-cage system to restrict the deformations to local region. Furthermore, their technique also increases the continuity of the coordinates across the boundaries between cages by computing a blending function applied in a parametrized neighbourhood of these faces.

Finally, Jacobson et al. [9] propose the *Bounded Biharmonic Weights* (BBW) that allow multiple deformation controls. The user can operate with cages, skeletons and isolated control points to accomplish the desired deformation. Their method reaches its goals through a space discretisation and the minimisation of a Laplacian energy.

3. Overview of Celestial Coordinates

Let us consider a set of \mathbb{R}^3 vertices $\mathcal{V} = \{\mathbf{v}_1, \dots, \mathbf{v}_n\}$. Unless explicitly defined otherwise, in what follows we will consider that the Deformation Domain \mathcal{D} of the set \mathcal{V} is its Convex Hull. Generalised Barycentric Coordinates (GBC) assign n coordinates to any 3D point $\mathbf{p} \in \mathcal{D}$,

$$\mathbf{p} = \sum_{i=1}^n \mathbf{v}_i \alpha_i(\mathbf{p}), \quad (1)$$

where $\alpha_i(\mathbf{p})$ is the Generalised Barycentric Coordinate of \mathbf{p} with respect to the i -th control vertex \mathbf{v}_i . These coordinates are only computed once for any relevant point of the deformable geometric model. Deformations are then driven by \mathcal{V} in a very simple way through Equation (1), using the initial coordinates $\alpha_i(\mathbf{p})$ and the new positions of the control vertices \mathbf{v}'_i .

By converting Equation (1) into $\sum_{i=1}^n (\mathbf{v}_i \alpha_i(\mathbf{p}) - \mathbf{p}) = 0$, we can write

$$\sum_{i=1}^n \beta_i \vec{\mathbf{q}}_i = \vec{\mathbf{0}} \quad (2)$$

where $\vec{\mathbf{q}}_i = \frac{(\mathbf{v}_i - \mathbf{p})}{d_i}$ and $d_i = \|(\mathbf{v}_i - \mathbf{p})\|$. With this expression, the final coordinates α_i are obtained as

$$\alpha_i(\mathbf{p}) = \frac{\omega_i(\mathbf{p})}{\sum_{j=1}^n \omega_j(\mathbf{p})}, \quad (3)$$

where $\omega_i(\mathbf{p}) = \frac{\beta_i}{d_i}$, a well-known alternative formulation of GBC.

In what follows, we will use the term *Celestial Coordinates* to identify the family of positive Generalised Barycentric Coordinate (GBC) schemes. They are defined by Equation (2) by imposing that $\forall i \beta_i \geq 0$

$$\sum_{i=1}^n \beta_i \vec{\mathbf{q}}_i = \vec{\mathbf{0}} \quad \forall i \beta_i \geq 0. \quad (4)$$

The name of *Celestial Coordinates* (CC) comes from the use of unit vectors $\vec{\mathbf{q}}_i$, which are the projection of the control vertices \mathbf{v}_i over a unit Sphere during their computation process in Equation (4). We call this Sphere the *Celestial Sphere*.

Positive Mean-Value Coordinates [5] are a good example of a member of the Celestial Coordinates family since they are always positive in their domain. However, most of the schemes discussed in the previous Section are not CC. For example, Mean-Value Coordinates [2] are only positive if \mathbf{p} is located inside the Kernel of the user-defined cage. Thus, they behave as Celestial Coordinates if, and only if, the Cage is convex. A similar situation takes place with the Spherical Barycentric Coordinates [10]. Unlike these schemes, the goal of the next two Sections is to propose CC schemes which are only based on the position of the control vertices \mathbf{v}_i and which do not depend on user-defined Cages nor on automatically computed connectivity among these control vertices. We therefore define a CC-subfamily called *Point-Based Celestial Coordinates*. They include all schemes in the CC family that do not need any kind of connectivity between the control vertices to compute the set of coordinates.

Any algorithm providing a set of positive β_i values fulfilling Equation (4) for every 3D point $\mathbf{p} \in \mathcal{D}$ is a CC-scheme candidate (it should also fulfill standard GBC properties as reproduction of the identity, reproduction of the unity and smoothness, as discussed in Section 6). Anyway, Equation (4) has two possible interpretations:

- First, it can be seen as a set of three scalar products between \mathbb{R}^n vectors. Let us define the vectors $\chi, y, z \in \mathbb{R}^n$ as the x-coordinates, y-coordinates and z-coordinates of the projected vectors $\vec{\mathbf{q}}_i$, and the vector $\boldsymbol{\beta} = \{\beta_1, \dots, \beta_n\}$. Then,

Equation (4) requires that $\beta \in \mathbb{R}_+^n$ and also that $\beta \in V^\perp$ where V is the linear space spanned by χ, y, z . In other words, β must belong to the region $V^\perp \cap \mathbb{R}_+^n$ in \mathbb{R}^n .

- Equation (4) can also be interpreted in \mathbb{R}^3 , by defining a convex linear combination of the projected vectors \vec{q}_i which must result in the null vector.

Observe that $V^\perp \cap \mathbb{R}_+^n$ is always non-empty for points $\mathbf{p} \in \mathcal{D}$, as any point inside a Convex Hull can be expressed as a convex combination of the vertices that define this Convex Hull, and \mathbf{p} is always in the Convex Hull of the projections of the control vertices on the unit Sphere.

The following Sections present two new schemes that belong to the *Point-Based Celestial Coordinates* family. They are the *T-Celestial Coordinates* and the *S-Celestial Coordinates*. *T-Celestial Coordinates* derive from the first \mathbb{R}^n interpretation, whereas *S-Celestial Coordinates* come from the \mathbb{R}^3 one.

4. T-Celestial Coordinates

T-Celestial Coordinates derive from the \mathbb{R}^n interpretation in Section 3. They are based on a transformation function T which maps any vector in the span of the vectors $\chi, y, z \in \mathbb{R}^n$ to the positive region \mathbb{R}_+^n . The computation of the vector β for any point \mathbf{p} by means of the function T requires solving a system of equations, which we prove always has a solution.

Given a particular point \mathbf{p} , the *T-Celestial Coordinates* work with the vectors $\chi, y, z \in \mathbb{R}^n$. Any v in the span of these three vectors can be encoded by three coordinates $\psi = (\psi_\chi, \psi_y, \psi_z)$, such that $v = \psi_\chi \chi + \psi_y y + \psi_z z$. Let $S \subset \mathbb{R}^n$ be the span of χ, y, z . The vector β and the final set of coordinates for \mathbf{p} also depend on the triplet $(\psi_\chi, \psi_y, \psi_z)$. The proposed method consists in, first of all, applying a transformation function $T(\psi_\chi, \psi_y, \psi_z)$ to v , which lends the name to the present coordinates. Such transformation sends v to \mathbb{R}_+^n

$$T : \begin{array}{l} S \subset \mathbb{R}^n \rightarrow \mathbb{R}_+^n \\ v \rightarrow T(\psi_\chi, \psi_y, \psi_z). \end{array} \quad (5)$$

Now, by imposing

$$\begin{aligned} T(\psi_\chi, \psi_y, \psi_z) \cdot \chi &= 0 \\ T(\psi_\chi, \psi_y, \psi_z) \cdot y &= 0 \\ T(\psi_\chi, \psi_y, \psi_z) \cdot z &= 0, \end{aligned} \quad (6)$$

we can solve for the unknowns $(\psi_\chi, \psi_y, \psi_z)$. Let

$$\beta = T(\psi_\chi, \psi_y, \psi_z)$$

for the particular triplet $(\psi_\chi, \psi_y, \psi_z)$ that solves System (6). The final set of coordinates $\alpha_i(\mathbf{p})$ can, then, be computed as shown in Equation (3).

