EUROGRAPHICS 2009 / P. Dutré and M. Stamminger
(Guest Editors)

Volume 28 (2009), Number 2

Complex Barycentric Coordinates with Applications to Planar Shape Deformation

Ofir Weber, Mirela Ben-Chen and Craig Gotsman

Technion – Israel Institute of Technology

Abstract

Barycentric coordinates are heavily used in computer graphics applications to generalize a set of given data values. Traditionally, the coordinates are required to satisfy a number of key properties, the first being that they are real and positive. In this paper we relax this requirement, allowing the barycentric coordinates to be complex numbers. This allows us to generate new families of barycentric coordinates, which have some powerful advantages over traditional ones. Applying complex barycentric coordinates to data which is itself complex-valued allows to manipulate functions from the complex plane to itself, which may be interpreted as planar mappings. These mappings are useful in shape and image deformation applications. We use Cauchy's theorem from complex analysis to construct complex barycentric coordinates on (not necessarily convex) polygons, which are shown to be equivalent to planar Green coordinates. These generate conformal mappings from a given source region to a given target region, such that the image of the source region is close to the target region. We then show how to improve the Green coordinates in two ways. The first provides a much better fit to the polygonal target region, and the second allows to generate deformations based on positional constraints, which provide a more intuitive user interface than the conventional cage-based approach. These define two new types of complex barycentric coordinates, which are shown to be very effective in interactive deformation and animation scenarios.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism

1. Introduction

Barycentric coordinates are a very useful mathematical tool for computer graphics applications. Since they allow to infer continuous data over a domain from discrete or continuous values on the boundary of the domain, barycentric coordinates are used in a wide range of applications - from shading, interpolation [JSW05], and parameterization [DMA02, SAPH04] to, more recently, space deformations [JMD*07, LKCOL07, LLCO08].

Traditionally, barycentric coordinates in R^n are defined as the real coefficients of an affine combination of vectors in R^n. As such, they operate identically on each coordinate. When working in the plane, barycentric coordinates in R^2 can also be considered as an affine combination of *complex* numbers with real coefficients, thus we propose to consider also the case where the coefficients are themselves allowed to be complex. This new point of view has a few advantages: First, it allows the definition of *complex barycentric coordinates*, permitting a *different* linear operation for each of the two coordinates, through which new effects can be achieved. Second, it unleashes the rich theory of complex analysis, simplifying the underlying theory considerably.

Published by Blackwell Publishing, 9600 Garsington Road, Oxford OX4 2DQ, UK and 350 Main Street, Malden, MA 02148, USA.

Complex barycentric coordinates are especially useful for 2D shape and image deformation. In a typical application scenario, the user defines a *source* contour, usually a polygon, and deforms it to a *target* contour by moving its vertices. This indicates to the application that the region within the source contour should be deformed in some natural way to the region within the target contour, such that the per-edge correspondence is respected. As this operation is fundamental for 2D animation applications, many algorithms have been proposed for it, a large number of them based on barycentric coordinates [JMD*07, LKCOL07]. Using barycentric coordinates means that the coordinates of a point x in the source polygon are expressed as an affine combination of the coordinates of the source polygon vertices, and x is then mapped to $f(x)$ – the same affine combination of the coordinates of the vertices of the target polygon. Recently, observing that traditional (real) barycentric coordinates *by definition* reproduce affine transformations, hence not ideal for shape-preserving deformations because they may introduce shears, Lipman et al. [LLC08] generalized the concept of barycentric coordinates to be the standard linear combination of the coordinates of the vertices of the polygon *plus a linear combination of the normals to the edges of the polygon*. They showed how to use these coordinates in order to generate a *conformal* mapping of the interior of the source polygon. Conformal mappings preserve infinitesimal angles, hence preserve details better than arbitrary mappings

We show that Lipman's Green coordinates are a special case of complex barycentric coordinates, and provide a very simple analytic formula for them. In addition, we propose new complex barycentric coordinates for 2D shape deformation, which are shown to improve the Green coordinates, as the deformation better fits the user's specifications, without losing any of the properties of the Green coordinates. Finally, we give simple analytic formulae for the *derivatives* of the Green coordinates. This allows us to define constraints on the derivative of the deformation, in addition to positional constraints, This allows to replace the source-target polygon contour (“cage”) user interface with a more intuitive and user-friendly point-to-point user interface.

1.1. Previous work

The simplest real barycentric coordinates in the plane are defined on a basis of three points forming a triangle. These unique coordinates are just ratios between various areas. Real barycentric coordinates satisfy a number of important properties, such as non-negativity, constant precision, linear precision, interpolation and smoothness. The main challenge in developing new recipes for barycentric coordinates is to find coordinates which can be applied to a wider range of bases – such as convex polygons, general polygons, continuous contours, and complexes in higher dimensions – while maintaining the attractive properties of the simple barycentric coordinates on a triangle. Many generalizations have been developed in recent years, and we will mention only those most relevant to our work.

One of the most commonly-used barycentric coordinates are the mean-value coordinates, first introduced by Floater [Flo03]. These coordinates have the nice property that they are positive on convex polygons, and can be generalized to R^3 [JSW05, FKR05, LBS06] and to the exterior of polygons [HF06]. In addition, they can also be generalized to be positive on non-convex polygons [LKCOL07]. Mean-value coordinates are derived from the mean-value theorem for harmonic functions, applied to a piecewise linear contour – a polygon. As we will show in the next sections, Green coordinates [LLCO08] can be derived by applying Cauchy's integral formula – which is the complex equivalent of the mean-value theorem for holomorphic functions – to a polygon. In this sense, the Green coordinates are conceptually a generalization of mean-value coordinates to complex functions.

Floater et al. [FHK06] have showed that mean-value coordinates are members of a large family of barycentric coordinates, known as "three-point coordinates". We show how to derive in a similar way a family of complex three-point coordinates, and show that the Green coordinates are a member of this family.

In recent works [Bel06, WSHD07], barycentric coordinates were developed for continuous planar contours. The main challenge here is to find a barycentric coordinate function – or *kernel* – such that the resulting transform will have linear precision. We show that the Green coordinates on a polygon originate in a simple kernel on a continuous contour, which easily achieves linear precision and conformality.

One of the applications of planar barycentric coordinates is planar shape deformation, for example, for image warping. This is also a highly active research area, and we will only mention a few recent works, which are closest in spirit to ours. There are two major approaches to planar deformation – the first requires discretizing the shape into finite elements, so that the deformation is performed by solving an optimization problem on this discretization, see [BS08] for a survey of recent linear methods. The biggest disadvantage of this approach is that the computation time is dominated by the complexity of the discretization, and not by the intrinsic complexity of the shape itself. The second approach is more analytic and does not require a discretization of the domain. An example of this approach is Moving Least Squares (MLS) [SMW06] which allows for affine, similar and rigid deformations. However, the MLS approach deforms the entire plane, and does not take into account the underlying shape of the deformed object.

In order to allow shape-aware deformations, Ju *et al.* [JSW05] suggested to define a "cage" around the shape and control the deformation by manipulating the cage. Most of the cage-based deformations use barycentric coordinates, for example [LKCOL07, JMD*07]. A notable set of barycentric coordinates are the so-called *harmonic* coordinates [JMD*07]., named that way since they are the unique solution to the Laplace equation with Dirichlet boundary conditions on the cage. In contrast to other barycentric coordinates on polygons, no analytic expression is available for harmonic coordinates. They are quite difficult to compute, and are typically approximated using a discretization of the interior of the cage. Despite the usefulness of barycentric coordinates in data interpolation applications, Lipman et al.

