Cage based deformations: a survey

Jesús R. Nieto, Antonio Susín

Abstract Cage based deformation techniques aims to be an easy to use tool for graphics modeling, texturing and animation. In this paper we describe the most important methods, their foundations, and the desirable properties that they should satisfy. We also present a comparative to show the strong and weak points of each one, taking into account their distinctive utilities. Finally, we discuss some applications that exploit cage capabilities in order to create a more complex deformation system or to simplify other deformation techniques.

1 Introduction

Mesh deformation is a common process in geometry modeling and computer animation. Modeling can be very accurate, like in engineering design, or be flexible to allow the artist to freely express his creative ideas. Similarly, in computer animation we may want a realistic behavior, simulating physics, or rather a stylized and artistic animation far from what is really possible. But, regardless of the preferred approach, we need flexible tools for mesh deformation to achieve the desired results easily. In the past years there have been many efforts in this direction, from different points of view: Free-form deformation (FFD), generalized barycentric coordinates, Radial Basis Functions (RBF), curve based deformation, skeletons and physics simulation [1, 2, 3, 20, 26, 27, 29, 32]

Character articulation, also called rigging, has a significant place in the field of mesh deformations; It is an important component in high-end applications used in film and audiovisual content. Professional softwares, specially Softimage XSI, Maya and 3DStudio, provide a wide range of character articulation methods. Some

Jesús R. Nieto
Universitat Politecnica de Catalunya, Barcelona, e-mail: jrodrigueznieto@gmail.com

Antonio Susín
Universitat Politecnica de Catalunya, Barcelona e-mail: toni.susin@upc.edu

classic examples are Enveloping[20] and blend shapes[17], but there are also many chainnable deformers that achieve amazing results.

Regarding character animation, there are a number of constraints that some of the previous methods do not fulfill. Firstly we need a deformation method able to work in real-time, for interactive applications, which limits the computation time. Also, it is desirable to have a convenient and easy to use system which makes it simple for a broad array of users to get quickly familiar with it. Cage based methods are good candidates for this purpose. Even if nowadays these methods are not the most used in professional animation applications, they have undergone significant developments so that could push forward the usual rigging techniques.

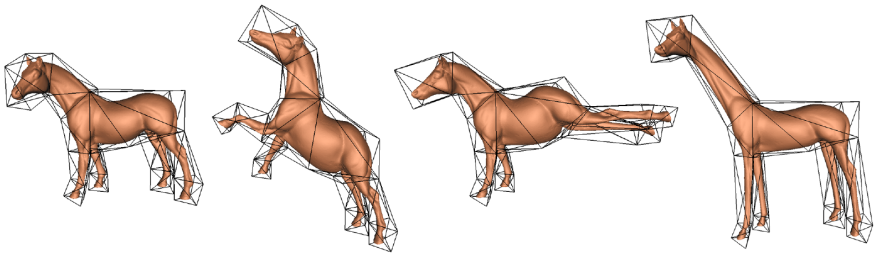


Fig. 1 An object wrapped by a cage. In the left, the cage and the object are in rest pose for coordinates computing. Then we can transform cage vertexes for producing a deformation in the volume and consequently in the object. From [18]

Research in cage based methods were born at the beginning of feature film animation. The pioneers were Sederberg and Parry[27], in 1986, with the Free Form Deformation. This development become popular for several reasons. First, because it offers a smooth and intuitive control over the character using limited parameters: free form lattice control points. Besides, the model to be deformed does not need to satisfy other geometric constraint apart from being inside the control lattice. On the other hand, this kind of deformer has a well number of drawbacks: if the deformations are complex, such as character articulated with several limbs, it becomes difficult or even impossible to implement. A lattice will never fit perfectly the character shape, since the topologic rigidity of lattice is not flexible enough for that, and combining several cages would be a hard task [16].

Next, we introduce the methods that will be the main line of discussion throughout the paper. We need a cage which better fits the character shape. The first complete environment that allow to do this is Mean Value Coordinates (a solution proposed and developed simultaneously by to papers: Floater et al.[9] and Ju et al. [18], in 2005). This method lead mesh deformation to the world of generalized barycentric coordinates. The original concepts were introduced by F. Möbius in 1827, and have been developed by many mathematicians since then [25, 33, 7, 6, 12, 24]. The main point to solve is the relationship between a cage and its interior. If there is an object inside a cage, we can deform it using the cage no matter the complexity of

the object. For this purpose, Floater [8] proposed to use the mean value theorem, but his formulation has some problems; the main one, being that the coordinates could be negative, which will produce anti-intuitive and annoying results.

Even if this problem only applies to certain cases, the relevance of this issue advises to discard this method for character animation, as Joshi [16] claims. He proposed an alternative solution adopting a slightly different approach which does not produce negative coordinates: Harmonic coordinates. This approach produces a more local deformation, which is a very positive feature, but this also has drawbacks, i. e. computation time and discretization accuracy.

Similarly, Lipman [21] proposes an improvement to Mean Value Coordinates that leads to positive coordinates. He uses GPU visibility render techniques for analyzing the volume inside the cage in order to cut off negative coordinates. In spite of the improvements of this method, also has some smoothness problems with concave shapes (as the original Mean Value Coordinates does).

Then, Lipman [22] realized that the surface details were not preserved, especially if the deformation is large, thereby suggesting that more data should be used for being able to relate the cage to the object. While Mean Value Coordinates and Harmonic Coordinates only use cage vertex positions, Lipman's Green Coordinates also uses face normals.

These are strictly cage based deformation methods, but there are other deformation methods that use cages in other ways. This is the case for Biharmonic weights [15] and subspace gradient domain deformations [14]. There have also been improvements and new developments for using these concepts, such as volume-preserving deformation [4], Cage-based deformation transfer [5] and skinning templates [19]

In the following section we are going to describe some basic concepts for deeper understanding cage based methods, which we analyze afterwards in section 3. Then, in section 4, the main conclusions are summarized and, finally, in the appendix we collect some pseudocode algorithms that can be very useful for a better understanding of these methods.

2 Barycentric coordinates and cage based deformation

Let us denote as a cage C any triangle mesh, or more generally a polyhedric mesh, convex or not, which is closed, and that envelop a model to be deformed. For perfect model fitting the cage must be topologically flexible, and may be manually or automatically generated [35]. As far as there is a well defined relationship between the cage surface and its inside volume, the deformation applied to the cage will also affect the volume, and therefore any object it contains. This procedure endows us with an easy to use control handles (cage vertexes) to deform whatever complex model inside the cage.

2.1 Barycentric coordinates definition

The first mathematical approach defining a cage-to-model relationship was introduced by Möbius in 1827 with barycentric coordinates for triangles [25]. The issue was set out in the following terms: Which weights w_1, w_2, w_3 must be given to the vertexes A, B, C of a triangle to obtain P as the center of gravity of these weights? Thus, point P is the barycenter, while the value of the vertex weights are the barycentric coordinates of P for each vertex. Generalizing the problem, v can be considered as the barycenter of points $v_1, \ldots, v_n$ if and only if

$$v = \frac{w_1(v)v_1 + \ldots + w_n(v)v_n}{w_1(v) + \ldots + w_n(v)}. \tag{1}$$

More precisely, these coordinates are homogeneous and need to be normalized by

$$\lambda_i(v) = \frac{w_i(v)}{\sum_i w_i(v)} \quad , \quad \sum_{i=0}^{n} \lambda_i(v) = 1. \tag{2}$$

Where all the values λ_i are between 0 and 1. For the case of triangles, there is also a relation between the coordinate w_i (from vertex v_i) of the interior point v and the area of the triangle $[v, v_{i+1}, v_{i+2}]$ (Fig. 2). Therefore, they are also called *area coordinates*. The barycentric coordinates are a linear transformation of Cartesian Coordinates over a triangle. Then, they vary linearly over the boundary, inside and outside the triangle. This means that we can use it as an interpolation function ϕ inside the polygon using the values defined at the vertexes $\phi(v_i)$ (eq. 3)

$$\phi(v) = \sum_{i=0}^{n} \lambda_i(v)\phi(v_i). \tag{3}$$

An example of this interpolation is Phong and Gouraud shading, a rendering algorithms widely used in computer graphics [12]. These interpolations can also be employed for geometry deformation. The coordinates define the relationship between the triangle vertexes and the triangle inside. After having distorted the triangle vertexes, we can relocate the inside points constrained with the same relationship (Fig. 2). In fact, by using identity as interpolation function $\phi(v_i) = v_i$ in equation 3, we have an interpolation of the vertex space positions.