Next, we show that we can define a function $T(\psi_\chi, \psi_y, \psi_z)$ such that System (6) has a solution.

4.1. The Transformation Function $T(\psi_\chi, \psi_y, \psi_z)$

We want to define a transformation function $T(\psi_\chi, \psi_y, \psi_z)$, Equation (5). In particular, we define our transformation using the exponential function. Hence,

$$T(\psi_\chi, \psi_y, \psi_z) = (e^{\psi \cdot \vec{q}_1}, \dots, e^{\psi \cdot \vec{q}_n}) = (e^{v_1}, \dots, e^{v_n}), \quad (7)$$

where $\psi = (\psi_\chi, \psi_y, \psi_z)$ and $\vec{q}_i = (\chi_i, y_i, z_i)$. Additionally, a particular user may also want to establish specific weights for each of the deformation handles, so their influence regions can be adapted. In order to add this new property, we modify the original transformation function in Equation (7) to take into account an initial fixed user-defined point $s \in \mathbb{R}_+^n \cup \{\mathbf{0}\}$. Hence, the final transformation function is the following

$$T(\psi_\chi, \psi_y, \psi_z) = \sum_{i=1}^n s_i + e^{\psi \cdot \vec{q}_i}. \quad (8)$$

4.1.1. Proof of Existence

It can be shown that the System (6), when $T(\psi_\chi, \psi_y, \psi_z)$ is defined by Equation (8), always has a solution. The replacement of Equation (8) into System (6) leads to the new arranged system

$$\sum_{i=1}^n e^{\psi \cdot \vec{q}_i} \vec{q}_i = \mathbf{r}, \quad (9)$$

where $\mathbf{r} = -\sum_{i=1}^n s_i \vec{q}_i$. Hence, it is necessary to show that $\forall \mathbf{r} \in \mathbb{R}^3, \exists \psi = (\psi_\chi, \psi_y, \psi_z)$ such that the System (9) is solved.

Proposition 1. $\forall \mathbf{r} \in \mathbb{R}^3, \exists \psi = (\psi_\chi, \psi_y, \psi_z)$ so that $\sum_{i=1}^n e^{\psi \cdot \vec{q}_i} \vec{q}_i = \mathbf{r}$.

We defer the proof to Appendix A.

4.2. Implementation Details

The transformation function in Equation (8) converts the System (6) into a nonlinear one. Consequently, we can not compute an analytic solution for it, so it has to be solved by means of a numerical method. We use the Newton Method [11], which iteratively approximates the solution of nonlinear systems of equations, to solve the System (9). Each iteration k of this method consists in solving the following matrix system

$$\psi^k = \psi^{k-1} - (J(\psi^{k-1})^{-1})^\top \cdot f(\psi^{k-1}),$$

where ψ^{k-1} is the solution of the previous iteration, J is the jacobian matrix of the system and $f(\psi^{k-1}) = \sum_{i=1}^n e^{\psi^{k-1} \cdot \vec{q}_i} \vec{q}_i - \mathbf{r}$ is the system itself. The jacobian matrix is the one in Expression (A.1). We consider that we have reached a precise enough solution when $\sum_{i=1}^n e^{\psi \cdot \vec{q}_i} \vec{q}_i - \mathbf{r} \leq \epsilon$, for $\epsilon = 10^{-6}$.

To find the first estimation for the Newton Method in \mathbf{p}_0 , we know that the first two terms of the Taylor series for an exponential function of the form e^x are $e^x \simeq 1 + x$. Therefore, we use this estimation to approximate the exponential term of the transformation function $T(\psi_\chi, \psi_y, \psi_z)$ as follows

$$T(\psi_\chi, \psi_y, \psi_z) = \sum_{i=1}^n s_i + e^{\psi \cdot \vec{q}_i} \simeq \sum_{i=1}^n s_i + 1 + \psi \cdot \vec{q}_i.$$

This simplification allows us to linearise the System (9) into $\sum_{i=1}^n (1 + \boldsymbol{\psi} \cdot \vec{\mathbf{q}}_i) \vec{\mathbf{q}}_i = \mathbf{r}$. The resulting triplet $\boldsymbol{\psi} = (\psi_x, \psi_y, \psi_z)$ is used as the initial guess $\boldsymbol{\psi}^0$ to start the iterative Newton Method.

Every time the method achieves a good set of coordinates for a given model point \mathbf{p}_i , those neighbours of \mathbf{p}_i which do not yet have their coordinates computed are sorted from the nearest to the farthest to \mathbf{p}_i and queued. Then, each time that a point \mathbf{p}_j is stacked out, the Newton Method is initialised with the triplet $\boldsymbol{\psi} = (\psi_x, \psi_y, \psi_z)$ of its closest neighbour among those that already have a good set of coordinates computed.

We use the *jacobian distance* to select the closest point to \mathbf{p}_j . This metric defines the closest neighbour \mathbf{p} to \mathbf{p}_j as the one that minimises the distance $d_j = \mathbf{p}_j^\top \cdot J(\boldsymbol{\psi}_\mathbf{p})^\top \cdot J(\boldsymbol{\psi}_\mathbf{p}) \cdot \mathbf{p}_j$, where $J(\boldsymbol{\psi}_\mathbf{p})$ is the jacobian matrix at the model point \mathbf{p}_j by means of the triplet $\boldsymbol{\psi}_\mathbf{p} = (\psi_x, \psi_y, \psi_z)$, which has been obtained during the process to compute the coordinates of \mathbf{p} .

5. S-Celestial Coordinates

S-Celestial Coordinates derive from the R^3 interpretation in Section 3. They are based on driving the center of mass $\boldsymbol{\ell}$ of a set of n vectors $\mu_i \vec{\mathbf{q}}_i$ to the point \mathbf{p} . The main idea consists in using a wrapping Sphere parameterized by its center \mathbf{c} , see Figure 1, that defines these n vectors by giving more importance to the different orientations of the projected vectors as a function of the location of \mathbf{c} . In this way, the center of mass $\boldsymbol{\ell}$ is also a function of \mathbf{c} , $\boldsymbol{\ell} = f(\mathbf{c})$. The overall scheme is based on a demonstration that shows that a point \mathbf{c} such that $\boldsymbol{\ell} = f(\mathbf{c}) = \mathbf{0}$ always exists.