[LLCO08] have observed that the affine invariance property of barycentric coordinates is in fact *harmful* for shape deformation, as it introduces undesirable shears, see for example Fig. 1. This is probably inevitable if the source region is forced to deform to fit precisely to the target region. Relaxing this constraint, Lipman et al. propose to work with *two* sets of barycentric coordinates, one for the vertices of the cage, and the second for the normals of its edges. The resulting deformation is shown to be conformal, although not interpolating. We show how to derive these coordinates as simple *complex* barycentric coordinates, and, in addition, show how to improve them.

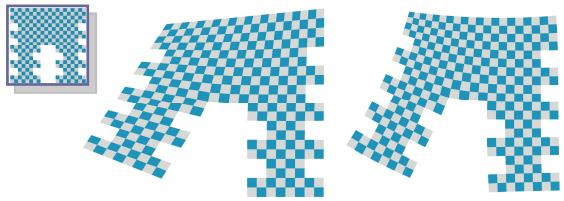


Figure 1: *Affine invariance of harmonic coordinates causes shearing, whereas Green coordinates generate a conformal map, better preserving the details.*

In some scenarios, it is easier to control a deformation by using control points instead of a cage. The smaller number of control points leaves extra degrees of freedom, which may be used to impose constraints on the *derivative* of the deformation. We develop the derivatives of the Green coordinates, and show how to exploit them in a more user-friendly deformation setting.

The rest of the paper is organized as follows. In the next section we introduce complex barycentric coordinates, and show examples of such coordinates – the discrete and continuous Cauchy-Green coordinates, and their generalization to a family of complex three-point coordinates. In Section 3, we modify the Cauchy-Green coordinates to be *variational* in the sense that the deformation attempts to minimize a given objective function, and introduce the Szegö and Point-to-Point Cauchy coordinates. Section 4 presents experimental results which show how these new coordinates out-perform current state-of-the-art methods for planar deformation. We conclude with a discussion of open issues in Section 5.

2. Complex barycentric coordinates

2.1 Definitions

Let $S = \{v_1, v_2, \dots, v_n\} \subset R^2$ be the vertices of a simply connected planar polygon, oriented in the counter clockwise direction, $v_j = (x_j, y_j)$. Let $z_j = x_j + iy_j$ be the representation of the vertices as complex numbers, with $i = \sqrt{-1}$, $z_j \in \mathbb{C}$. Denote by Ω the interior of S. Given a point $v = (x,y) \in \Omega$, define $z = x + iy$ and consider the following *complex* linear combination:

$$\sum_{j=1}^{n} k_j(z) z_j$$

where $k_j(z):\Omega \to \mathbb{C}$.

We say that the functions $k_j(z)$ are *complex barycentric coordinates* with respect to S if the following two properties hold for all $z \in \Omega$:

Constant precision:

$$\sum_{j=1}^{n} k_j(z) = 1 \tag{1}$$

Note that this implies that the real part of the coordinates sums to 1, and the imaginary part sums to 0.

Linear precision:

$$\sum_{j=1}^{n} k_j(z) z_j = z \tag{2}$$

Constant precision is sometimes called "reproduction of unity", and linear precision called "reproduction of the identity". Given complex barycentric coordinates $k_j(z)$ for S, we may consider the complex function $g_{S,F}(z)$ which results from applying the complex barycentric coordinates to the vertices of a target polygon $F = \{f_1, f_2, \dots, f_n\} \subset \mathbb{C}$, as in Fig. 2:

$$g_{S,F}(z) = \sum_{j=1}^{n} k_j(z) f_j \tag{3}$$

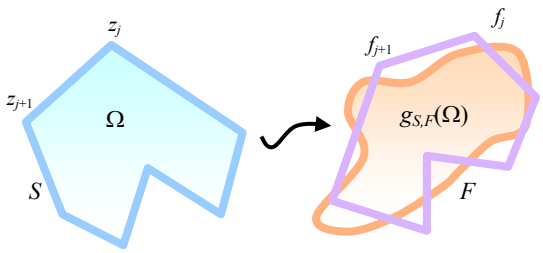


Figure 2: *Planar mapping from Ω - the interior of the polygon S, to $g_{S,F}(\Omega)$ using complex barycentric coordinates $k_j(\Omega)$.*

Note that F should also be simply connected and oriented counter-clockwise. The complex function g can be viewed as a planar mapping from Ω - the interior of S - to its image $g(\Omega)$. Note that $g(\Omega)$ will usually *not* be the interior of the polygon F. However, since the coordinates reproduce unity and the identity, they will reproduce any linear function of z. Recalling that a linear function of a single complex variable is equivalent to a similarity 2D transformation in the plane, we have:

Theorem 1: Complex barycentric coordinates reproduce similarity transformations.

This means that if $f_j = f(z_j)$ for some similarity transformation f, then $g_{S,F} = f$.

Note that although all complex barycentric coordinates reproduce similarity transformations, not all reproduce affine transformations. It is quite straightforward to see that:

Theorem 2 Complex barycentric coordinates $k_j(z)$ reproduce affine transformations if and only if the complex conjugates of the coordinates $k_j(z)$ also have linear precision:

$$\sum_{j=1}^{n} \overline{k}_j(z) z_j = z$$

In particular, all real barycentric coordinates reproduce affine transformations. Somewhat counter-intuitively, the

fact that most complex barycentric coordinates do not reproduce affine transformations is an advantage, as, in many applications, non-uniform scale is undesirable.

On top of not reproducing affine transformations, we depart from another standard property of real barycentric coordinates: we do not require the barycentric coordinates to be interpolating (sometimes called the *Lagrange* property of the coordinates). We expect the image $g(\Omega)$ to be close in some sense to the target polygon F, but do not impose the strict interpolation requirement: $g(z_j)=f_j$. In many case this allows for more natural mappings.

Complex barycentric coordinates can be easily generalized to continuous contours in the following way. Let Ω be a simply connected open planar region with a smooth boundary S. Given $z \in \Omega$ and $w \in S$, consider the complex function $k(w,z):S\times\Omega\to\mathbb{C}$. Analogously to the discrete case, we say that $k(w,z)$ is a barycentric coordinate function if it satisfies the following properties for all $z\in\Omega$:

Constant precision:

$$\oint_S k(w,z)dw = 1 \tag{4}$$

Linear precision:

$$\oint_S k(w,z)w dw = z \tag{5}$$

The function $k(w,z)$ is sometimes called a *kernel* function. The main difference between this and the continuous definition of real barycentric coordinates [Bel06, WSHD07] is that here the integral over S is a *complex* integral, where $dw = T(w)ds$. $T(w)$ is the unit-length tangent vector to S at w, and ds is the usual arc-length differential element. See Fig. 3.

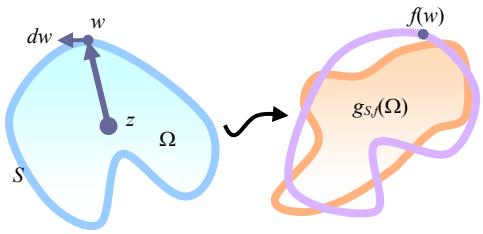


Figure 3: *Continuous planar mapping from* Ω *- the interior of the closed curve* S *- to* $g_{S,f}(\Omega)$ *using the complex barycentric kernel* $k(S\times\Omega)$.