2.1.1 Barycentric coordinates generalization

The above formulation works well with simplexes, but a number of difficulties arise when the issue is generalized to more complex polygons. Let us see now some of the many approaches that have been tried to implement such a generalization. The first attempt was proposed by Wachspress [33], who tried to deal with finite element methods. The goal of this kind of methods is to solve equations, approximating continuous values as a set of discrete points, usually distributed into a grid.

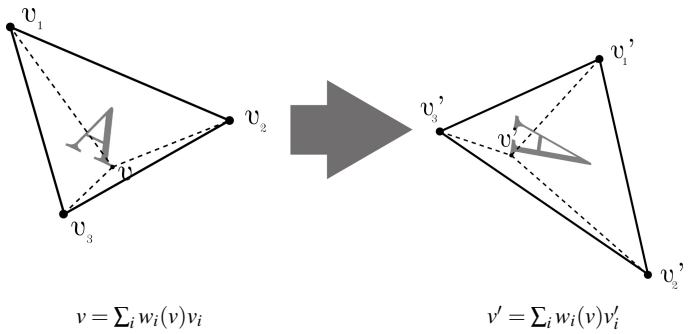


Fig. 2 The coordinates w_1, w_2, w_3 define the relationship between v_1, v_2, v_3 and the interior point v. After having distorted the triangle vertexes, the inside points are relocated constrained with the same relationship.

The Wachspress's definition is only suited to the case of convex polygons. Other studies have developed these methods for quasi-convex polygons [23] and arbitrary polygons [31].

We distinguish quasi-convex polygons from convex ones to denote this kind of polygons which are not convex but have a convex kernel, and therefore are convex in some sense. The kernel of a polygon is the region K of a polygon Ω such that, taking any point $v \in K$, for all vertex v_i of the cage, the segment $[v, v_i]$ only intersects with the polygon boundary at v_i.

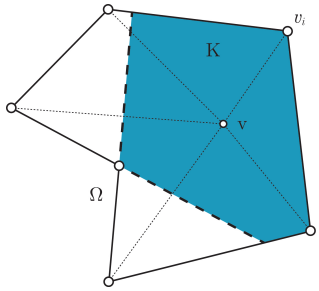


Fig. 3 The kernel of a polygon K of a polygon Ω

Another approach to barycentric coordinates generalization is point cloud interpolation. The objective of this method is to define the interior volume of the point's convex hull [28] [10] [11].

Finally, we consider that the most interesting approach as far as this paper is concerned, is piecewise surface interpolation [7, 8]. It is particularly relevant because it enables volume deformation by cage control. In the same way that a triangle is able to define a deformation for its support plane, a mesh can define one for its inside

volume. The most interesting works in this area have been done in turn, ending in Mean Value Coordinates, Harmonic Coordinates and Green Coordinates, that we discuss in the next section, but first we will summarize the main properties of cage based deformation methods.

2.2 Cage deformation desired properties

The final goal is to develop a procedure for producing natural and intuitive deformations by using some control handle vertexes that form a cage. For this purpose, some properties may be defined.

The first concept to be analyzed is *deformation domain*, which is the space region influenced by the cage. To start with, deformation needs to be well defined inside the mesh volume, without singular points. To this aim, we can constraint the object to be totally inside the cage. However if only a part of the object is aimed to deform, we need the cage to cover just that part, intersecting the object, which is a problem when the domain is not well defined also at the boundary and outside the cage. There are different approaches for outside coordinate's extension from cage, as we will see in the next sections.

Coordinates can be managed as a function, so that, if coordinates are well defined (ensuring continuity all over the domain) we also expect *smoothness*, which means continuity at first derivatives.

Interpolation, that is, coincidence values for the vertexes, is the main application. As long as interpolation is ensured, the fact that $\phi(v) = 1$ implies $\sum_{i=1}^{n} \lambda_i(v) = 1$, for all points v in the domain. This fact is known as *affine invariance* (eq. 3).

Besides, *local deformation* will be needed to restrict vertexes influence to their neighborhood only at local state. The implication follows that a control point (cage vertex) must cut off the influence of other control points over a part of the cage volume, if it is placed between them.

In deformation methods, *conformality* is an important property that relates transmission of local rotation transformations and, accordingly, surface detail preservation. While conformal mapping maps infinitesimal spheres to other infinitesimal spheres, quasi-conformal mapping is able to map infinitesimal spheres into infinitesimal ellipsoids (with a certain ratio between axis). This property works towards a natural deformation without losing shape semantic. (Fig. 4)

Another important property for character articulation is *positiveness*. This feature ensures that the coordinate values are positive in the entire domain, or at least in some controlled parts. In as much as this is achieved, deformation will be successfully intuitive.

Moreover, if we want to develop a real-time interactive deformation tool, *computation time* needs to be reduced as much as possible. All the implementations segment deformation process in two parts to break off the heaviest computations into a pre-process (getting coordinates), so that the actual application of deformation is isolated in a simple computation to achieve real-time. Using GPU parallel

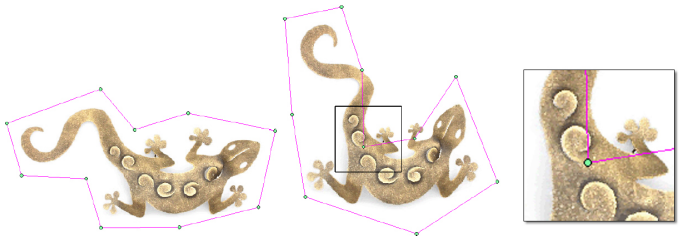


Fig. 4 Rotation transformation is naturally applied keeping the object shape semantic, extracted from [22]

techniques speed up the process, since transformations for each vertex can be done simultaneously. In order to develop such a low time-consuming computations we need to allocate in memory coordinates for every point of the object. Depending on the method, a fast access to memory for many little sets of data (sets of vertex coordinates) will be needed.

3 Mean Value Coordinates (MVC)

The first approach to MVC comes from Floater et al. [9] and Ju et al. [18]. Both studies aimed at looking for an interpolation method for surfaces. Initially, Floater [8] presented a 2D mean value coordinates over quasi-convex polygons, where the coordinates were well defined inside and outside the polygon. Even if there was a problem of discontinuity at boundaries, it could be solved easily [12] by extending coordinates from the interior domain. In this case, the coordinates were well defined, but it was not ensured that the values were necessarily positive for the entire domain. In fact, positive coordinates can only be ensured inside the kernel of the polygon (a convex or star-shaped one) [9]. Afterwards, Floater developed a 3D version of the algorithm, but the negativity problem remained unsolved. Then, Ju et al. developed another solution by using the Floater's 2D approach but addressing the issue from a slightly different point of view. Their attempt to solve the matter as a 3D interpolator provided us with a solution which is more robust than the 3D one presented by Floater (the pseudocode of Ju's MVC is included in the appendix). In Summary, these coordinates satisfy most of the properties enumerated in the previous section [18] (mainly the fact of being a good interpolation method [12]) but the lack of positiveness and locality are serious drawbacks for mesh deformation.