The wrapping sphere \mathcal{S}_c is centered at the point \mathbf{c} and its radius is $r = \lambda + \|(\mathbf{c} - \mathbf{p})\|$, with $\lambda > 1$. Hence, it completely wraps the unit sphere \mathcal{S}_p , as shown in Figure 1. The n vectors $\mu_i \vec{\mathbf{q}}_i$ are automatically defined by \mathcal{S}_c : their norm μ_i is the distance between \mathbf{p} and the intersection point between \mathcal{S}_c and the straight line coming from \mathbf{p} in the direction $\vec{\mathbf{q}}_i$, for all i , see Figure 1. In other words, \mathcal{S}_c defines the n oriented segments $\mu_i \vec{\mathbf{q}}_i$, which obviously have the center of mass

$$f(\mathbf{c}) = \frac{1}{\sum_{j=1}^n \mu_j} \sum_{i=1}^n \frac{\mu_i}{2} \mu_i \vec{\mathbf{q}}_i, \quad (10)$$

Our goal is to find a centerpoint \mathbf{c} such that this center of mass coincides with \mathbf{p} . Without loss of generality, we can assume that \mathbf{p} is at the origin. The point \mathbf{c} must therefore fulfil the following equation

$$\sum_{i=1}^n \mu_i^2 \vec{\mathbf{q}}_i = \mathbf{0}, \quad (11)$$

This equation has always a solution, as discussed in Section 5.2. Therefore, Equations (4) and (11) automatically define a valid set of coordinates based on the set of values $\beta_i = \mu_i^2$.

We detail the mathematical expression of the intersection points $\mu_i \vec{\mathbf{q}}_i$ in Section 5.1. Then, we demonstrate the existence of the proposed S-Celestial Coordinates in Section 5.2. Finally, Section 5.3 describes the methodology to numerically compute them.

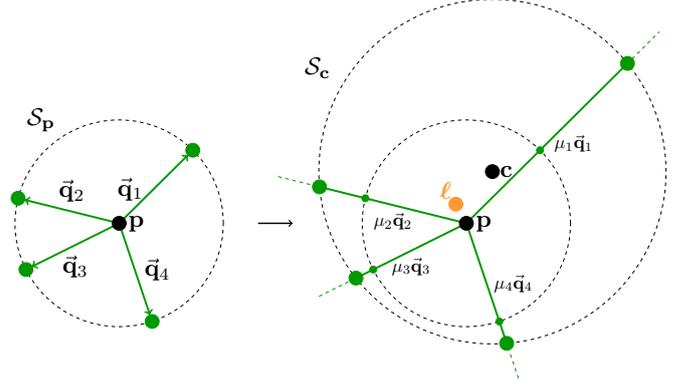


Figure 1: The unit sphere \mathcal{S}_p and the projected control points $\vec{\mathbf{q}}_i$ are shown on the left part of the image. The right side shows a random wrapping sphere \mathcal{S}_c , the intersected segments and their center of mass $\boldsymbol{\ell}$.

5.1. The Ray-Sphere Intersection

This section derives the mathematical expression to compute the intersection points between the rays defined above and the sphere \mathcal{S}_c . Without losing generality and for the sake of simplicity, let us assume that the point \mathbf{p} is located at the origin of the euclidean space, that is $\mathbf{p} = (0, 0, 0)$.

On the one hand, we name r_i the ray that corresponds to the projected control point $\vec{\mathbf{q}}_i$. It can be defined as

$$r_i = t \vec{\mathbf{q}}_i, \quad t > 0. \quad (12)$$

On the other hand, the sphere \mathcal{S}_c is described as

$$\mathcal{S}_c : (x - c_x)^2 + (y - c_y)^2 + (z - c_z)^2 = r^2. \quad (13)$$

Additionally, we know that the ray r_i intersects the sphere \mathcal{S}_c when $t = \mu_i$. Therefore, we substitute the value μ_i into Equation (12) and this equation into the Expression (13) to compute the intersection

$$(\mu_i \vec{\mathbf{q}}_i - \mathbf{c})^2 = (\lambda + \|\mathbf{c}\|)^2. \quad (14)$$

Now, we isolate μ_i from Equation (14)

$$\mu_i = \vec{\mathbf{q}}_i \cdot \mathbf{c} + \sqrt{(\vec{\mathbf{q}}_i \cdot \mathbf{c})^2 + \lambda^2 + 2\lambda \|\mathbf{c}\|}. \quad (15)$$

Notice that $\sqrt{(\vec{\mathbf{q}}_i \cdot \mathbf{c})^2 + \lambda^2 + 2\lambda \|\mathbf{c}\|} > \vec{\mathbf{q}}_i \cdot \mathbf{c}$, hence we always take the positive value of the square root to ensure the positivity of μ_i , which is required by the definition of the rays in Equation (12).

Once we have obtained the expression for the weights μ_i , we can substitute it in the original Function (10). Thus,

$$f(\mathbf{c}) = \frac{\sum_{i=1}^n \left(\vec{\mathbf{q}}_i \cdot \mathbf{c} + \sqrt{(\vec{\mathbf{q}}_i \cdot \mathbf{c})^2 + \lambda^2 + 2\lambda \|\mathbf{c}\|} \right)^2 \vec{\mathbf{q}}_i}{2 \sum_{j=1}^n \vec{\mathbf{q}}_j \cdot \mathbf{c} + \sqrt{(\vec{\mathbf{q}}_j \cdot \mathbf{c})^2 + \lambda^2 + 2\lambda \|\mathbf{c}\|}}. \quad (16)$$

5.2. Proof of Existence

We must prove that, for any point \mathbf{p} , there exists a point \mathbf{c} such that Equation (11) has a solution. Consider the function

$$g(\mathbf{c}) = \mathbf{c} - f(\mathbf{c}). \quad (17)$$

The idea is to look for a convex and compact domain \mathcal{B} such that

$$\begin{aligned} g : \mathcal{B} &\rightarrow \mathcal{B} \\ \mathbf{c} &\rightarrow g(\mathbf{c}). \end{aligned}$$

Once we have established \mathcal{B} , we can apply the Brouwer Fixed Point Theorem [12], which states that there exists a point $\mathbf{c}' \in \mathcal{B}$ such that $g(\mathbf{c}') = \mathbf{c}'$. Therefore $f(\mathbf{c}') = \mathbf{c}' - g(\mathbf{c}') = \mathbf{0}$.

Still assuming \mathbf{p} is at the origin and for any given point \mathbf{c} , let us define the vector $\vec{\mathbf{e}} = \frac{\mathbf{c}-\mathbf{p}}{\|\mathbf{c}-\mathbf{p}\|}$ such that $\mathbf{c} = \gamma\vec{\mathbf{e}}$ and $\gamma = \|\mathbf{c} - \mathbf{p}\|$. The following Lemmas and Propositions show that the compact and convex domain \mathcal{B} exists and, therefore, there also exists a point $\mathbf{c}' \in \mathcal{B}$ such that $f(\mathbf{c}') = \mathbf{0}$. The proofs of these Lemmas and Propositions can be found in Appendix B.

Lemma 1. Let $\mathcal{B}_c^1 = \{\mathbf{x} : d(\mathbf{x}, \frac{\mathbf{c}}{2}) \leq \frac{\gamma+\lambda}{2}\}$. Then, $f(\mathbf{c}) \in \mathcal{B}_c^1$.

Lemma 2. Let $\mathcal{B}_c^2 = \{\mathbf{x} : d(\mathbf{x}, \mathbf{c}) \leq \gamma\}$. Then, $\mathcal{B}_c^1 \cap \mathcal{B}_c^2 \neq \emptyset$.

Proposition 2. For any vector $\vec{\mathbf{e}}$, there exists a value $\bar{\gamma}$ such that, for any $\gamma \geq \bar{\gamma}$, $f(\mathbf{c}) \in \mathcal{B}_c^1 \cap \mathcal{B}_c^2$.

Proposition 3. Let $\bar{\gamma} = \max_{\vec{\mathbf{e}}} \bar{\gamma}(\vec{\mathbf{e}})$. $\forall \gamma > \bar{\gamma}$, then $g : \partial\mathcal{B}_{\mathbf{p}, \bar{\gamma}} \rightarrow \mathcal{B}_{\mathbf{p}, \bar{\gamma}}$.