Analogously to (3), given a continuous complex function $f(S)$: $S \to \mathbb{C}$, we can define a planar mapping $g_{S,f}(\Omega)$ as follows:

$$g_{S,f}(z) = \oint_S k(w,z)f(w)dw$$

As in the case of real barycentric coordinates, the main challenge is to find kernels $k(w,z)$, or, in the discrete case, coordinate functions $k_j(z)$, which satisfy the required properties. Next we will show how a simple complex kernel can be used both in the continuous and discrete settings to obtain useful barycentric coordinates, and how to generalize this kernel to a family of discrete complex three-point-coordinates.

2.2 Continuous Cauchy coordinates

Consider the complex function:

$$C(w,z) = \frac{1}{2\pi i}\frac{1}{w-z}$$

C is well known from complex analysis, where it is called the *Cauchy kernel* [Bel92]. C satisfies the two properties (4) and (5), namely:

$$\frac{1}{2\pi i}\oint_S \frac{1}{w-z}dw = 1; \quad \frac{1}{2\pi i}\oint_S \frac{w}{w-z}dw = z;\ z\in\Omega$$

because these two identities are special cases ($h(w)=1$ and $h(w)=w$) of Cauchy's integral formula [Ahl79], which asserts that the values of a function on the boundary of a simply-connected region determine its value at every point inside the region:

$$\frac{1}{2\pi i}\oint_S \frac{h(w)}{w-z}dw = h(z) \tag{6}$$

Cauchy's integral formula holds for the class of complex functions known as *holomorphic functions*. Such functions are the linear subspace of "well behaved" complex functions, and also have a geometric interpretation – a holomorphic function whose first derivative does not vanish is a conformal mapping. See Ahlfors [Ahl79] for a detailed introduction to holomorphic functions. So, in fact, the Cauchy kernel reproduces *all holomorphic functions*. We call the resulting coordinates *Cauchy coordinates.*

Applying the Cauchy coordinates to a target contour $f(S)$ defines the following mapping:

$$g_{S,f}(z) = \frac{1}{2\pi i}\oint_S \frac{f(w)}{w-z}dw \tag{7}$$

Note, that equations (6) and (7) are very different, as h in (6) is a holomorphic function defined on S and on Ω , and f in (7) is a function defined *only* on S.

The mapping $g(\Omega)$ in (7) is sometimes called the *Cauchy transform* of f [Bel92]. It has various interesting properties, one of which being that if f is continuous on S, then g is always holomorphic on Ω [Bel92, Theorem 3.1]. Hence, if we apply these coordinates in the context of planar shape deformation, the deformation is guaranteed to be conformal (if the derivatives do not vanish). In addition, since holomorphic functions are infinitely differentiable, the mapping will be smooth.

2.3 Discrete Cauchy-Green coordinates

In a practical shape deformation scenario, the contour S is usually a polygon (sometimes called "cage") which the user deforms to a new polygon F, as in Fig. 2. Let us now consider what (7) reduces to when S is a polygon $S = \{z_1, z_2, \ldots, z_n\}$. Although S does not have a tangent vector at z_i ,(6) is still valid, by applying it to each edge $e_j = (z_{j-1}, z_j)$ separately:

$$g_{S,f}(z) = \frac{1}{2\pi i}\sum_{j=1}^{n}\int_{e_j}\frac{f(w)}{w-z}dw$$

Since F is also a polygon, f maps each edge of S linearly to an edge of F. Hence, for $w \in (z_{j-1}, z_j)$:

$$f(w) = f_{j-1} + \frac{(f_j - f_{j-1})(w - z_{j-1})}{(z_j - z_{j-1})}$$

Computing the integral on a single edge e_j:

$$\int_{e_j} \frac{f(w)}{w-z}\,dw = \log\frac{B_j(z)}{B_{j-1}(z)}\left(f_{j-1}\frac{B_j(z)}{A_j} - f_j\frac{B_{j-1}(z)}{A_j}\right) + f_j - f_{j-1}$$

where $B_j(z) = z_j - z$ and $A_j = z_j - z_{j-1}$, as in Fig 4. Summing over all edges, and rearranging the terms yields:

$$g_{S,f}(z) = \sum_{j=1}^{n} C_j(z) f_j$$

$$C_j(z) = \frac{1}{2\pi i}\left(\frac{B_{j+1}(z)}{A_{j+1}}\log\left(\frac{B_{j+1}(z)}{B_j(z)}\right) - \frac{B_{j-1}(z)}{A_j}\log\left(\frac{B_j(z)}{B_{j-1}(z)}\right)\right) \quad (8)$$

Cauchy-Green complex barycentric coordinates

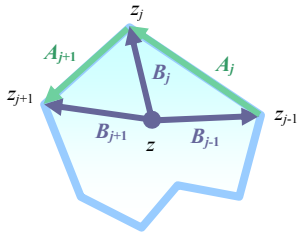


Figure 4: *Notations for the Cauchy-Green coordinates.*

We call $C_j(z)$ *discrete Cauchy coordinates* and g the *discrete Cauchy transform* of f. Note that we have defined $C_j(z)$ only on the interior of the polygon, since on the boundary the expression may be singular. See Fig. 5 for an example of a mapping of a polygon generated by discrete Cauchy coordinates. Note that the image of the polygon is not a polygon. The mapping is just a linear combination of the n holomorphic coordinate functions $C_j(z)$, one of which is visualized in the figure.

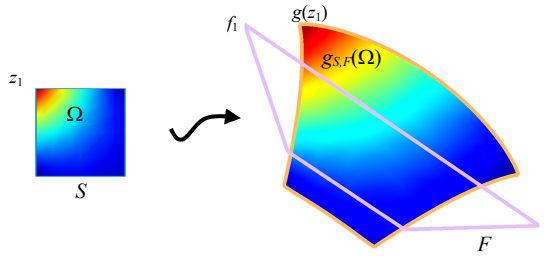


Figure 5: *Example of a mapping of source polygon S, guided by target polygon F, using the discrete Cauchy coordinates. The results is the region $g_{S,F}$. The real part of $C_1(z)$ is visualized using color-coding both on the source polygon and on $g(\Omega)$.*

The discrete Cauchy transform has a number of desirable properties. First, like the continuous Cauchy transform, the resulting function g is holomorphic, and infinitely differentiable. Hence, the mapping from Ω to $g(\Omega)$ is conformal. In addition, as any complex barycentric coordinate function, it reproduces similarity transformations (see Theorem 1).

As it turns out, the discrete Cauchy coordinates have already been discovered, albeit in a different guise.

Theorem 3: Lipman's 2D Green coordinates [LLCO08] are identical to discrete Cauchy coordinates.