MVC uses mean value theorem to relate the points of a cage with its interior, which is oriented by harmonic coordinates theory. Next, we present some concepts that may help understanding the process, and also be helpful for discussing further methods.

3.1 Harmonic functions

The main problem to solve the desired cage-object relationship starts with a more theoretical problem, the approximation of harmonic functions by piecewise linear functions over triangulations, in such way that the injective property is preserved. An harmonic function u is a function over the reals for which the second derivative is continuous and satisfy Laplace's equation

$$\triangle u = 0 \Longleftrightarrow \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} = 0. \tag{4}$$

Dirichlet raised the problem of finding a relation between a continuous function f over a boundary $v|_{\partial\Omega}$ of a region Ω and its interior, by using a harmonic function u assuming as known the values of f at $v|_{\partial\Omega}$. Basically, he explores whether such a function u can exist and, if it does, if it is unique. The conclusion of such an attempt is to establish the existence of a function with these characteristics and named his solution as Dirichlet's principle.

Following Dirichlet development, Floater approximates a solution of u that satisfies the Dirichlet's boundary conditions, $u|_{\partial\Omega} = f_i$, by means of a linear piecewise function $u_{\mathscr{T}}$ over the triangulation. This leads us to a linear system where the values of $u_{\mathscr{T}}$ are at the inside vertexes of the triangulation T, (observe the similitude with eq. (3))

$$u_{\mathscr{T}}(v) = \sum_{i=1}^{k} \lambda_i u_{\mathscr{T}}(v_i) \tag{5}$$

3.2 Mean Value theorem

Floater carries out this approximation by using the mean value theorem, which states that for a circumference $B = B(v,r) \subset \Omega$ (where r is the radius of a circumference centered at point v, completely inside the region Ω) and its circumference perimeter Γ, the equation (6) approximates the function $u_{\mathscr{T}}$ fitting the Dirichlet's conditions, as is expressed in equation (5)

$$u(v) = \frac{1}{2\pi r} \int_{\Gamma} u(v|_{\partial\Omega}) dS. \tag{6}$$

It is also relevant to analyze the geometric interpretation. Recall *area coordinates*, we compute the i coordinate of interior point v as a ratio between the area of the opposite sub-triangle (v, v_{i+1}, v_{i+2}) from vertex v_i and the total triangle area. Similarly, there is also a ratio to compute Mean Value Coordinates, this time between partial perimeter of a circumference centered at point v, obtained from the projection of the cage segment $[v_i, v_{i+1}]$ over the circumference, and the total circumference perimeter. Note that Floater, as an extension of his analysis, uses the

angles between segments $[v, v_i]$ and $[v_i, v_{i+1}]$ to obtain the same ratio (Fig. 5 (a)). It is easy to see that, by considering angles (or perimeters) with sign there may be negative values for the coordinates. This is the case of concave polygons (Fig. 5 (b)). The following expressions are derived by Floater to obtain Mean Value Coordinates w_i and their normal form λ_i :

$$\lambda_i = \frac{w_i}{\sum_{j=1}^{k} w_j} \quad , \qquad w_i = \frac{tan(\alpha_{i-1}/2) + tan(\alpha_i/2)}{\|v_i - v_0\|} \tag{7}$$

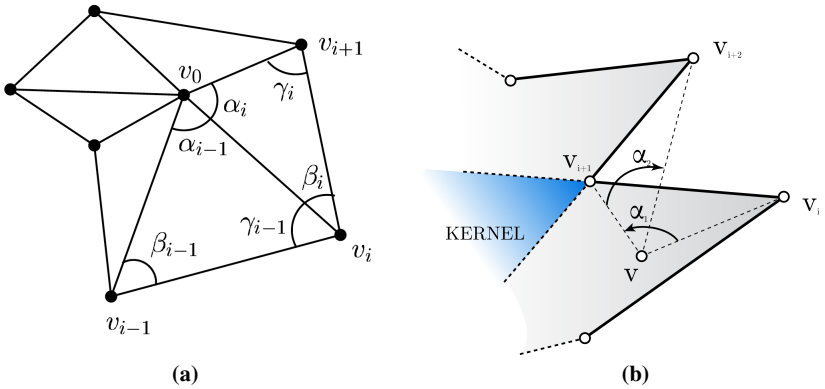


Fig. 5 (a) parameter interpretation to obtain mean value coordinates. It is easy to see that if we consider angles with sign there may be negative values for some coordinates, this is the case of concave polygons as in (b). The angle α_1 corresponding to segments $[v, v_i]$ and $[v_i, v_{i+1}]$ is positive while the angle α_2 defined from segments $[v, v_{i+1}]$ and $[v_{i+2}]$ is negative.

In a similar way, mean value coordinates in R^3 can be derived. In this case, the coordinates are obtained as the area ratio over a sphere. With a sphere S centered at point v, the coordinates of point v are equal to the ratio of the projected area of every cage simplex over the sphere and the total sphere surface area. To get a closed form expression, Floater uses other tools. He claims that, in fact, 2D coordinates can be computed as a proportion of normals integration of the projected cage piece over the circumference. Thus, we can integrate in 3D the normals over the partial sphere. The integration of sphere normals over the entire sphere is equal to 0, from which we can derive:

$$0 = \int_S n(p) = \int_S (p - v) = \sum_{T \in \mathcal{T}} \int_{T'} (p - v) \tag{8}$$

and define the Spherical Barycentric Coordinates of every partial sphere that comes from the cage triangles projection over the sphere. These coordinates can be used as a $u_{\mathcal{T}}$ approximation, which is what we need. The derivation of these coordinates in a closed form expression is given by:

$$0 = \sum_{i=1}^{n} \sum_{v_i \in T} \mu_{i,T} e_i = \sum_{i=1}^{n} w_i (v_i - v) \tag{9}$$

$$w_i = \frac{1}{r_i} \sum_{v_i \in T} \mu_{i,T} > 0 \quad , \qquad \mu_{i,T} = \frac{\beta_{jk} + \beta_{ij} n_{ij} \cdot n_{jk} + \beta_{ki} n_{ki} \cdot n_{jk}}{2 e_i \cdot n_j k} \tag{10}$$

Following the scheme of figure 6, where $\beta_{rs} \in (0, \pi)$ is the angle between two segments $[v, v_r]$ and $[v, v_s]$, and n_{rs} as the unitary vector of the face $[v, v_r, v_s]$ pointing into the tetrahedron. (r and s take the values of i, j, k)

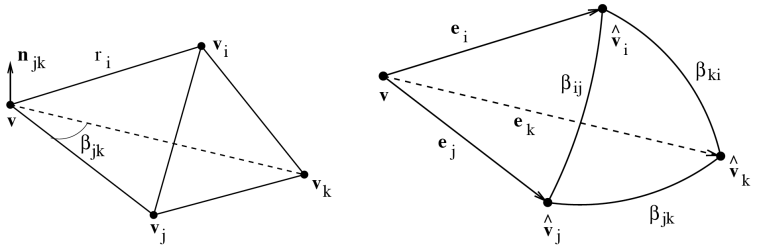


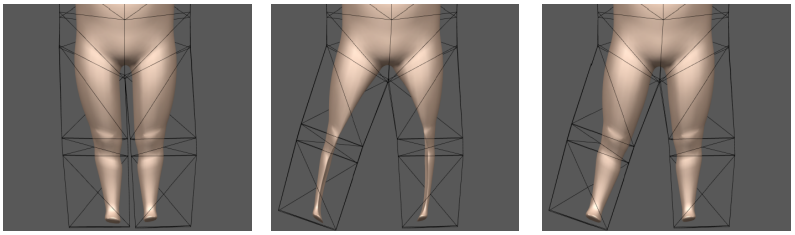
Fig. 6 a) tetrahedron b) "tetrahedron" defined by a spheric triangle. From [9]

Floater analyzes this case carefully and concludes that the equation (10) is not well defined on the boundary. Besides, he proves that the coordinates can be extended to the boundary and beyond without losing of continuity and smoothness. The closed form expressions proposed by Floater to obtain coordinates is the most appropriate context for computing with parallel GPU techniques.