Proposition 4. Let $\bar{\bar{\gamma}} = \max_{\mathcal{B}_{\mathbf{p}, \bar{\gamma}}} \|g(\mathbf{c})\|$. Then, $g : \mathcal{B}_{\mathbf{p}, \bar{\gamma}} \rightarrow \mathcal{B}_{\mathbf{p}, \bar{\bar{\gamma}}}$, $\mathcal{B}_{\mathbf{p}, \bar{\bar{\gamma}}} = \{\mathbf{x} : d(\mathbf{x}, \mathbf{p}) \leq \bar{\bar{\gamma}}\}$ and, therefore, there exists a solution $f(\mathbf{c}) = \mathbf{0}$ for $\|\mathbf{c} - \mathbf{p}\| \leq \bar{\bar{\gamma}}$.

5.3. Implementation Details

In this section, we present the methodology to solve Equation (11)

$$\sum_{i=1}^n \left(\vec{\mathbf{q}}_i \mathbf{c} + \sqrt{(\vec{\mathbf{q}}_i \mathbf{c})^2 + \lambda^2 + 2\lambda\|\mathbf{c}\|} \right)^2 \vec{\mathbf{q}}_i = \mathbf{0}, \quad (18)$$

which is a nonlinear system of equations. As a consequence, we again use the Newton Method to solve it. In the particular situation of the S-Celestial Coordinates, each iteration k of this method consists in solving the matrix system below

$$\mathbf{c}^k = \mathbf{c}^{k-1} - \left(J(f(\mathbf{c}^{k-1}))^{-1} \right)^T \cdot f(\mathbf{c}^{k-1}),$$

where \mathbf{c}^{k-1} is the solution of the previous iteration and J is the jacobian matrix of the System (18). This jacobian matrix is

$$J(f(\mathbf{c})) = 2 \begin{pmatrix} \sum t_{ix} q_{ix} & \sum t_{iy} q_{ix} & \sum t_{iz} q_{ix} \\ \sum t_{ix} q_{iy} & \sum t_{iy} q_{iy} & \sum t_{iz} q_{iy} \\ \sum t_{ix} q_{iz} & \sum t_{iy} q_{iz} & \sum t_{iz} q_{iz} \end{pmatrix},$$

where

$$\begin{aligned} t_{i\ell} &= \left(\mathbf{c}\vec{\mathbf{q}}_i + \sqrt{(\vec{\mathbf{q}}_i \mathbf{c})^2 + \lambda^2 + 2\lambda\|\mathbf{c}\|} \right) \cdot \\ &\cdot \left(q_{i\ell} + \frac{\mathbf{c}\vec{\mathbf{q}}_i q_{i\ell} + \frac{c_\ell \lambda}{\|\mathbf{c}\|}}{\sqrt{(\vec{\mathbf{q}}_i \mathbf{c})^2 + \lambda^2 + 2\lambda\|\mathbf{c}\|}} \right). \end{aligned}$$

We consider a precise enough solution when the function $f(\mathbf{c})$, in Expression (18), reaches $f(\mathbf{c}) \leq \epsilon$, for $\epsilon = 10^{-6}$.

We define the initial approximation for the Newton Method in \mathbf{p}_0 as the inverted center of mass $-\ell$ of the intersected segments $\mu_i \vec{\mathbf{q}}_i$ with a sphere centered on \mathbf{p} with a radius $r = \lambda$. That is, $\mathbf{c}^0 = -\ell$.

For subsequent points, we initialise the Newton Method with the center \mathbf{c} corresponding to its closest neighbour already visited. In the case of the S-Celestial Coordinates we use the euclidean distance to compute the closest neighbour to a given point \mathbf{p} .

6. Properties and Analysis

The two Point-Based Celestial Coordinate systems presented in the previous sections reproduce the identity and the unity, two basic properties of a Generalised Barycentric Coordinate system. They also produce very similar coordinates and, hence, deformations, as shown in Figure 2. However, the S-Celestial Coordinates are much more efficient in their computation, as Figure 3 illustrates. Moreover, S-Celestial Coordinates are rotational-invariant while T-Celestial Coordinates are not. We have presented T-Celestial Coordinates as an alternate scheme for the sake of comparison but S-Celestial Coordinates are the choice in practical deformation applications.

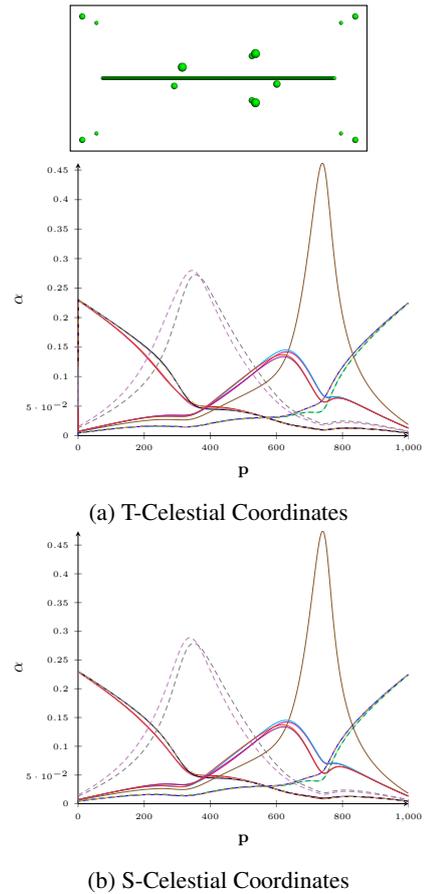


Figure 2: The T-Celestial Coordinate and the S-Celestial Coordinate functions corresponding to the initial configuration of a line surrounded by fifteen control points.

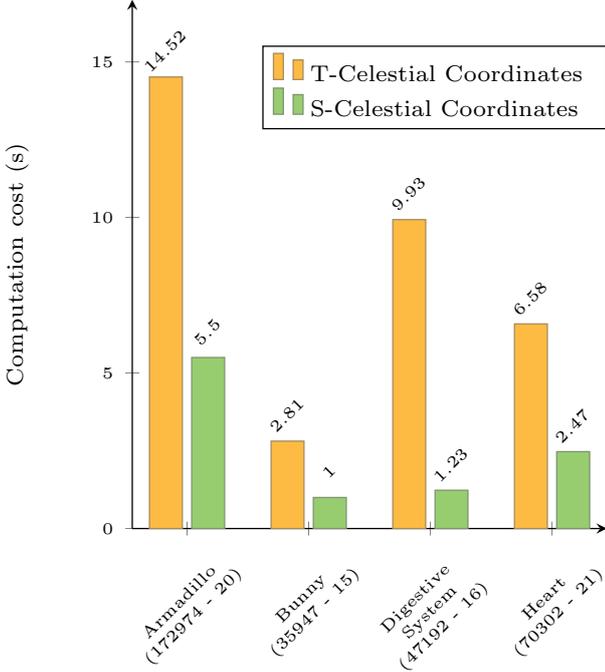


Figure 3: Comparison between the time, in seconds, used to compute the T-Celestial Coordinates and to compute the S-Celestial Coordinates. Below the name of each model are the number of points of that model and the number of control points of the configuration, in parenthesis.