Proof: The 2D Green coordinates are defined in the following way:

$$g(\eta) = \sum_{k\in V}\phi_k(\eta)v_k + \sum_{j\in V}\psi_j(\eta)n(t_j)$$

where v_k and t_j are the vertices and edges of the cage, respectively, and $n(t_j)$ are the un-normalized normals to the edges. The coordinate functions ϕ_k and ψ_j are the closed form integrals present in Green's third identity, and have somewhat complicated expressions. Denoting by z_k the complex representation of the cage vertices v_k, we get:

$$g(\eta) = \sum_{k\in V}\phi_k(\eta)z_k + \sum_{j\in V}\psi_j(\eta)i(z_{j+1} - z_j)$$

since the un-normalized normal to an edge is just the edge rotated by $\pi/2$, which is equivalent to multiplication by i in the complex plane. Rearranging terms we have:

$$g(\eta) = \sum_{j\in V}\left(\phi_j(\eta) + i\left(\psi_{j-1}(\eta) - \psi_j(\eta)\right)\right)z_j$$

At this point, we can plug in the formulas for ϕ_j and ψ_j given in [LLCO08] and simplify the resulting expression. The derivation is extremely technical, but following it through results in (8) – the discrete Cauchy coordinates. ♦

It should actually not come as a surprise that the discrete Cauchy coordinates and the Green coordinates are equivalent. One is derived from Cauchy's integral formula, and the second from Green's third identity. Since these are known to be equivalent (in the sense that one can be derived from the other [Ahl79]), the connection between the two is inevitable. Thus, in the sequel, we will refer to these coordinates as the *Cauchy-Green* coordinates.

In the next section, we will consider a general family of complex barycentric coordinates analogous to the "three-point coordinates" family defined in [FHK06].

2.4 Complex three-point coordinates

Floater *et al.* [FHK06] described a family of barycentric coordinates such that the coordinate k_j depends only on the points v_{j-1}, v_j, v_{j+1}, and showed that known planar barycentric coordinates, such as the mean-value, Wachpress and discrete harmonic coordinates are all members of this family. It turns out that a similar classification can be made for complex barycentric coordinates, and that the Cauchy-Green coordinates are a member of this family.

Theorem 4: Let $m_j(z):\Omega\to\mathbb{C}$, $j = 1,..,n$ be complex functions, and let $B_j(z) = z_j - z$ and $A_j = z_j - z_{j-1}$, as in Fig. 4. Then the functions:

$$k_j(z) = m_j(z)\frac{B_{j+1}(z)}{A_{j+1}} - m_{j-1}(z)\frac{B_{j-1}(z)}{A_j}$$

satisfy:

$$\sum_{j=1}^{n} k_j(z)(z_j - z) = 0$$

Proof: The key to the proof, and one of the differences between complex and real coordinates, is the fact that an arbitrary complex number z can be expressed uniquely using a complex affine combination of only *two* other

points, while an arbitrary 2D vector requires an affine combination of *three* points. Specifically, given a complex number z, and two other complex numbers z_j and z_{j+1}, there exist complex numbers $\alpha_j(z)$ and $\beta_j(z)$ such that:

$$z = z_j\beta_j(z) + z_{j+1}\alpha_j(z)$$
$$\alpha_j(z) + \beta_j(z) = 1$$

Finding $\alpha_j(z)$ and $\beta_j(z)$ is a matter of solving these two linear complex equations in two variables.

It is easy to check that the solutions are:

$$\alpha_j(z) = \frac{z - z_j}{z_{j+1} - z_j}, \quad \beta_j(z) = \frac{z_{j+1} - z}{z_{j+1} - z_j}$$

Following Floater *et al.*, [FHK06], we can write:

$$D_j(z) = \beta_j(z)(z_j - z) + \alpha_j(z)(z_{j+1} - z) \equiv 0$$

Now, any linear combination of $D_j(z)$ vanishes, hence:

$$\sum_{j=1}^{n} m_j(z) D_j(z) = 0$$

Plugging back $\alpha_j(z)$ and $\beta_j(z)$, and rearranging the terms:

$$\sum_{j=1}^{n}\left(m_j(z)\frac{B_{j+1}(z)}{A_{j+1}} - m_{j-1}(z)\frac{B_{j-1}(z)}{A_j}\right)(z_j - z) = 0 \qquad (9)$$

which concludes the proof.

Given a set of functions $k_j(z)$ as in Theorem 4, whose sum is non-zero for all z, it is straightforward to see that the functions $w_j(z) = k_j(z)/\sum k_j(z)$ have the constant precision and linear precision properties, and hence are by definition complex barycentric coordinates.

In fact, using a proof which is almost identical to the one in [FHK06], we can show that the converse of Theorem 4 is also true:

Theorem 5: Any set of complex functions $k_j(z)$ which satisfy

$$\sum_{j=1}^{n} k_j(z)(z_j - z) = 0$$

can be expressed in the form (9).

The family of complex three point coordinates is generated by restricting $m_j(z)$ to be a complex function of only $B_j(z)$ and $B_{j+1}(z)$.

Comparing the expression for the discrete Cauchy-Green coordinates from (8) to the expression in (9), we can see that the discrete Cauchy-Green coordinates are members of the complex three-point coordinate family, with:

$$m_j(z) = \frac{1}{2\pi i}\log\frac{B_{j+1}(z)}{B_j(z)} = \frac{1}{2\pi i}\int_{z_j}^{z_{j+1}} \frac{1}{w - z}\,dw$$

So far, we have introduced complex barycentric coordinates, and showed how to easily derive the planar Cauchy-Green coordinates in this context. In the next section, we will demonstrate that the complex point of view not only gives insight into existing barycentric coordinates, but also allows us to generate new coordinates which perform better than state-of-the-art algorithms for planar shape deformation.

3. Cauchy-type coordinates and shape deformation

In planar shape deformation applications, the user deforms a shape, which is usually a planar mesh or an image. In cage-based applications, the user draws a "cage" around the shape of interest, and modifies the shape by deforming the cage. Two main requirements must be met for an algorithm to be practical – the mapping should preserve as much as possible the details of the shape or image, and the deformation should be fast enough to be run interactively.

Since conformal maps preserve angles and the shape of small details, they are good candidates for detail-preserving deformations. Barycentric coordinates are extremely fast to compute – the complexity of the computation for a single point in the deformed domain depends only on the complexity of the cage, which is usually significantly smaller than the complexity of the deformed shape. Thus complex barycentric coordinates which produce conformal maps could provide the best of both worlds and be a very useful tool for planar shape deformation.

Unfortunately, in general, given two arbitrary polygons with corresponding vertices, there does not exist a conformal mapping which maps the corresponding edges linearly to each other. Thus, to achieve conformality we must relax the interpolation requirement. Luckily, the space of conformal mappings from a given source polygon to the region close to another given target polygon is quite large, so we can add an additional requirement – find the complex barycentric coordinate functions which give a conformal mapping *and* minimize a functional of our choice.

As described in Section 2.2, the Cauchy transform takes as input a continuous function f on a contour S, and outputs g, a function holomorphic on the interior of S, see Fig. 3. Thus the Cauchy transform can also be interpreted as a *projection* from the linear subspace of continuous functions on S, to the linear subspace of holomorphic functions on Ω. It is a projection because of its holomorphic function reproduction property – if f is holomorphic on $S \cup \Omega$, then $g=f$ on Ω. An interesting question is whether the Cauchy transform is an *orthogonal projection*, meaning that the resulting holomorphic function g is the *closest* holomorphic function, in some metric, to the given function f. For the metric derived from the following inner product of two complex functions of a complex variable:

$$< f, g >= \oint_S (f \cdot \bar{g})\,ds$$

namely, the metric that measures goodness of fit of the image of the source contour $g(S)$ to the target contour $F = f(S)$, the answer to this question is negative – the Cauchy projection is an *oblique projection*, and given an arbitrary function f there exists a holomorphic function whose boundary values are closer to f than $g=Cf$. This, in general, is not good news – it means the mapping generated by the Cauchy transform is indeed conformal, but it is not as close as it could be to the user's specification. But there are also good news: the orthogonal projection of f, also known as the *Szegö projection* [Bel92], onto the space of holomorphic functions, is relatively well-known, and computable.