4 Harmonic coordinates (HC)

Joshi et al. [16] developed another method to set the coordinates successfully avoiding some of the MVC drawbacks. Their concern was focussed on articulation in feature film animation, so that the mentioned issues were annoying for that purpose. The main problem is the fact that MVC are based in euclidean distances, which neglects the visibility between cage points and object points. Then, the deformation results are non-intuitive and require additional work to solve or to avoid the problem. Consider the case of figure 7: the cage's left limb acts over the object's left and right limbs, resulting a non-expected displacement.

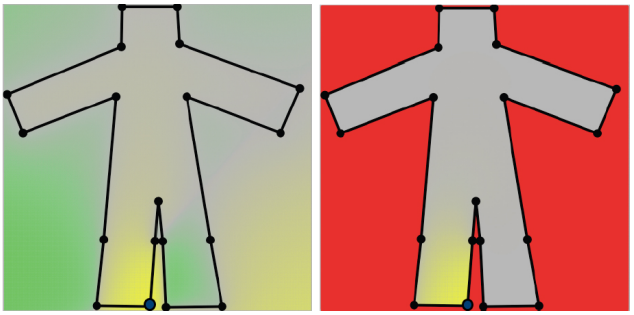
When analyzing further the matter, two important issues are found. Firstly, the *negative coordinates*. The influence of the vertexes of the cage's left limb delivers positive values for points inside the cage's left limb, but negative ones for the cage's right limb. Figure 8 show coordinate's value distribution for a marked vertex.



(a) Bind pose **(b) Mean Value Coordinates** **(b) Harmonic Coordinates**

Fig. 7 Comparative of MVC and HC in a conflictive deformation case of character articulation. From [16]

The second issue is related to *global influence*: every cage vertex affects all object vertexes producing a general deformation, even if control vertex and object vertex are in opposite parts of the cage. Note though that in this case the influence is small.



(a) Mean Value Coordinates **(b) Harmonic Coordinates**

Fig. 8 Coordinate values for the blue marked vertex, all over the domain. Note that Mean Value Coordinates takes two colors for coordinate representation: yellow for positive values and green for negative ones. In Harmonic coordinates, the domain only complains cage interior, but values are always positive. From [16]

Joshi exposes a local method, like heat diffusion, to decay control vertexes influence as it flows through the interior of the cage. The coordinates values inside the cage have a top bound value that corresponds to control vertexes values, usually 1, and decay to 0 as they are placed farther from the cage vertexes. Hence, all the coordinates will be whithin the range [0..1], but are constrained to be inside the cage, since the boundary breaks off the influence diffusion. This approach ensures positive coordinate values all over the domain: the cage interior, as shown in figure (8 (b)). Joshi's approximation for piecewise linear function $u_{\mathscr{T}}$ is the identity function, the simplest harmonic function. It satisfies Laplace's equation and can propagate the values from boundary to cage interior as we want. By simply applying the laplacian

operator, we get the desired results. Note that if the cage is a triangle the resulting coordinates are the simple barycentric coordinates. Unfortunately, this process encloses the domain to cage interior, allowing no possible influence outside, so that continuity at boundaries will only be preserved by Dirichlet boundary conditions. Therefore, to get a smooth result, the cage needs to wrap completely the object without intersections, or to trick the process for achieve some smoothness at boundaries. An interesting feature of this method is that we can use interior control points that behave like cage pieces. These will propagate the influence to the cage volume in the same way as the boundary cage does. But in accordance with that, there will be a lack of smoothness in deformation at points where the object intersects these interior control points. Joshi presents an example that sorts out this problem by subdivision surface objects. Subdivision control points ensure smoothness over the surface even when the deformation applied to its control points is not smooth.

4.1 Implementation and its consequences

The method of applying a simple laplacian operator inside the cage works theoretically (with infinitesimal equations) but computationally we need to discretize a volume big enough to cover the cage, and to compute a coordinate value for every volume division cell. Joshi implemented some empirical tests to fit the best volume subdivision and then propose dividing it into 2^s cells, being $s = 6$ in the 2D version, and $s = 7$ in the 3D case. To compute the coordinates he proposed the next process: first, to mark every cell as it belongs to cage boundary (BOUNDARY), inside (INTERIOR) or outside (EXTERIOR), and then assign cage vertex values to its corresponding cells. Next, throught an iterative process these values will be propagated through the volume if cells are "INTERIOR" marked, with a 4 cell-connected laplacian operator in two dimensions, and 6 cell-connected one for 3D. Joshi proposes some optimizations for reducing computation time, since convergence of laplacian operators is too slow and requires many iteration-steps before it reaches the end process condition. This condition consist of imposing a threshold that sets a top bound for cell values variation between two laplacian operator successive steps. He proposes to use the threshold $t = 10^{-5}$

One of the main advantages of this approach is the fact that the inside object can be changed whenever is needed, given that the coordinates are computed over the grid cells and not over the object. The coordinates can be reassigned to the new object with no more computation than reading values from the grid. Nevertheless, saving the coordinates every cell is a waste of memory, unless the number of cells is smaller than the number of object vertexes. On the other hand, discretization causes some precision errors that need to be managed to deliver right deformation outcomes. In some sense, the problem may be reduced if the implementation is done by applying an adaptive resolution grid.

5 Positive MVC (PMVC)

The Harmonic Coordinates procedure solves the negativity problem of Mean Value Coordinates, but the fact that it lacks closed form expressions makes it very consuming time, even with optimizations. While MVC and Green Coordinates have a closed form formulation that enables parallel computation, in the case of HC is too hard to apply parallelism. To prevent negative values Lipman proposes applying a local Mean Value Coordinates which is accomplished with visibility testing adopting GPU rendering techniques. The normals integration over the sphere (or circumference) centered at point v is computed if the corresponding cage's simplex for the coordinate w_i is viewed by point v. This process allows us to localize the influence of each cage vertex. The results are very good (as Fig. 9 illustrates), but there are some singularities, specially in concave polygons. Lipman trivializes the problem claiming that such a distortion is not significant enough to be taken into account, as his experiments prove. Yet, the problem arises as a consequence of visibility computation, since the visibility region from point v (similar to a kernel centered at point v) has a really sharp boundary. This fact introduces a little lack of smoothness which as long as the vertex forms a concavity becomes a problem. Thus, as a way to overcome this difficulty, Lipman proposes to split out the cage at the points where this problem is presented.