Additionally, we want to emphasise some important features that they show:

- **Positivity:** the T-Celestial Coordinates and the S-Celestial Coordinates are positive inside the convex hull of the control points. This positivity is intrinsically ensured by the functions that define the two schemes, the T function and the center of the mass $f(\mathbf{c})$.
- **Smoothness:** the function $f(\mathbf{c})$ in Equation (10) is the restriction to $\mathbf{p} = \mathbf{0}$ of a function $F(\mathbf{p}, \mathbf{c})$ that implicitly defines the point \mathbf{c} corresponding to each position \mathbf{p} (because the μ_i and \vec{q}_i in Equation (10) depend on \mathbf{p}). Verifying the smoothness near a solution $(\bar{\mathbf{p}}, \bar{\mathbf{c}})$ of the coordinates is therefore equivalent to showing that $\left| \frac{\partial F}{\partial \mathbf{p}}(\bar{\mathbf{p}}, \bar{\mathbf{c}}) \right| \neq \mathbf{0}$. We are presently not able to offer a formal proof, but have exhaustively studied the behavior of the coordinates in numerous cases and have found it to be always smooth.
- **Point-based user-intended awareness:** the Point-Based Celestial Coordinates allow the user to locate the control points wherever he or she considers more appropriate to achieve the desired deformation. The only requirement is that the model to be deformed must lie inside the convex hull of these control points, which can always be ensured adding, if necessary, more control points.
- **Non-locality:** the T-Celestial Coordinates and the S-Celestial Coordinates suffer from global impact. The modification of any control point leads to a deformation that

affects all the points inside the convex hull of these handles. Obviously, the deformation has a bigger impact on the vicinity of the modified control point.

Both systems of coordinates suffer from global effect, as shown in Figure 4. This drawback is particularly evident in the second triplet of these images, where the control points around the ears of the *Bunny* also influence the chest and the back of the model. However, the flexibility of the deformation handles combined with the efficiency of the Space Deformation paradigm make the Point-Based Celestial Coordinates very suitable to be applied in a medical context. In medical applications, doctors want interactive response to their interaction and also appreciate the freedom to locate the deformation handles wherever they consider more suitable, so they can experiment or simulate soft organs and tissue deformations. Accordingly, Figures 5 and 6 show the effect of our coordinates on some soft organs. In both cases, the set of control points includes eight external points (not shown in the Figures) to ensure that the organs are inside the convex hull of all control points. Additionally, the video attached as supplementary material to the electronic version of this paper (see Figure 7) illustrates the interaction for placing control points around a model. This process consists in two steps: first the user defines a 3D line when he or she clicks on a pixel \mathcal{P} of the screen. The line is defined as the viewline going through \mathcal{P} and gets visible when the view is rotated. Finally, the user can place the new control point by clicking on this line.

7. Comparison with Existing Schemes

This section compares the T-Celestial Coordinates (TCC) and the S-Celestial Coordinates (SCC) with the Mean Value Coordinates (MVC), the Green Coordinates (GC) and the Bound-Ed Biharmonic Weights (BBW). All these systems reproduce the identity and the unity, interpolate the control points and are continuous, smooth and positive, although the MVC are only positive inside the kernel of the user-defined cage and the GC are continuous inside and outside the cage, but not on its boundary.

Our Point-Based Celestial Coordinates are cage-free methodologies while the MVC and the GC are used in a cage-based paradigm. This allows these three systems of coordinates to have an easy formulation and their computation is efficient. On the contrary, the BBW have the advantage of combining multiple types of handles, so they can use skeletons, cages and isolated points. This feature makes this methodology more flexible. However, it increases both the complexity and the cost of the computation of the weights, as Figure 8 shows. Moreover, the BBW need to discretise the deformation domain together with the model and, therefore, the weights depend on how this discretisation is done.

Cage-based methods suffer from non-locality of the deformation, a drawback that also affects our coordinate systems. In addition, our coordinates can present non monotonic influence. These disadvantages can be overcome combining these

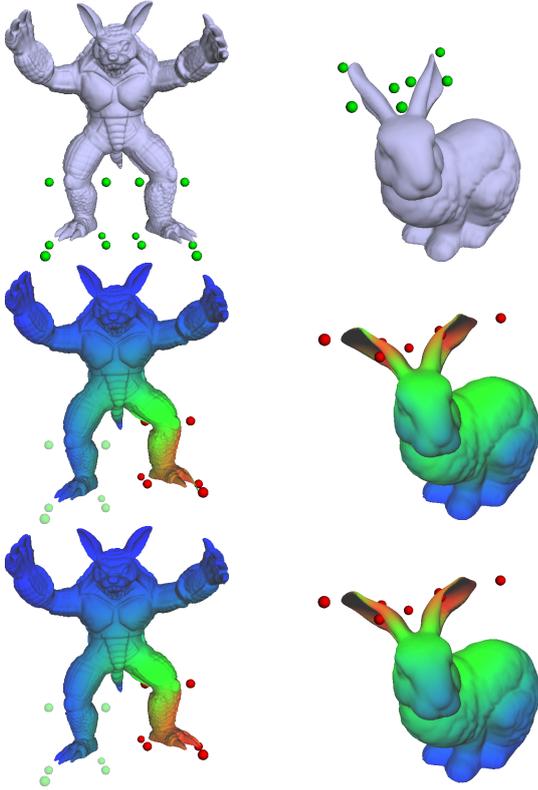


Figure 4: Example of two deformation processes to compare the T-Celestial Coordinates (TCC) with the S-Celestial Coordinates (SCC). The first row shows the initial configuration, the second one the TCC and the third row the SCC. Please, see video *Comparison between the T-Celestial Coordinates and the S-Celestial Coordinates* attached as supplementary material.

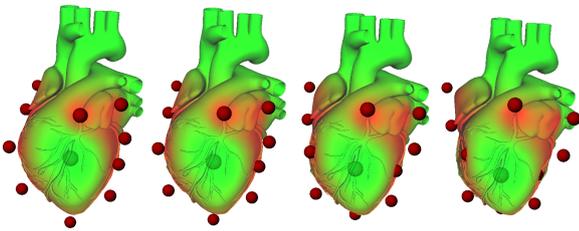


Figure 5: Heartbeat by means of the S-Celestial Coordinates. Please, see video *S-Celestial Coordinates: Animation of the Heart and the Digestive System* attached as supplementary material.

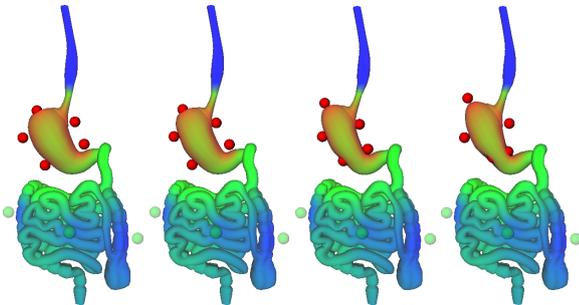


Figure 6: Stomach movement by means of the S-Celestial Coordinates. Please, see video *S-Celestial Coordinates: Animation of the Heart and the Digestive System* attached as supplementary material.

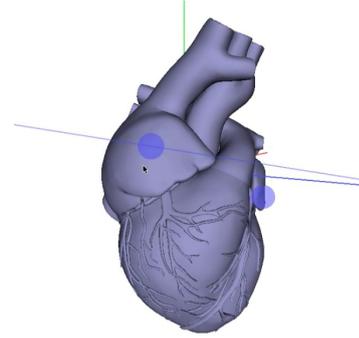


Figure 7: Frame of the video *Definition of the Control Points*, attached as supplementary material, that illustrates the process to place the control points around the model.