Since we have the closed form (8) for the discrete Cauchy-Green coordinates, we would like to stay within

the n-dimensional subspace of holomorphic functions spanned by these coordinate functions, which we call the *Cauchy-Green subspace*, yet produce a better-quality deformation. More formally:

Problem 1: Given a source polygon S with n vertices and interior Ω, and a functional $E_S(g)$ defined on holomorphic functions g: $\Omega \cup S \to \mathbb{C}$, find complex numbers $u_1,..,u_n$ such that:

$$g_u(z) = \sum_{j=1}^{n} C_j(z) u_j$$

minimizes $E_S(g)$ among all possible choices of u.

We can think of solving Problem 1 as finding a *virtual target polygon* u, whose discrete Cauchy transform minimizes $E_S(g)$.

Next, we describe two choices for such a functional E, that can be applied to achieve useful effects in the context of planar shape deformation.

3.1 Szegö coordinates

As mentioned in the previous section, the main disadvantage of the Cauchy-Green coordinates is that the resulting deformation might be far from the target cage, see for example Fig. 5. An obvious improvement would result if we could find a better fit to the target cage within the Cauchy-Green subspace. Inspired by the Szegö projection, we define the following functional, which will help us generate a better fit.

Let f be a continuous complex function on the source polygon S. Define:

$$E_S^{Szegö}(g) = \oint_S \left| g(w) - f(w) \right|^2 ds \qquad (10)$$

We would like to minimize (10) within the Cauchy-Green subspace, which means, in other words, finding the virtual polygon u of Problem 1. However, so far we have only defined the Cauchy-Green coordinate functions on the *interior* of S, and not on S itself, where they are singular. Since we are now interested in the boundary values of the transformation image, we define the values of the coordinate functions on S to be their limit when approaching the boundary from the interior Ω:

$$C_j(z) = \lim_{z^{in} \to z} C_j(z^{in}), \quad z^{in} \in \Omega,\ z \in S$$

These limits always exist. We refer to Appendix A for the resulting formulae.

Having defined C_j also on S, we may now rewrite (10) as

$$E_S^{Szegö}(g) = \oint_S \left| g(w) - f(w) \right|^2 ds = \oint_S \left| \sum_{j=1}^{n} C_j(w) u_j - f(w) \right|^2 ds$$

To solve the problem in practice, we approximate the integral as a sum over a k-sampling of S. This sample may be expressed as a product of the n-vector z of the vertices of S with a $k \times n$ sampling matrix H, such that $w = Hz$ is a complex k-vector of points sampled on the polygon S. Now $C_j(w)$, $f(w)$ are also complex k-vectors (by evaluating the respective function at the entries of w), and we can express the functional in matrix form:

$$E_S^{Szegö}(g) = \left\| Cu - f_s \right\|_2^2$$

C is the complex $k \times n$ matrix whose columns are $C_j(w)$, namely, the values of the Cauchy-Green coordinate function on the sampled boundary, u is a complex n-vector and f_s a complex k-vector whose entries are $f(w)$. This is a simple linear least-squares problem over the complex numbers, and its solution is known to be [Bjö96]:

$$u^{Szegö} = C^{+} f_s = (C^{*}C)^{-1} C^{*} f_s \qquad (11)$$

where C^{+} is the pseudo-inverse of C and C^{*} is the conjugate transpose of C. Note that the size of the matrix $C^{*}C$ is $n \times n$, where n is the number of the vertices of the polygon S, hence this computation involves the inversion of a very modest sized matrix. Now that we have the virtual polygon $u^{Szegö}$, we define the deformation of an interior point $z \in \Omega$ to be:

$$g^{Szegö}(z) = \sum_{j=1}^{n} C_j(z) u_j^{Szegö} \qquad (12)$$

Fig. 6 shows an example of applying the Szegö coordinates for image deformation. As is evident from the image, the mapping is detail preserving (conformal), and the deformed shape remains close to the target cage.

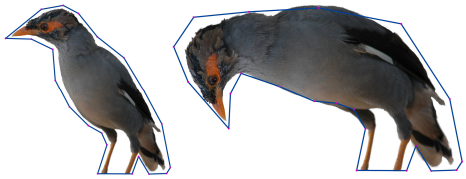


Figure 6: *Deformation of a bird using Szegö coordinates. The cage has 21 vertices.*

When working with deformation applications, time complexity is an issue, and we would like to avoid computing the virtual polygon $u^{Szegö}$ every time the user modifies the target function f. Fortunately, if the target function f is itself a polygon F, the deformation (12) can be made more compact by formulating it with barycentric coordinates applied to F.

Let $F = \{f_1, f_2, \ldots, f_n\}$ be the target polygon, then its sampled version f_s can also be obtained using the sampling matrix H: $f_s = HF$. Then:

$$u^{Szegö} = C^{+} HF$$

Thus the deformation $g_{S,f}$ is defined in terms of the *discrete Szegö coordinates* $G_j(z)$ of an interior point $z \in \Omega$:

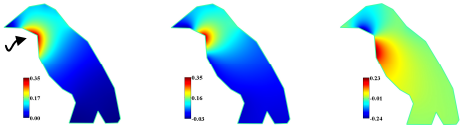
$$g_{S,f}(z) = \sum_{j=1}^{n} G_j(z) f_j$$

$$G_j(z) = \sum_{k=1}^{n} C_k(z) M_{k,j} \quad , \qquad M = C^{+} H$$

Szegö complex barycentric coordinates

M is an $n \times n$ matrix, called the *Szegö correction matrix*. It depends *only* on the source polygon S, thus may be computed once. Fig. 7 shows the color coding of the real and imaginary parts of the one of the Szegö coordinates from Fig. 6.

It is relatively easy to see that M has a number of useful properties, such as its rows sum to unity and $Mz=z$ (where z is a complex n-vector of the vertices of S), which imply that the Szegö coordinates also have constant and linear precision. Note that each Szegö coordinate function is a linear combination of *all* the Cauchy-Green coordinate functions. Thus they depend on *all* the vertices of the polygon S, so cannot be local three-point coordinates like the Cauchy-Green coordinates.



Absolute **Real** **Imaginary**

Figure 7: *Absolute, real and imaginary component values of the Szegö coordinate function of the marked point. Note that the real part is centered at the vertex and the imaginary part at the two adjacent edges.*

Next, we consider a different deformation paradigm which leads to another easily-minimized functional E.

3.2 Point-to-point Cauchy coordinates

The complex barycentric coordinates that we have discussed so far are "cage-based" deformations. The user modifies the location of the target cage vertices and thus controls the deformation. Unfortunately, in practice, the cage of a complicated shape can contain hundreds of vertices, and modifying each vertex independently to form the new cage is a time-consuming and unintuitive operation.

A much more intuitive user interface for the modeler will be manipulating a small number of positional constraints – control points - not necessarily on the shape boundary. This way the tedious task of manually positioning the cage vertices can be replaced by an efficient optimization process which derives suitable cage vertex positions automatically.