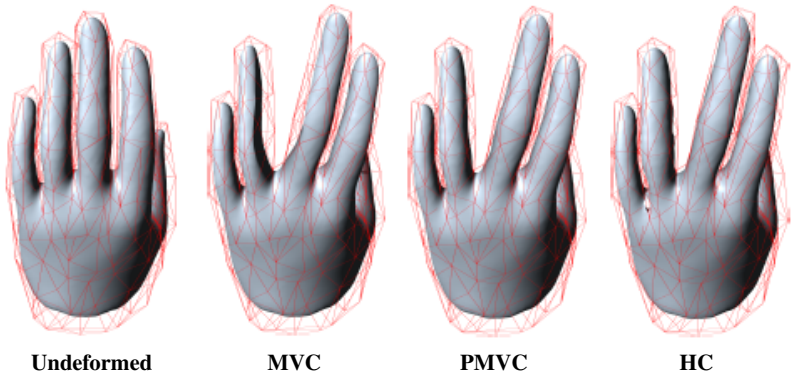


Fig. 9 This example, created by Lipman, shows clearly how negative value coordinates are presented on every kind of coordinates. From [21]

6 Green Coordinates (GC)

The main advantage of deformation methods based on cages are their simplicity, flexibility and fastness. Every object vertex is deformed independently under

these techniques, which are transparent to object surface representation and are ”discretization-error-free” in most of the cases. Unfortunately, the methods discussed previously do not preserve surface details, which is a drawback compared to other deformation techniques (such as those based in surface differentials [30]), especially if deformation is too big. Green Coordinates, developed by Lipman [22], try to apply generalized barycentric coordinates taking into account this fact.

MVC and HC uses only cage vertex positions which, as Lipman noticed, induce an axis independent deformation (i.e., any translation over x axis of a cage vertex does not translate into the object in the y and z axis, resulting an unnatural deformation (Fig. 10))

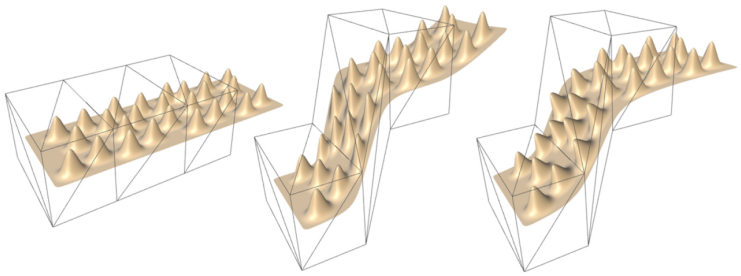


Fig. 10 Detail preservation is exhibited using Green Coordinates (on the right), where the details adhere to the surface deformation and rotate accordingly. In the middle, the MVC result is depicted where the details maintain their original orientation and therefore shear. From [22]

To achieve a natural deformation with shape preserving, Lipman adds cage faces data to the deformation operator.

$$v = F(v;\, C) = \sum_{i \in I_V} \omega_i(v) v_i + \sum_{j \in I_T} \psi_j(v) n(t_j) \tag{11}$$

Being the cage $C = (V,T)$, where V are the cage vertexes and T the cage simplexes (edges or faces). ω_i values are the coordinates based in cage's vertexes and ψ_i are the ones based in the normal of the simplexes, which are the face data added. And $n(t_j)$ is the normal outward unitary vector of the simplex t_j. The deformation over the object is obtained in this way:

$$v \mapsto F(v;\, C') = \sum_{i \in I_V} \omega_i(v) v'_i + \sum_{j \in I_T} \psi_j(v) s_j n(t'_j) \tag{12}$$

where v'_i and t'_j are the modified vertex and faces of C' respectively. The expression introduces a new term $\{s_j\}_{j \in I_T}$ to ensure some properties such as scale invariance. The result is very good as it preserves better and more natural than other methods the object shape and details (Figs. 4, 10, 12) producing a conformal mapping in 2D and quasi-conformal in 3D.

6.1 Coordinates derivation

To obtain closed form expressions Lipman realizes that the harmonic functions also follows the Green's theory, thereby permitting to apply the third Green identity to solve the problem.

$$u(v) = \int_{\partial D} \left(u(\varepsilon) \frac{\partial G(\varepsilon, v)}{\partial n(\varepsilon)} - G(\varepsilon, v) \frac{\partial u(\varepsilon)}{\partial n(\varepsilon)} \right) d\sigma_\varepsilon \tag{13}$$

where ε is a point over the surface and G is the fundamental solution of the Laplace equation, which can be solved by

$$G(\varepsilon, v) = \frac{1}{(2-d)A_d} \|\varepsilon - v\|^{2-d} \text{ dimension } d \geq 3 \tag{14}$$

$$G(\varepsilon, v) = \frac{1}{2\pi} \log \|\varepsilon - v\| \text{ dimension } d = 2 \tag{15}$$

where A_d is the area of a unit sphere in R^d, which in the 3D case is equal to 4π. By deriving these equations, we obtain the closed form expression of the two coordinate sets for each object vertex, $\omega_i(v)$ and $\psi_j(v)$, which are used in the equation (12) to obtain the deformation.

$$\omega_i(v) = \int_{\varepsilon \in N\{v_i\}} \Gamma_k(\varepsilon) \frac{\partial G}{\partial n} d\sigma_\varepsilon, \quad i \in I_V \tag{16}$$

$$\psi_i(v) = -\int_{\varepsilon \in t_j} G(\varepsilon, v) d\sigma_\varepsilon, \quad j \in I_T \tag{17}$$

where $N\{v_i\}$ is the neighborhood 1-connected vertexes of vertex v, which are points used for computing $\Gamma_k(\varepsilon)$, a linear combination to get ε depending on the vertexes of $N\{v_i\}$, in such way that it satisfies $\varepsilon = \sum_{k=1}^{d} \Gamma_k(\varepsilon) v_k$.

The scale factor s_j depends on the growing or decreasing of every cage simplex, and is computed in real-time when the deformation is applied:

$$s_j = \|t'_j\| / \|t_j\| \qquad \text{in 2D case.} \tag{18}$$

$$s_j = \frac{\sqrt{\|u'_1\|^2 \|u_2\|^2 - 2(u'_1 \cdot u'_2)(u_1 \cdot u_2) + \|u_1\|^2 \|u'_2\|^2}}{\sqrt{8}\text{area}(t_j)} \qquad \text{in 3D case.} \tag{19}$$

where vectors u_1 and u_2 define the edges of the triangle t_j, and where u'_1 and u'_2 are the corresponding edges of the modified cage triangle t'_j. This parameter ensures a natural deformation, which can though be modified to distort slightly the uniform scaling (see the pseudocode in the appendix)

Lipman has studied the existing distortion in MVC, HC and GC, with the ratio $\frac{\sigma_{max}(v)}{\sigma_{min}(v)}$, where $\sigma_{max}(v)$ and $\sigma_{min}(v)$ are the maximum and minimum singular value of

the differential F (jacobian matrix) at point v. Any map with a top bounded distortion is named quasi-conformal, but only Green Coordinates present such upper bound (except in some degenerated cases). MVC and HC distortions are proportional to the deformation applied.