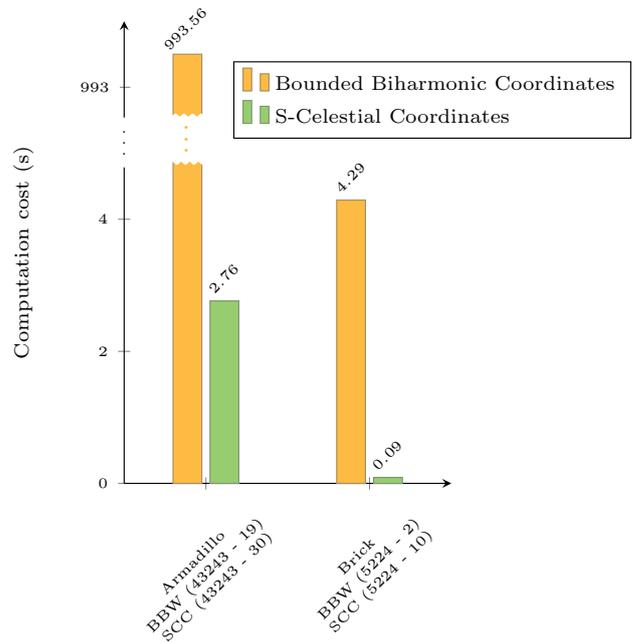


Figure 8: Comparison between the time, in seconds, used to compute the Bounded Biharmonic Weights and the S-Celestial Coordinates. Below the name of each model are the number of points of that model and the number of control points of both configurations, in parenthesis. The examples are those provided by the *libigl* library [13].

schemes with similar techniques to the multi-cage system [8]. In contrast, the BBW have been experimentally proven local.

Although the TCC and the SCC allow the user to locate the control points wherever he or she considers more appropriate to achieve the desired deformation, they can distort the local details of the deformed model. A similar situation also occurs with the MVC. The GC solve this disadvantage by means of the computation of a convex combination of the normals of the cage, which slightly increases their complexity, as shown in Figure 9. The BBW also tend to preserve these details better due to the customised energy minimisation performed to compute their weights.

Table 1 summarises the exposed features for the MVC, the GC, the BBW, the TCC and the SCC. Figure 10 shows two de-

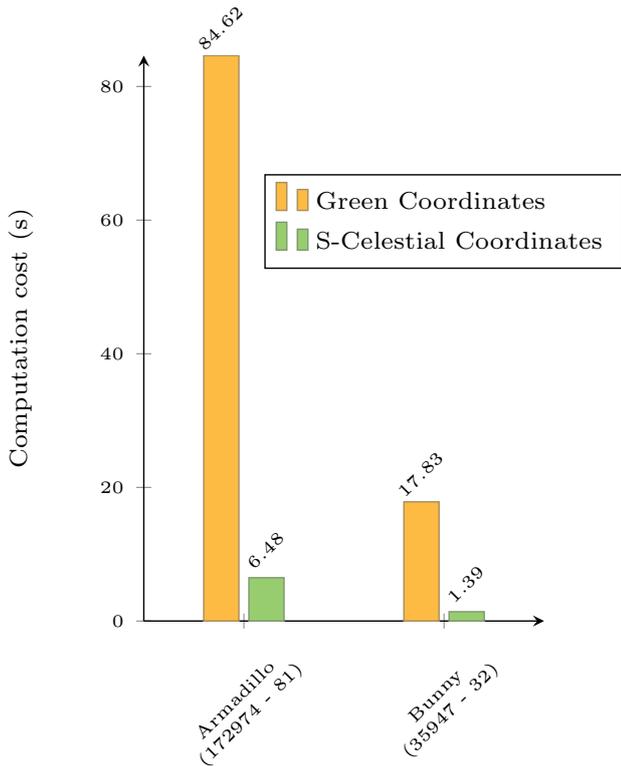


Figure 9: Comparison between the time, in seconds, used to compute the Green Coordinates and the S-Celestial Coordinates. Below the name of each model are the number of points of that model and the number of control points, in parenthesis.

formations of the *Armadillo* model: the first one is driven by the SCC and the second one has been performed by means of the BBW. The BBW deformation uses fifteen control points that shape a skeleton and four isolated handles located in the tail and the ears of the model. Our deformation uses thirty control points. The first eight are used to create a large convex hull that encloses the *Armadillo*. Then, the nineteen control points used in the BBW example are also added to our deformation. Finally, we have added three additional control points at the nose and at the end of the feet of the model. Notice that the SCC can achieve a similar deformation to that shown in the BBW example, especially at the legs, the ears and the mouth of the model. However, the arms cannot be emulated. This situation is caused by the different kind of handles used in each deformation. The BBW deforms the arms by means of a skeleton while our method deforms them by the use of isolated control points, so it cannot completely mimic large bones transformations. Instead, the presented systems of coordinates produce flexible and plausible deformations of soft models, as shown in Figures 5 and 6.

Figure 11 presents a comparison between the GC and the SCC. Observe that GC use a skeleton-adapted cage with handles also on their face normals. On the other hand, SCC are not aware of the object skeleton, showing a better performance in skeleton-free objects like those in Figures 5 and 6. They also have a much simpler way to define control points. We have fo-

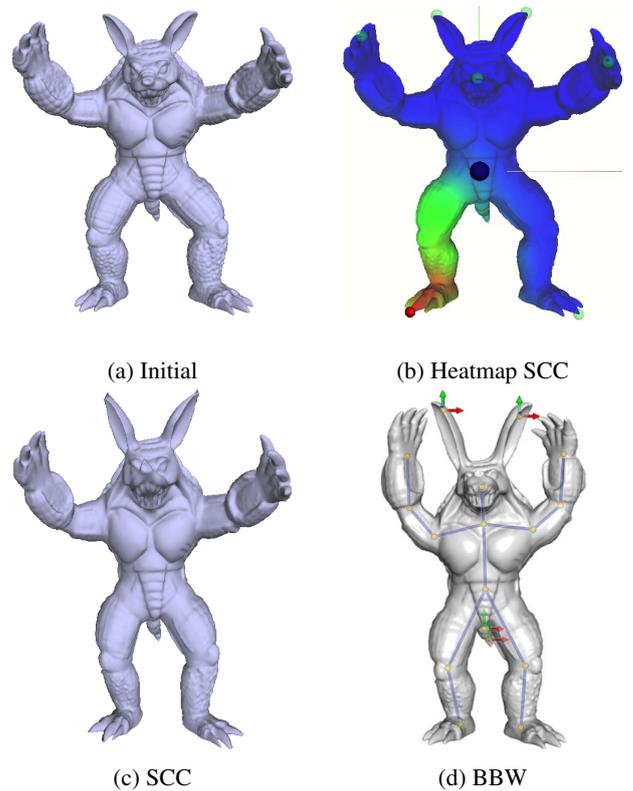


Figure 10: Comparison between the deformation by means of the Bounded Biharmonic Weights and the S-Celestial Coordinates. The image in (d) has been obtained from [9]. Please, see video *Comparison between the Bounded Biharmonic Weights and the S-Celestial Coordinates* attached as supplementary material.

cused on the leg deformation in (c) to remark the simplicity of the SCC deformations while making it independent from potential global behaviour effects. The heatmap of the deformation is shown in (b).