We now show how to easily adopt such an interface, while maintaining all the nice deformation properties that the Cauchy-Green coordinates provide. As the number of control points is typically much smaller than the number of cage vertices, it is possible to use the extra degrees of freedom to *regularize* the deformation. This means minimizing some aggregate differential quantity. Luckily, the Cauchy-Green coordinates have very simple derivatives and, like the transform itself, will also be a complex linear combination of the cage vertices. Hence we can easily minimize a functional combining both positional constraints inside the cage, and derivatives on the boundary of the cage.

Let f be a mapping from a set of p points $r_1, r_2, \ldots, r_p \in \Omega$, to the complex plane C, such that $f(r_k) = f_k$, and let:

$$E_S^{Smooth}(g) = \int_S \left|g''(w)\right|^2 ds \quad , \quad E_S^{Pts}(g) = \sum_{k=1}^{p} \left|g(r_k) - f_k\right|^2$$

Minimizing the first functional requires the mapping g to be as smooth as possible on the boundary of the cage. Minimizing the second functional imposes a finite set of positional constraints on the deformation in the interior of S.

Define the following combined weighted functional:

$$E_S^{PtoP}(g) = E_S^{Pts}(g) + \lambda^2 E_S^{Smooth}(g)$$

for some real λ. We will now attempt to solve Problem 1 for this functional.

Using the usual rules of linearity, the second derivative of the discrete Cauchy-Green transform (8) is:

$$g''(z) = \sum_{j=1}^{n} d_j(z) z_j$$

where

$$d_j(z) = \frac{1}{2\pi i}\left(\frac{1}{B_{j-1}(z)B_j(z)} - \frac{1}{B_j(z)B_{j+1}(z)}\right) \qquad (13)$$

As was the case for the discrete Szegö coordinates, E_S^{Smooth} is defined on the boundary of Ω, where d_j is singular when z is on the edge $e_j = (z_{j-1}, z_j)$, or on the edge $e_{j+1} = (z_j, z_{j+1})$. So here too, we need to use the limits of d_j for these cases:

$$d_j(z) = \lim_{z^{in} \to z} d_j(z^{in}),\ z^{in} \in \Omega,\ z \in S$$

It turns out that the limits everywhere except at the vertices are equal to $d_j(z)$, hence we can use (13) as is:

$$E_S^{smooth}(g) = \oint_S \left|g''(w)\right|^2 ds = \int_S \left|\sum_{j=1}^{n} d_j(w) u_j\right|^2 ds$$

As with the Szegö coordinates, we can rewrite the combined functional in matrix form:

$$E_S^{PtoP}(g) = \left\|Cu - f\right\|_2^2 + \lambda^2 \left\|Du\right\|_2^2$$

where D is a complex $k \times n$ matrix whose columns are $d_j(w)$. As in Section 3.1, C is the $p \times n$ matrix whose (i,j) entry is $C_j(r_i)$, u is a complex n-vector and f a complex k-vector whose entries are the positional constraints f_k. Some algebra leads to the Point-to-Point Cauchy-Green barycentric coordinates:

$$g_{S,f}(z) = \sum_{j=1}^{p} P_j(z) f_j$$

$$P_j(z) = \sum_{k=1}^{n} C_k(z) N_{k,j} \quad , \quad N = \left(A^+ = \begin{pmatrix} C \\ \lambda D \end{pmatrix}^+\right)_{1..p}$$

P2P Cauchy-Green complex barycentric coordinates

We now have only p barycentric coordinate functions - P_j. A^*A is an $n \times n$ matrix which can be easily inverted. N is the $n \times p$ matrix consisting of the first p columns of A^+. Using properties of N, it is relatively easy to show that the point-to-point Cauchy-Green coordinates have constant and linear precision.

Fig. 8 shows an example of a deformation produced by the P2P Cauchy-Green coordinates. As promised, the deformation is conformal, yet much easier to control since the user needs to manipulate only a small number of control points.

The number of coordinate functions P_j is now p - the number of control points, rather than n - the number of cage vertices (even though an n-vertex cage implicitly participates in the process). These functions are centered around the control points. Fig 9 shows the color-coded values of one of the coordinates functions of the shape of Fig. 8. As with the Szegö coordinates, the influence of a

coordinate function associated with a specific control points is local, in the sense that it affects only the areas of the cage which are geodesically close to it. This property guarantees that modifying one part of the shape will not affect another part, even if the *Euclidean* distance between them is small.

Figure 8: *A giraffe, and its deformation using Point-to-point Cauchy-Green coordinates, with 16 control points. The cage has 113 vertices.*

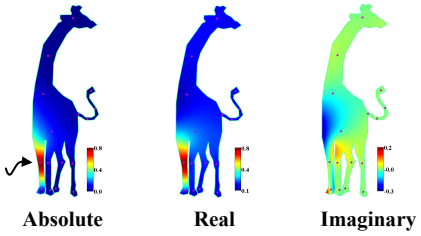


Figure 9: *P2P coordinate function of the marked point.*

4. Experimental Results

We have implemented an image deformation system using the discrete Szegö coordinates and the point-to-point Cauchy-Green coordinates as plugins to the Maya® commercial modeling and animation system. We compared their performance to that of existing state of the art planar deformation algorithms – the original cage-based Cauchy-Green coordinates [LLCO08] and control-point-based MLS [SMW06]. The image was represented as a texture map on a triangulation of the 2D domain, containing m vertices. Each of the n barycentric coordinates were pre-computed on these m vertices and stored in a dense complex $m \times n$ matrix B. A typical cage contains about n=150 vertices and m=15,000 interior vertices. The pre-process time to compute B is less than 10 seconds. The serial runtime complexity of a subsequent deformation operation is $O(mn)$ – the time required to multiply B by a complex n-vector. However, this multiplication was implemented in the GPU using Nvidia's CUDA programming language on an Nvidia Geforce 8800 GTX graphics card, resulting in a very significant speedup. For example, a single deformation of the "lady with whip" image (Fig. 12), which has m=12,000, n=272 and p=26 control points, takes approximately 0.05 milliseconds. Moreover, the pre-process time can be significantly reduced by implementing it also on the GPU, but we have not yet done this.

Before starting our qualitative comparison, we state upfront a few downsides of our coordinates. First and foremost, we currently do not have an extension to 3D deformations, as the original Green coordinates have. In addition, our coordinates generate only conformal deformations, and we do not currently support "as rigid as possible"-type deformations, as the MLS method does. Despite these shortcomings, we believe that our coordinates are useful in many 2D deformation scenarios, as we will now demonstrate.

4.1 Szegö vs. Cauchy-Green

As was already mentioned in previous sections, the main disadvantage of the Cauchy-Green coordinates is that the image of the domain, $g(\Omega)$, might be quite far from the target contour F. On the other hand, the Szegö coordinates produce (by definition) the conformal map in the Cauchy-Green subspace, whose boundary values are closest to the target polygon.