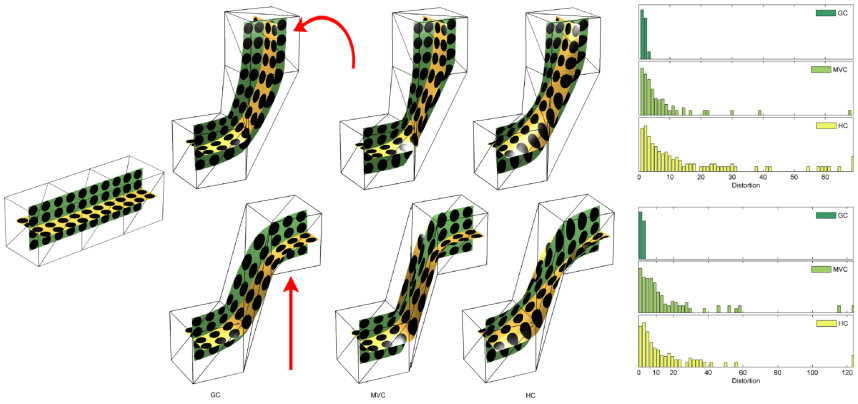


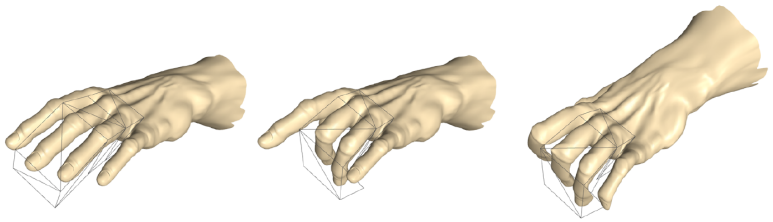
Fig. 11 There are two successive deformations applied to the same object using all methods. The graphics show the distortion error accumulation at every deformation. Only GC presents a top bounded error. From [22]

The coordinates expressed in equations 11, 12, 16, 17 are smooth and well defined all over its limited domain: inside the cage. Besides, $\omega_i(v)$ as a function (eq. 16), present derivate discontinuities over the simplexes at vertexes intersections and, therefore is not smooth at the boundaries. In fact, this function cannot be used as a surface interpolation method. In spite of that, it seems to be a really good system for character articulation. Lipman does not mention anything about the negativity of his coordinates, but attending to the corresponding closed expressions, we expect positive values at the entire domain.

6.2 Outwards cage extension

The lack of smoothness introduced by the mentioned discontinuities at the boundaries entails limitations for the expressed formulation to be inside the cage. That is why Lipman adds some terms to extend the inside coordinates to reach beyond the cage. He considers the outside region as a set of subregions linked to some simplexes. Whenever an object crosses through a simplex, the whole part of the object that is outside the cage will be affected by the transformations of this simplex in the way of affine transformations. Then, a set of simplexes that works with the same similarity transformation could be used as a unique simplex that propagates a unique affine transformation outside the cage. Therefore, we will compute as many affine

transformations as separated parts of the object are placed outwards the cage. (Fig. 12)



(a) Bind pose **(b) Green coordinates** **(c) Mean Value Coordinates**

Fig. 12 GC cage deformation affects only center fingers of the model, leaving deformation-free the rest of the model. MVC affect all the model with a natural extension of the coordinates, but producing unwanted results. From [22]

7 Other developments of cage based methods

There are two early approaches of barycentric coordinates. The first one, presented by Hormann and Sukumar[13], is called Maximum entropy coordinates (MEC). These coordinates are non-negative for arbitrary polytopes, are smooth inside the domain, and can be computed locally at any point v inside the cage like MVC and GC, but they are only defined inside the convex hull of the polytope, and not everywhere in R^d. MEC do not present a significant improvement over the already discussed methods. Weber [34] presents another kind of coordinates named Complex Barycentric Coordinates. This interesting approach reformulates the barycentric coordinates definition with expressions based in complex numbers. Unfortunately, it only works for 2D, and seems to be very difficult to be extended to a higher dimension.

There are some studies that make use of cage based methods and reach further. We have already discussed direct cage based deformation, but enriching a more complex behavior to this deformation will be interesting. Since the cage provides a simplified version of an object for simpler deformation, it enables computing complex constraints over the object with a simpler computation over the cage. Actually, this constraint assignment is further complex, we should apply constraints to the object rather than to the cage. Therefore, in a previous step, we group object constraints on cage vertexes for collapse computations. Generalized barycentric coordinates can do this for us. An interesting contribution in this regard is the one made by Cervero el al.[4], who develop a method for object volume preservation over any kind of generalized barycentric coordinates. Volume preservation is a highly esteemed skill for creating believable deformations in character animation. Another paper that follows

such kind of scheme is Huang et al.[14], in which object deformation is obtained by subspace gradient domain techniques constrained by some conditions. They uses this constraint grouping method for improving the computations which would not otherwise be made in real or at least close-to-real time.

There is still another interesting work that has a close relationship with the essence of cage based methods: biharmonic coordinates [15]. The goal of all the mentioned methods consist of defining a volume from its cage surface, in the case of biharmonic coordinates the cage is the object itself. As long as we have a good knowledge of the volume (the relationship between points within) we will be able to process deformations by modifying some points inside or over the surface while preserving these relations as a constraint. Control points could be vertexes, segments or faces. Besides, they can be inside, on the surface or outside the object. These coordinates, properly created, generate smooth deformations everywhere and can be used with other character articulation tools, such as point deformers, skeletons and cages.

Animation transfer using cages is another approach that also uses generalized barycentric coordinates [5]. Whenever for the cases in which two characters are using exactly the same cage (or different cages but with the same cage topology), every deformation applied to one cage can be translated into the other one easily. A different approach pursuing a similar goal is the one proposed by Ju et al.[19]. They use templates, as cage predefined setups, for applying similar animations to several characters easily.

8 Conclusions

In this paper we have described cage based deformation methods and some of the applications that exploit their utility. Each of these methods has strong and weak points, which are more or less relevant depending on the purpose. In the following section, we discuss the most significant points in a schematic way, as a summary of the concepts previously exposed. In the appendix we collect some pseudocode implementations, extracted from their corresponding papers, that could be useful for further clarifying these concepts.

8.1 discussions

The three methods described have been developed by gradually approximating the issue. Floater [9] first proposed a generalization for barycentric coordinates, by introducing harmonic functions theory, which proves mean value theorem. The main purpose of Mean Value Coordinates was to interpolate values from some points over a surface into its volume (i.e. vertex color). The same formulation is used to interpolate the space position of v based on cage vertex positions. Then, a cage could

Table 1 Cage based methods properties comparative

Property	Mean Value Coords	Harmonic Coords	Green Coords
Cage topology	Triangles (vertex)	No matter (vertex)	Triangles (vertex + normal)
Conformality	No	No	conformal(2D) quasi-conformal(3D)
Interpolation	Boundary + inside	Inside	Inside
Closed formulation	Yes	No	Yes
Control point influence	Global (euclidean distances)	local	Global
Positiveness	No	Yes	Yes
Outside cage extension	natural formulation extension	No	Segmented affine transformation

be created wrapping a model, and deform the model maintaining the same interpolation values (coordinates) before and after the cage transformation. Note that until this point there have been used only point data to compute the interpolation, regardless of the cage topology. This is important since the lack of data will cause problems like negative coordinates and no-local deformations. These drawbacks are specially annoying for character animation. The Harmonic Coordinates approach solves these problems by adding more data such as intra-cage-space point relations, but to satisfy the neighboring relationships at the time of computing the coordinates, is too time and memory consuming. On the other hand, grid discretization process proposed for HC computation is very flexible for character articulation because we can reassign coordinates to different models, composed with a unique mesh or in several parts, without recomputing anything else, provided they use the same cage. Green Coordinates was born as a solution of MVC and HC drawbacks, taking into account also surface details preservation. To achieve it, is added more data into the computation (cage simplexes normal) for describing better the behavior of deformation at surface, and translate it accordingly to the interior, producing a conformal or quasi-conformal mapping. On table 1 we summarize the features that characterize each method.

MVC are the best solution for value interpolation, given that are well defined inside the volume and over the surface, and their simple closed form expressions encourages to use they in the fastest real-time applications, specially if they are computed by GPU techniques. The fact of having negative values and global deformation makes this method less interesting to character articulation.