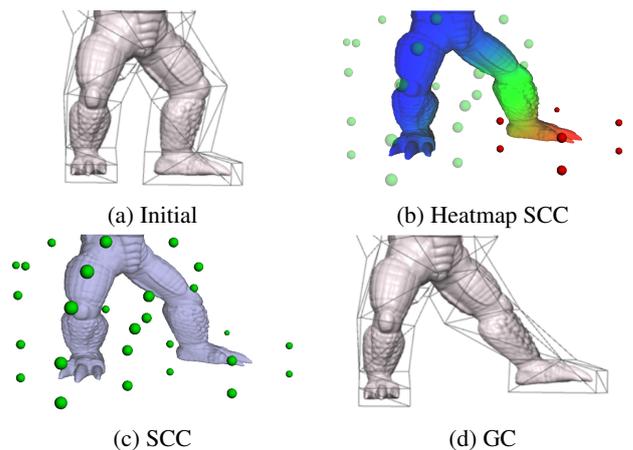


Figure 11: Comparison between the deformation by means of the Green Coordinates and the S-Celestial Coordinates. The images in (a) and (d) have been obtained from [6].

All the experiments have been executed in a PC with an Intel Core i7 CPU at 3.3 GHz, a NVIDIA GeForce GTX 470, 6

	MVC	GC	BBW	Point-Based Celestial Coordinates	
				TCC	SCC
Well-defined	✓	✓	✓	✓	✓
Continuous and smooth	✓	✓	✓	✓	✓
Multiple handles	✗	✗	✓	✗	✗
Cage-free	✗	✗	✓	✓	✓
Efficient	✓	✓	✗	✓	✓
Simple	✓	✗	✗	✓	✓
Do not need discretisation	✓	✓	✗	✓	✓
Local	✓	✓	✓	✓	✓
Detail-preserving	✗	✓	✓	✗	✗

Table 1: Comparison of the attributes of the Mean Value Coordinates, the Green Coordinates, the Bounded Biharmonic Coordinates, the T-Celestial Coordinates and the S-Celestial Coordinates.

GB of RAM and running Debian 8.0.

7.1. Limitations

Both proposed systems of Point-Based Celestial Coordinates present two main limitations. They inherit the globality of the space deformation techniques and they do not ensure the preservation of the local details on the deformed model. Their non-local behaviour is evident in skeleton-based examples as the ones shown in Figures 10 and 11, where deformation of certain body parts can affect other unwanted regions. These disadvantages could be overcome combining our schemes with other methodologies similar to *Cages [8], as already mentioned. We plan to address these aspects in our future work.

8. Conclusions and Future Work

We have presented the Celestial Coordinates, a family of positive Generalised Barycentric Coordinate systems that compute the coordinates of a point \mathbf{p} with respect to the control points \mathbf{v}_i by means of the projection of these handles over the *celestial sphere* of \mathbf{p} . We focus our attention specifically on the Point-Based Celestial Coordinates, a subfamily of Celestial Coordinates that aggregates those schemes that do not need any kind of connectivity between the deformation handles. Particularly, this group includes the T-Celestial Coordinates and the S-Celestial Coordinates.

The T-Celestial Coordinates and the S-Celestial Coordinates are efficient, positive, smooth and interpolate the control points. Additionally, the S-Celestial Coordinates are also rotational-invariant. Although both schemes suffer from global effect, the presented Point-Based Celestial Coordinates show an important improvement over the classic cage-based systems such as the Mean Value Coordinates and the Green Coordinates. They make the deformation process much more flexible due to their cage-free paradigm. Moreover, they can carry out similar deformations to those achieved by the Bounded Biharmonic Weights with just a few more handles. In addition, we have shown that they are a good tool to perform flexible, efficient, pleasant and plausible deformations of organs, tissues and medical models.

We plan to improve the presented techniques in three main directions. The first one refers to the local effect of the Celestial Coordinates. The second one aims to study the flexibility in adding and removing control points and its impact on the interactivity of the system. Finally, the third point concerns the study of the dimension-independence of the presented systems of coordinates.

Acknowledgments

The work presented in this paper has been partially supported by the project TIN2014-52211-C2-1-R of the Spanish *Ministerio de Economía y Competitividad* and *fondos FEDER*.

Appendix A. Existence of the T-Celestial Coordinates

This Appendix is devoted to prove Proposition 1, which is enunciated in Section 4 and states as follows:

Proposition 1. $\forall \mathbf{r} \in \mathbb{R}^3, \exists \boldsymbol{\psi} = (\psi_x, \psi_y, \psi_z)$ so that $\sum_{i=1}^n e^{\boldsymbol{\psi} \cdot \vec{\mathbf{q}}_i} \vec{\mathbf{q}}_i = \mathbf{r}$.

Proof Let us define the function $g(\boldsymbol{\psi}) = \sum_{i=1}^n e^{\boldsymbol{\psi} \cdot \vec{\mathbf{q}}_i} \vec{\mathbf{q}}_i$ from \mathbb{R}^3 to \mathbb{R}^3 . We need to demonstrate that it is surjective.

We first compute the partial derivatives of $g(\boldsymbol{\psi})$ as follows

$$\begin{aligned} \frac{\partial g(\boldsymbol{\psi})}{\partial \psi_x} &= \sum_{i=1}^n q_{ix} e^{\boldsymbol{\psi} \cdot \vec{\mathbf{q}}_i} \vec{\mathbf{q}}_i \\ \frac{\partial g(\boldsymbol{\psi})}{\partial \psi_y} &= \sum_{i=1}^n q_{iy} e^{\boldsymbol{\psi} \cdot \vec{\mathbf{q}}_i} \vec{\mathbf{q}}_i \\ \frac{\partial g(\boldsymbol{\psi})}{\partial \psi_z} &= \sum_{i=1}^n q_{iz} e^{\boldsymbol{\psi} \cdot \vec{\mathbf{q}}_i} \vec{\mathbf{q}}_i. \end{aligned}$$

Then, the corresponding jacobian matrix $J(\boldsymbol{\psi})$ is the one shown below

$$J(\boldsymbol{\psi}) = \begin{pmatrix} \left[\frac{\partial g(\boldsymbol{\psi})}{\partial \psi_x} \right]_x & \left[\frac{\partial g(\boldsymbol{\psi})}{\partial \psi_y} \right]_x & \left[\frac{\partial g(\boldsymbol{\psi})}{\partial \psi_z} \right]_x \\ \left[\frac{\partial g(\boldsymbol{\psi})}{\partial \psi_x} \right]_y & \left[\frac{\partial g(\boldsymbol{\psi})}{\partial \psi_y} \right]_y & \left[\frac{\partial g(\boldsymbol{\psi})}{\partial \psi_z} \right]_y \\ \left[\frac{\partial g(\boldsymbol{\psi})}{\partial \psi_x} \right]_z & \left[\frac{\partial g(\boldsymbol{\psi})}{\partial \psi_y} \right]_z & \left[\frac{\partial g(\boldsymbol{\psi})}{\partial \psi_z} \right]_z \end{pmatrix}, \quad (\text{A.1})$$