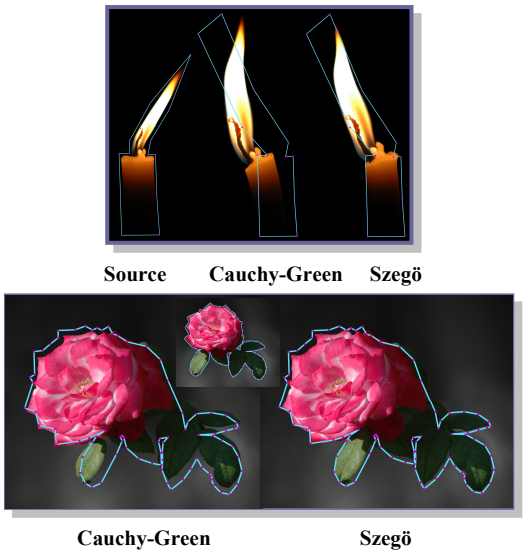


Figure 10: *Comparison of the Cauchy-Green coordinates and the Szegö coordinates. The Szegö coordinates better fit the cage, preventing an undesirable bend of the candle, and an undesirable movement of the rose leaves.*

Fig. 10 compares the deformation of some images using the Cauchy-Green coordinates and the Szegö coordinates. It can easily be seen that the Szegö coordinates provide a better match to the target contour. See, in addition, the accompanying video for a live demo of interactive deformations using both methods. There the Szegö coordinates can also be seen to be more stable during deformation.

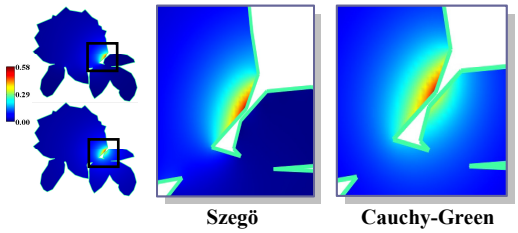


Figure 11: *Absolute value of the Szegö and Cauchy-Green coordinates. The Cauchy-Green coordinate "spills" into the leaf near it, whereas the Szegö coordinate does not.*

In addition to better fitting the contour, the Szegö coordinates have another important property – they are local

within the cage. Fig. 11 shows an example of the absolute value of the discrete Cauchy coordinates vs. the discrete Szegö coordinates on a given polygon. As is obvious from the image, the effect of the Szegö coordinates is more local in comparison to that of the Cauchy coordinates, for which deformation of one part of the shape may influence other parts which are geodesically far away.

The computation of the correction matrix M is done in the preprocess step, after the source cage is defined but before the actual deformation. Thus the runtime complexity of the preprocess step depends on k – the source contour sample density, but the runtime complexity of the actual deformation is exactly the same as that of the Cauchy-Green coordinates.

4.2 Point to point Cauchy-Green vs. MLS

In some scenarios cage deformations are not very useful. If the cage is complicated, as is usually the case for real-life shapes, cage-controlled deformations are less intuitive than a small number of simple control points strategically placed in the domain. This interface was also used by Igarashi *et al*. [IMH05] and the MLS system [SMW06].

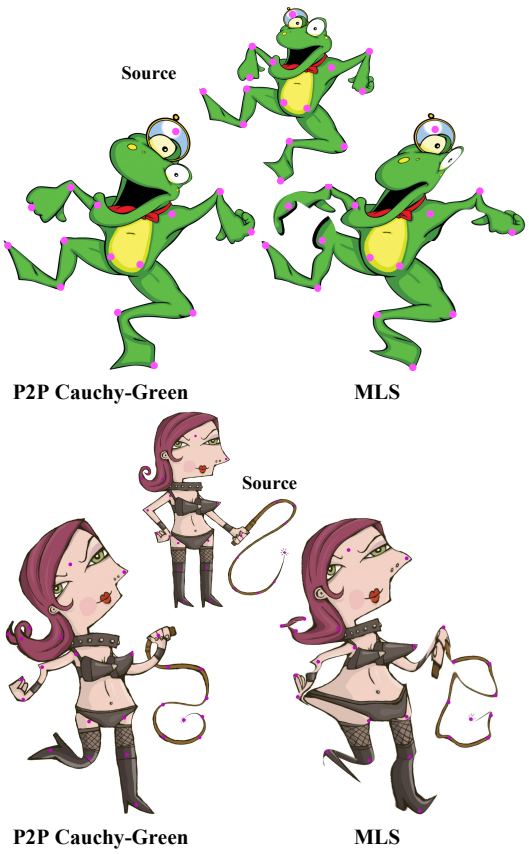


Figure 12: *Comparison of the P2P Cauchy-Green coordinates and the MLS coordinates. The P2P coordinates better handle control points whose Euclidean distance is small, yet their geodesic distance within the cage is large.*

Specifically, the MLS approach allows to define a set of control points and finds a local as-similar-as-possible or as-rigid as-possible deformation which satisfies the positional constraints imposed by the control points. As the point-to-point Cauchy-Green approach produces a conformal map by definition, the most relevant comparison is to the similarity version of MLS. Close examination of the MLS equations reveals that there exist complex barycentric coordinates which are equivalent to the similarity version.

Fig. 12 and the accompanying video compare some deformations generated by both methods with exactly the same constraints. The results show that point-to-point Cauchy-Green coordinates do much better at preserving the geometry of the shape. Using the additional information – the cage – this method is better at separating the extremities – e.g the hand of the frog from its leg. Since the MLS method deforms the entire plane, ignoring the geometry of the shape, separating the hand of the frog from its foot results in serious artifacts. On the other hand, as opposed to the MLS method, the point-to-point Cauchy-Green coordinates do not satisfy the positional constraints imposed by the control points precisely, rather treat them as soft constraints. This can be alleviated, if needed, by reducing the value of λ in the optimization problem of Section 3.2, albeit at the expense of the mapping smoothness.

As stated above, the MLS deformation can also be formulated as complex barycentric coordinates centered around the control points. Obviously, since the MLS has no knowledge of the cage, the coordinates' effect depends on Euclidean distances. This is clearly seen in Fig. 13 which compares the absolute values of the MLS coordinates and the point-to-point Cauchy-Green coordinates. Of course, it might be possible to modify MLS to use geodesic distances instead of Euclidean distances, but this would require discretization of the interior of the cage, which we wish to avoid.

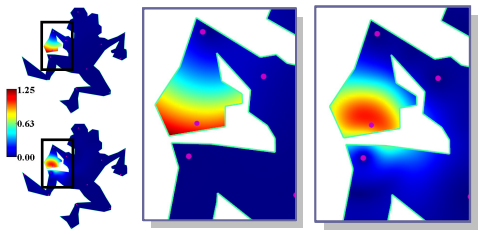


Figure 13: *Absolute value of the P2P and MLS coordinates of the point on the left hand. The MLS coordinate "spills" into the leg near it, whereas the P2P coordinate does not.*

The time complexity of deformation using P2P Cauchy-Green coordinates is very similar to that of the Szegö coordinates, as most of the computation is done in the preprocess step. Here, however, we have an additional benefit – the complexity of the deformation during user interaction is proportional to p - the number of control points, as opposed to n - the number of cage vertices, which is typically much larger.

5. Conclusions and Discussion

We have generalized the concept of barycentric coordinates from real numbers to complex numbers, and provided a few examples of known and new coordinates which can be expressed quite simply in this framework. In addition, we have shown how the new coordinates can be used in the context of planar shape deformations to produce results

superior to state-of-the-art methods, at a small extra cost in computational complexity, in pre-process time only.

We believe there is still much research to be done on the theory and applications of complex barycentric coordinates. Their most obvious drawback is the fact that they are only defined for planar domains, which would seem to rule out the possibility of using them for 3D shape deformation. However, this should still be possible if the complex numbers are interpreted geometrically and analogs found in higher dimensions.

Another important research direction is relaxing the conformality requirement. As real barycentric coordinates are a special case of complex coordinates, obviously non-conformal complex coordinates exist. We have seen that the MLS coordinates are such. An interesting challenge is to find non-conformal complex coordinates which will generate the more useful quasi-conformal or "as rigid as possible"-type deformations.