HC is a good solution for character articulation, specially to add a specific and controlled deformation to a part of an object, due to its flexibility to control influence propagation. Although we must take into account the smooth discontinuities at boundaries to achieve a good overcome. In a rigging process is really valuable that the model could change after the cage creation and coordinates computation, so that HC become a better solution in that cases.

Finally, GC is the best approach for general purpose in character articulation, since surface detail preservation with few control points makes it effective and easy to use. Its formulation is more complex than the other two methods, so that the deformation application will be slower. Besides, its global deformation is a small drawback that must be solved in further research.

We present a table for analyzing the preprocessor computation time. It has been prepared by a non-optimized implementation of the MVC, HC, and GC papers discussed. The timings were measured on a 2.8 Ghz Quad-Core Intel Xeon with 4GB of RAM and expose clearly that MVC is the fastest algorithm, followed by GC with a near to 5 times factor, and far behind, the HC with several magnitude orders over. Note that while MVC and GC increases time computation proportionally with model and cage complexity, HC depends on the grid size, specifically the INSIDE marked cells count.

Table 2 Cage based methods, preprocessor computation time comparative (in seconds)

Model name	model vertexes	model faces	cage vertexes	cage faces	MVC	HC	GC
Column	2202	4400	24	44	0.051	31.905	0.241
Bust	255358	510712	32	60	8.232	229.534	38.348
Horse	19851	39698	90	176	1.614	513.549	8.692

Acknowledgements We are grateful to Pedro García, John Grieco and Guillermo Posadas for helping us to enrich the text with their corrections. This work was partially supported by TIN2010-20590-C02-01.

Appendix

First, we expose Green coordinates pseudocode, extracted from Lipman's paper [22]. The 2D and 3D version for deforming object vertexes $\Lambda \subset C^{in}$. We have changed $\phi_i(v)$ from the original paper by $\omega_i(v)$ to keep coherence all over the document. We note that for exterior or boundary points one should add to these coordinates the $\{\alpha_k\}$ and β as is introduced in subsection 6.2, and explained in depth in section 4 of Lipman's paper. Note that α_k and β also posses a simple closed-form

formula employing the regular barycentric coordinates in triangles (3D) or edges (2D).

2D version of Lipman's Green Coordinates.

==========

Input: cage C = (V,T), set of points $\Lambda = \{\eta\}$

Output: 2D GC $\omega_i(\eta)$, $\psi_j(\eta)$, $i \in I_V, j \in I_T, \eta \in \Lambda$

/* Initialization

set all $\omega_i = 0$ and $\psi_i = 0$

/* Coordinate computation

foreach point $\eta \in \Lambda$ do

  foreach face $j \in I_T$ with vertices v_{j1}, v_{j2} do

    $a := v_{j2} - v_{j1}$; $b := v_{j1} - \eta$

    $Q := a \cdot a$; $S := b \cdot b$; $R := 2a \cdot b$

    $BA := b \cdot \|a\| n(t_j)$; $SRT := \sqrt{4SQ - R^2}$

    $L0 := log(S)$; $L1 := log(S+Q+R)$

    $A0 := \frac{tan^{-1}(R/SRT)}{SRT}$

    $A1 := \frac{tan^{-1}((2Q+R)/SRT)}{SRT}$

    $A10 := A1 - A0$: $L10 := L1 - L0$

    $\psi_j(\eta) := -\|a\|/(4\pi)[(4S - \frac{R^2}{Q})A10 + \frac{R}{2Q}L10 + L1 - 2]$

    $\omega_{j2}(\eta) := \omega_{j2}(\eta) - \frac{BA}{2\pi}[\frac{L10}{2Q} - A10\frac{R}{Q}]$

    $\omega_{j1}(\eta) := \omega_{j1}(\eta) - \frac{BA}{2\pi}[\frac{L10}{2Q} - A10(2 + \frac{R}{Q})]$

  end

end

3D version of Lipman's Green Coordinates.

==========

Input: cage C = (V,T), set of points $\Lambda = \{\eta\}$

Output: 3D GC $\omega_i(\eta)$, $\psi_j(\eta)$, $i \in I_V, j \in I_T, \eta \in \Lambda$

/* Initialization

set all $\omega_i = 0$ and $\psi_i = 0$

/* Coordinate computation

foreach point $\eta \in \Lambda$ do

  foreach face $j \in I_T$ with vertices v_{j1}, v_{j2}, v_{j3} do

    foreach l = 1,2,3 do

      $v_{jl} := v_{jl} - \eta$

    end

    $p := (v_{j1} \cdot n(t_j) n(t_j))$

    foreach l = 1,2,3 do

      $s_l := sign(((v_{jl} - p) \times (v_{jl+1} - p)) n(t_j))$

      $I_l := GCTriInt(p, v_{jl}, v_{jl+1}, 0)$

      $II_l := GCTriInt(0, v_{jl+1}, v_{jl}, 0)$

      $q_l := v_{jl+1} \times v_{jl}$

      $N_l := q_l / \|q_l\|$

    end

    $I := -\left|\sum_{k=1}^{3} s_k I_k\right|$

    $\psi_j(\eta) := -I$

    $w := n(t_j) I + \sum_{k=1}^{3} N_k II_k$

    if $\|w\| > \varepsilon$

      foreach l = 1,2,3 do

        $\omega_{jl}(\eta) := \omega_{jl}(\eta) + \frac{N_{l+1} \cdot w}{N_{l+1} \cdot v_{jl}}$

      end

    fi

  end

end

proc **GCTriInt**$(\mathbf{p}, \mathbf{v_1}, \mathbf{v_2}, \eta)$

  $\alpha := cos^{-1}\left(\frac{(v_2 - v_1)(p - v_1)}{\|v_2 - v_1\| \|p - v_1\|}\right)$

  $\beta := cos^{-1}\left(\frac{(v_1 - p)(v_2 - p)}{\|v_1 - p\| \|v_2 - p\|}\right)$

  $\lambda := \|p - v_1\|^2 sin(\alpha)^2$

  $c := \|p - \eta\|^2$

  foreach $\theta = \pi - \alpha$, $\pi - \alpha - \beta$ do

    $S := sin(\theta) \quad ; \quad C := cos(\theta)$

    $I_\theta := \frac{-sign(S)}{2}\left[2\sqrt{c} tan^{-1}\left(\frac{\sqrt{c}C}{\sqrt{\lambda + S^2 c}}\right) + \sqrt{\lambda}\left(\frac{2\sqrt{\lambda}S^2}{(1-C)^2}\left(1 - \frac{2cC}{c(1+C) + \lambda\sqrt{\lambda^2 + \lambda c S^2}}\right)\right)\right]$

  end

  $return \frac{-1}{4\pi}\left|I_{\pi-\alpha} - I_{\pi-\alpha-\beta} - \sqrt{c}\beta\right|$

end

Harmonic Coordinates implementation is exposed in the corresponding subsection (4.1), there is no pseudocode in the paper. Finally, Mean Value Coordinates pseudocode from Ju's paper [18] is presented. It is written for value interpolation, but with some modifications could be adapted for mesh deformation.