Equation (B.4), so

$$[f(\mathbf{c})]_x > -\frac{\lambda}{2} + \frac{\sum_{j=\ell+1}^n (\gamma q_{j_x} + \sqrt{(\gamma q_{j_x})^2 + \lambda^2 + 2\lambda\gamma})^2 q_{j_x}}{2 \sum_{k=1}^n \gamma q_{k_x} + \sqrt{(\gamma q_{k_x})^2 + \lambda^2 + 2\lambda\gamma}}.$$

Since $q_{j_x} \in (0, 1]$, the sum $\frac{\sum_{j=\ell+1}^n (\gamma q_{j_x} + \sqrt{(\gamma q_{j_x})^2 + \lambda^2 + 2\lambda\gamma})^2 q_{j_x}}{2 \sum_{k=1}^n \gamma q_{k_x} + \sqrt{(\gamma q_{k_x})^2 + \lambda^2 + 2\lambda\gamma}}$ is always positive. Additionally, the numerator is squared, so it grows faster than the denominator when γ also grows and the component $[f(\mathbf{c})]_x$ increases without bound as the value γ increases. Therefore, at a certain value γ , $[f(\mathbf{c})]_x$ overtakes a_x , which implies that the function $f(\mathbf{c})$ lies inside the ball $\mathcal{B}_{\mathbf{c}}^2$. Hence, there exists a bounding value $\bar{\gamma}$ such that, $\forall \gamma \geq \bar{\gamma}$, then, $f(\mathbf{c}) \in \mathcal{B}_{\mathbf{c}}^1 \cap \mathcal{B}_{\mathbf{c}}^2$.

Proposition 3. *Let $\bar{\gamma} = \max_{\vec{\mathbf{e}}} \bar{\gamma}(\vec{\mathbf{e}})$. $\forall \gamma > \bar{\gamma}$, then $g : \partial\mathcal{B}_{\mathbf{p}, \bar{\gamma}} \rightarrow \mathcal{B}_{\mathbf{p}, \bar{\gamma}}$.*

Proof By definition, $\bar{\gamma}$ is the maximum bounding value $\bar{\gamma}$ among all the vectors $\vec{\mathbf{e}}$. Since, $\bar{\gamma} \geq \bar{\gamma}$ for any $\bar{\gamma}$ and $\vec{\mathbf{e}}$, by Proposition 2, $f(\mathbf{c}) \in \mathcal{B}_{\mathbf{c}}^1 \cap \mathcal{B}_{\mathbf{c}}^2$. Now, from Equation (17), $g(\mathbf{c}) = \mathbf{c} - f(\mathbf{c})$. Hence, its magnitude is $\|g(\mathbf{c})\| = \|\mathbf{c} - f(\mathbf{c})\| \leq \bar{\gamma}$, which proves that $g(\mathbf{c}) \in \mathcal{B}_{\mathbf{p}, \bar{\gamma}}$.

Proposition 4. *Let $\bar{\bar{\gamma}} = \max_{\mathcal{B}_{\mathbf{p}, \bar{\gamma}}} \|g(\mathbf{c})\|$. Then, $g : \mathcal{B}_{\mathbf{p}, \bar{\gamma}} \rightarrow \mathcal{B}_{\mathbf{p}, \bar{\bar{\gamma}}}$, $\mathcal{B}_{\mathbf{p}, \bar{\bar{\gamma}}} = \{\mathbf{x} : d(\mathbf{x}, \mathbf{p}) \leq \bar{\bar{\gamma}}\}$ and, therefore, there exists a solution $f(\mathbf{c}) = \mathbf{0}$ for $\|\mathbf{c} - \mathbf{p}\| \leq \bar{\bar{\gamma}}$.*

Proof We must deal with two different situations. On the one hand, $\mathbf{c} \in \mathcal{B}_{\mathbf{p}, \bar{\gamma}} - \mathcal{B}_{\mathbf{p}, \bar{\gamma}}$ whose proper behaviour has been demonstrated in Proposition 3. On the other, $\mathbf{c} \in \mathcal{B}_{\mathbf{p}, \bar{\gamma}}$. We define $\bar{\bar{\gamma}}$ as the maximum magnitude that the function $g(\mathbf{c})$ can reach for any point $\mathbf{c} \in \mathcal{B}_{\mathbf{p}, \bar{\gamma}}$. Hence, the ball $\mathcal{B}_{\mathbf{p}, \bar{\bar{\gamma}}}$ also includes $g(\mathbf{c})$ when $\mathbf{c} \in \mathcal{B}_{\mathbf{p}, \bar{\gamma}}$. Consequently, $\bar{\bar{\gamma}}$ is the minimum radius that ensures $g : \mathcal{B}_{\mathbf{p}, \bar{\gamma}} \rightarrow \mathcal{B}_{\mathbf{p}, \bar{\bar{\gamma}}}$ and, therefore, there exists a solution $f(\mathbf{c}) = \mathbf{0}$ for $\|\mathbf{c} - \mathbf{p}\| = \bar{\bar{\gamma}}$.

We have constructed a ball $\mathcal{B}_{\mathbf{p}, \bar{\bar{\gamma}}}$ such that g maps it unto itself. Therefore the Brouwer Fixed Point Theorem guarantees that $g(\mathbf{c}) = \mathbf{c}$ has a solution within the ball, which in turn proves the existence of the S-Celestial Coordinates for any point inside the convex hull of the control points.

References

- [1] Floater MS. Mean value coordinates. *Computer Aided Geometric Design* 2003;20(1):19–27.
- [2] Floater MS, Kós G, Reimers M. Mean value coordinates in 3d. *Computer Aided Geometric Design* 2005;22(7):623–31.
- [3] Ju T, Schaefer S, Warren J. Mean value coordinates for closed triangular meshes. *ACM Transactions on Graphics* 2005;24(3):561–6.
- [4] Joshi P, Meyer M, DeRose T, Green B, Sanocki T. Harmonic coordinates for character articulation. *ACM Transactions on Graphics* 2007;26(3).
- [5] Lipman Y, Kopf J, Cohen-Or D, Levin D. Gpu-assisted positive mean value coordinates for mesh deformations. In: *Proceedings of the 5th Eurographics Symposium on Geometry Processing. SGP '07; Eurographics Association; 2007, p. 117–23.*

- [6] Lipman Y, Levin D, Cohen-Or D. Green coordinates. *ACM Transactions on Graphics* 2008;27(3):78:1–78:10.
- [7] Li Z, Levin D, Deng Z, Liu D, Luo X. Cage-free local deformations using green coordinates. *The Visual Computer* 2010;26(6-8):1027–36.
- [8] González-García F, Paradas T, Coll N, Patow G. *cages: A multilevel, multi-cage-based system for mesh deformation. *ACM Transactions on Graphics* 2013;32(3):24:1–24:13.
- [9] Jacobson A, Baran I, Popović J, Sorkine O. Bounded biharmonic weights for real-time deformation. *ACM Transactions on Graphics* 2011;30(4):78:1–8.
- [10] Langer T, Belyaev A, Seidel HP. Spherical barycentric coordinates. In: *Proceedings of the 4th Eurographics Symposium on Geometry Processing. SGP '06; Eurographics Association; 2006, p. 81–8.*
- [11] Hildebrand FB. *Introduction to numerical analysis.* General Publishing Company; 1956.
- [12] Brouwer LEJ. Über abbildung von mannigfaltigkeiten. *Mathematische Annalen* 1911;71(1):97–115.
- [13] Jacobson A, Panozzo D, et al. libigl: A simple C++ geometry processing library. 2015. [Http://libigl.github.io/libigl/](http://libigl.github.io/libigl/).