There are some connections between our method and other deformation methods. First, the formulation of the Green Coordinates is a special case of the Boundary Element Method formulation, where constant boundary elements are used. Second, the functional minimized for the P2P coordinates has some resemblance to functionals used with Radial Basis Functions. As both methods were previously used for shape warping and deformation [JP99,Boo89], it would be interesting to explore the relationship between these methods and ours.

One theoretical issue which we haven't addressed at all, is the connection between complex barycentric coordinates and the so-called "primal/dual ratio". As Mercat [Mer08] pointed out, complex primal/dual ratios will arise when the primal and dual edges are not orthogonal. We believe more insight into complex barycentric coordinates can be gained by studying more these concepts.

Discrete conformal maps have been used for many years for 3D mesh parameterization, where the domain is a discrete set of triangles. In fact, many of the most common recipes for real barycentric coordinates were motivated by this problem. The Least Squares Conformal Mapping (LSCM) method [LPRM02] for free-boundary mesh parameterization is based on a discretization of the Cauchy-Riemann equations on triangles, achieving an “As-Similar-As-Possible” effect. Adding soft positional constraints to some of the boundary vertices in the linear LSCM system allows extra control so that the deformed mesh is close in some way to some target geometry. The advantage of the Szegö coordinates proposed here is that they achieve a similar effect without any discretization of the plane. It would be interesting to develop the connection between LSCM and Szegö coordinates.

Acknowledgment

We wish to thank NVIDIA for their contribution of the GeForce 8800GTX graphics card.

References

[Ahl79] AHLFORS L.: *Complex Analysis, 3rd Edition*. McGraw-Hill Science, (1979)

[Bel92] BELL S.-R.: *The Cauchy Transform, Potential Theory and Conformal Mapping*. CRC-Press, (1992)

[Bel06] BELYAEV A.: On transfinite barycentric coordinates. *Proc. Symp. Geometry Processing* (2006), 89-99.

[Bjö96] BJÖRK A.: *Numerical Methods for Least Squares Problems*. SIAM, (1996).

[BS08] BOTSCH M., SORKINE O.: On linear variational surface deformation methods. *IEEE Transactions on Visualization and Computer Graphics, 14*, 1, (2008), 213-230

[Boo89] BOOKSTEIN F.L.: Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 11, 6, (1989), 567-585

[DMA02] DESBRUN M., MEYER M., ALLIEZ P.: Intrinsic parameterizations of surface meshes. *Computer Graphics Forum, 21* (2002), 209-218.

[FHK06] FLOATER M. S., HORMANN K., KÒS G.: A general construction of barycentric coordinates over convex polygons. *Adv. Comp. Math. 24*, 1-4 (2006), 311-331.

[FKR05] FLOATER M. S., KÒS G., REIMERS M.: Mean-value coordinates in 3D. *Comp. Aided Geom. Design 22* (2005), 623-631.

[Flo03] FLOATER M. S.: Mean-value coordinates. *Comp. Aided Geom. Design 20*, 1 (2003), 19-27.

[HF06] HORMANN K., FLOATER M. S.: Mean-value coordinates for arbitrary planar polygons. *ACM Trans. Graph.* (2006), 1424-1441.

[IMH05] IGARASHI T., MOSCOVICH T., HUGHES J.-F.: As-Rigid-As-Possible shape manipulation. *ACM Trans. Graph. (Proc. SIGGRAPH),. 24*, 3, (2005), pp.1134-1141.

[JMD*07] JOSHI P., MEYER M., DEROSE T., GREEN B., SANOCKI T.: Harmonic coordinates for character articulation. *ACM Trans. Graph. (Proc. SIGGRAPH), 26*, 3 (2007), 71.

[JP99] JAMES D.L., PAI D.K.: ArtDefo: Accurate real time deformable objects. *Proc. SIGGRAPH* (1999)

[JSW05] JU T., SCHAEFER S., WARREN J.: Mean value coordinates for closed triangular meshes. *ACM Trans. Graph. (Proc. SIGGRAPH), 24*, 3 (2005), 561-566.

[LKCOL07] LIPMAN Y., KOPF J., COHEN-OR D., LEVIN D.: GPU-assisted positive mean value coordinates for mesh deformations. *Proc. Symp. Geometry Processing* (2007), pp. 117-123.

[LLCO08] LIPMAN Y., LEVIN D., COHEN-OR D.: Green coordinates. *ACM Trans. Graph. (Proc. SIGGRAPH), 27*, 3 (2008).

[LPRM02] LÉVY B., PETITJEAN S., RAY N., MAILLOT J.: Least squares conformal maps for automatic texture atlas generation. *Proc. SIGGRAPH* (2002).

[LS08] LANGER T., SEIDEL H.-P.: Higher order barycentric coordinates. *Computer Graphics Forum (Proc. Eurographics) 27* (2), (2008), 459-466.

[Mer08] MERCAT C.: Discrete complex structure on surfel surfaces. *Lecture Notes in Computer Science,* 4992, (2008), 153-164.

[SAPH04] SCHREINER J., ASIRVATHAM A., PRAUN E., HOPPE H.: Inter-surface mapping. *ACM Trans. Graph. (Proc. SIGGRAPH). 23*, 3 (2004), 870-877.

[SMW06] SCHAEFER S., MCPHAIL T., WARREN J.: Image deformation using moving least squares. *ACM Trans. Graph. (Proc. SIGGRAPH). 23*, 3 (2004),

[WSHD07] WARREN J. D., SCHAEFER S., HIRANI A. N., DESBRUN M.: Barycentric coordinates for convex sets. *Adv. Comput. Math. 27*, 3 (2007), 319-338.

Appendix A: Limits of the discrete Cauchy-Green coordinates

The discrete Cauchy coordinates on the polygon S are:

$$C_j(z)=\frac{1}{2\pi i}\begin{cases}\dfrac{B_{j+1}(z)}{A_{j+1}}\log\left(\dfrac{B_{j+1}(z)}{B_j(z)}\right)-\dfrac{B_{j-1}(z)}{A_j}\log\left(\dfrac{B_j(z)}{B_{j-1}(z)}\right) & z\notin\{e_j,e_{j+1}\}\\ \left(1+\dfrac{(1-t)A_j}{A_{j+1}}\right)\log\left(1+\dfrac{A_{j+1}}{(1-t)A_j}\right)+t\left(\log\dfrac{1-t}{t}+\pi i\right) & z\in e_j\Rightarrow\ z=z_{j-1}(1-t)+z_jt,t\in(0,1)\\ (1-t)\left(\log\dfrac{1-t}{t}+\pi i\right)-\left(1+\dfrac{t\cdot A_{j+1}}{A_j}\right)\log\left(1+\dfrac{A_j}{t\cdot A_{j+1}}\right) & z\in e_{j+1}\Rightarrow\ z=z_j(1-t)+z_{j+1}t,t\in(0,1)\end{cases}$$

$$C_j(z)=\frac{1}{2\pi i}\begin{cases}M_j\log N_j & z=z_{j-1}\\ \log\dfrac{-A_{j+1}}{A_j} & z=z_j\\ N_j\log M_j & z=z_{j+1}\end{cases},\qquad M_j=1+\frac{A_j}{A_{j+1}},\quad N_j=1+\frac{A_{j+1}}{A_j}$$