Ju's version of Mean Value Coordinates in 3D

==========

for each vertex p_j with values f_j

$d_j \leftarrow \|p_j - x\|$

if $d_j < \varepsilon$ return f_i

$u_j \leftarrow (p_j - x)/d_j$

totalF $\leftarrow 0$

totalW $\leftarrow 0$

for each triangle with vertices p_1, p_2, p_3 and values f_1, f_2, f_3

$l_i \leftarrow \|u_{i+1} - u_{i-1}\|$ // for $i = 1,2,3$

$\theta \leftarrow 2arcsin[l_i/2]$

$h \leftarrow (\sum \theta_i)/2$

if (π - h $< \varepsilon$)

//x lies on t, use 2D barycentric coordinates

$w_i \leftarrow sin[\theta_i]d_{i-1}d_{i+1}$

return $(\sum w_i f_i)/(\sum w_i)$

$c_i \leftarrow (2sin[h]sin[h-\theta_i])/(sin[\theta_{i+1}]sin[\theta_{i-1}]) - 1$

$s_i \leftarrow sign[det[u_1, u_2, u_3]]\sqrt{1-c_i^2}$

if $\exists i, |s_i| \leq \varepsilon$

// x lies outside t on the same plane, ignore t

continue

$w_i \leftarrow (\theta_i - c_{i+1}\theta_{i-1} - c_{i-1}\theta_{i+1})/(d_i sin[\theta_{i+1}]s_{i-1})$

$totalF += \sum w_i f_i$

$totalW += \sum w_i$

$f_x \leftarrow totalF/totalW$

References

1. Barr, A.H.: Global and local deformations of solid primitives. SIGGRAPH Comput. Graph. **18**, 21–30 (1984)
2. Botsch, M., Kobbelt, L.: An intuitive framework for real-time freeform modeling. In: ACM SIGGRAPH 2004 Papers, SIGGRAPH '04, pp. 630–634. ACM, New York, NY, USA (2004)

3. Botsch, M., Kobbelt, L.: Real-time shape editing using radial basis functions. Computer Graphics Forum **24**(3), 611–621 (2005)
4. Cerveró, M.À., Vinacua, Á., Brunet, P.: Volume-preserving deformation using generalized barycentric coordinates pp. 57–66 (2010)
5. Chen, L., Huang, J., Sun, H., Bao, H.: Cage-based deformation transfer. Computers Graphics **34**(2), 107 – 118 (2010)
6. Floater, M., Hormann, K., Kós, G.: A general construction of barycentric coordinates over convex polygons. Advances in Computational Mathematics **24**, 311–331 (2006)
7. Floater, M.S.: Parametrization and smooth approximation of surface triangulations. Computer Aided Geometric Design **14**(3), 231 – 250 (1997)
8. Floater, M.S.: Mean value coordinates. Computer Aided Geometric Design **20**(1), 19 – 27 (2003)
9. Floater, M.S., Kós, G., Reimers, M.: Mean value coordinates in 3d. Computer Aided Geometric Design **22**(7), 623 – 631 (2005)
10. Gerald, Farin: Surfaces over dirichlet tessellations. Computer Aided Geometric Design **7**(1-4), 281 – 292 (1990)
11. Hiyoshi, H., Sugihara, K.: Voronoi-based interpolation with higher continuity. In: Proceedings of the sixteenth annual symposium on Computational geometry, SCG '00, pp. 242–250. ACM, New York, NY, USA (2000)
12. Hormann, K., Floater, M.S.: Mean value coordinates for arbitrary planar polygons. ACM Transactions on Graphics **25**(4), 1424–1441 (2006)
13. Hormann, K., Sukumar, N.: Maximum entropy coordinates for arbitrary polytopes. Computer Graphics Forum **27**(5), 1513–1520 (2008). DOI 10.1111/j.1467-8659.2008.01292.x. URL http://dx.doi.org/10.1111/j.1467-8659.2008.01292.x
14. Huang, J., Shi, X., Liu, X., Zhou, K., Wei, L.Y., Teng, S.H., Bao, H., Guo, B., Shum, H.Y.: Subspace gradient domain mesh deformation. In: ACM SIGGRAPH 2006 Papers, SIGGRAPH '06, pp. 1126–1134. ACM, New York, NY, USA (2006)
15. Jacobson, A., Baran, I., Popović, J., Sorkine, O.: Bounded biharmonic weights for real-time deformation. ACM Trans. Graph. **30**, 78:1–78:8 (2011)
16. Joshi, P., Meyer, M., DeRose, T., Green, B., Sanocki, T.: Harmonic coordinates for character articulation. ACM Trans. Graph. **26** (2007)
17. Joshi, P., Tien, W.C., Desbrun, M., Pighin, F.: Learning controls for blend shape based realistic facial animation. In: ACM SIGGRAPH 2005 Courses, SIGGRAPH '05. ACM, New York, NY, USA (2005)
18. Ju, T., Schaefer, S., Warren, J.: Mean value coordinates for closed triangular meshes. In: ACM SIGGRAPH 2005 Papers, SIGGRAPH '05, pp. 561–566. ACM, New York, NY, USA (2005)
19. Ju, T., Zhou, Q.Y., van de Panne, M., Cohen-Or, D., Neumann, U.: Reusable skinning templates using cage-based deformations. ACM Trans. Graph. **27**, 122:1–122:10 (2008)
20. Lewis, J.P., Cordner, M., Fong, N.: Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation. In: Proceedings of the 27th annual conference on Computer graphics and interactive techniques, SIGGRAPH '00, pp. 165–172. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA (2000)
21. Lipman, Y., Kopf, J., Cohen-Or, D., Levin, D.: Gpu-assisted positive mean value coordinates for mesh deformations. In: Proceedings of the fifth Eurographics symposium on Geometry processing, pp. 117–123. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland (2007)
22. Lipman, Y., Levin, D., Cohen-Or, D.: Green coordinates. ACM Trans. Graph. **27**, 78:1–78:10 (2008)
23. Malsch, E.A., Dasgupta, G.: Interpolations for temperature distributions: a method for all non-concave polygons. International Journal of Solids and Structures **41**(8), 2165 – 2188 (2004). DOI 10.1016/j.ijsolstr.2003.11.037
24. Meyer, M., Lee, H., Barr, A., Desbrun, M.: Generalized barycentric coordinates on irregular polygons. Computer **7**(1), 0–4 (2005)
25. Möbius, A.F.: Der barycentrische Calcül (1827)

26. Peng, Q., Jin, X., Feng, J.: Arc-length-based axial deformation and length preserved animation. Computer Animation **0**, 86 (1997)
27. Sederberg, T.W., Parry, S.R.: Free-form deformation of solid geometric models. SIGGRAPH Comput. Graph. **20**, 151–160 (1986)
28. Sibson, R.: A brief description of natural neighbour interpolation. Interpreting Multivariate Data pp. pages 21–36 (1981)
29. Singh, K., Eugene, F.: Wires: a geometric deformation technique. In: Proceedings of the 25th annual conference on Computer graphics and interactive techniques, SIGGRAPH '98, pp. 405–414. ACM, New York, NY, USA (1998)
30. Sorkine, O.: Differential representations for mesh processing. Computer Graphics Forum **25**(4), 789–807 (2006)
31. Sukumar, N., Malsch, E.: Recent advances in the construction of polygonal finite element interpolants. Archives of Computational Methods in Engineering **13**, 129–163 (2006)
32. Terzopoulos, D., Platt, J., Barr, A., Fleischer, K.: Elastically deformable models. In: Proceedings of the 14th annual conference on Computer graphics and interactive techniques, SIGGRAPH '87, pp. 205–214. ACM, New York, NY, USA (1987)
33. Wachspress, E.: A rational finite element basis. Academic Press, New York (1975)
34. Weber, O., Ben-Chen, M., Gotsman, C.: Complex barycentric coordinates with applications to planar shape deformation. Computer Graphics Forum **28**(2), 587–597 (2009)
35. Xian, C., Lin, H., Gao, S.: Automatic cage generation by improved obbs for mesh deformation. The Visual Computer pp. 1–13 (2011)