

Bijjective Composite Mean Value Mappings

T. Schneider¹ K. Hormann¹ M. S. Floater²

¹Faculty of Informatics, Università della Svizzera italiana, Lugano, Switzerland

²Department of Mathematics, University of Oslo, Norway

Abstract

We introduce the novel concept of composite barycentric mappings and give theoretical conditions under which they are guaranteed to be bijective. We then focus on mean value mappings and derive a simple procedure for computing their Jacobians, leading to an efficient GPU-assisted implementation for interactively designing composite mean value mappings which are bijective up to pixel resolution. We provide a number of examples of 2D image deformation and an example of 3D shape deformation based on a natural extension of the concept to spatial mappings.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Line and curve generation G.1.1 [Numerical Analysis]: Interpolation—Interpolation formulas

1. Introduction

A *barycentric mapping* between a *source polygon* $\Omega \subset \mathbb{R}^2$ with $n \geq 3$ source vertices $v_i, i = 1, \dots, n$, ordered anticlockwise, and a *target polygon* $\hat{\Omega} \subset \mathbb{R}^2$ with the same number of target vertices \hat{v}_i , is a mapping

$$f: \Omega \rightarrow \hat{\Omega}, \quad f(v) = \sum_{i=1}^n b_i(v) \hat{v}_i, \quad (1)$$

where the functions $b_i: \Omega \rightarrow \mathbb{R}, i = 1, \dots, n$ are a set of *barycentric coordinates* with respect to Ω . That is, the b_i form a *partition of unity*,

$$\sum_{i=1}^n b_i(v) = 1, \quad v \in \Omega,$$

allow us to express any point as an *affine combination* of the vertices,

$$\sum_{i=1}^n b_i(v) v_i = v, \quad v \in \Omega, \quad (2)$$

and satisfy the *Lagrange property*

$$b_i(v_j) = \delta_{i,j}, \quad j = 1, \dots, n,$$

so that f maps the source vertices to the target vertices,

$$f(v_j) = \hat{v}_j, \quad j = 1, \dots, n. \quad (3)$$

Some barycentric coordinates have the additional property of being piecewise linear along the boundary $\partial\Omega$ of Ω , implying that f maps $\partial\Omega$ to $\partial\hat{\Omega}$ in a piecewise linear fashion.

One of the main applications of barycentric mappings in computer graphics is *image warping*, where a *source image* $I: \Omega \rightarrow \mathcal{C}$ over Ω with respect to some *colour space* \mathcal{C} is deformed into a *target image* $\hat{I}: \hat{\Omega} \rightarrow \mathcal{C}$ by simply setting

$$\hat{I}(f(v)) = I(v). \quad (4)$$

The vertex interpolation property (3) guarantees that the user has intuitive control over the target image and can easily modify the latter by interactively moving the target vertices \hat{v}_i . However, according to (4) the target image is well-defined if and only if f is *bijective*. Indeed, if f is not *surjective*, then \hat{I} may be undefined for some $\hat{v} \in \hat{\Omega}$, and if f is not *injective*, so that $f(v) = f(w) = \hat{v}$ for some $v \neq w$, then the definition of $\hat{I}(\hat{v})$ is ambiguous.

Unfortunately, a set of barycentric coordinates that yields bijective mappings for any choice of source and target polygons does not exist [Jac12]. However, by splitting the mapping into a finite number of steps it is possible to define a *composite barycentric mapping* that bijectively maps from source to target polygon (see Section 2.2). Bijectivity is also crucial for the application of barycentric mappings to *shape deformation*, where some object contained in Ω is smoothly deformed by applying f to the object points.

1.1. Related work

While barycentric coordinates are unique for simplices, they can be generalized in several ways to arbitrary polygons and

polytopes in higher dimensions, and several such generalized barycentric coordinates have been found in the past few years. In \mathbb{R}^2 , the most well-known are Wachspress [Wac75], discrete harmonic [EDD*95, PP93], and mean value coordinates [Flo03], which are all given in closed form and can be evaluated efficiently. Harmonic [JMD*07] and maximum entropy coordinates [HS08] are more costly to evaluate, but have the advantage of being positive inside arbitrary polygons. All of these, as well as the recently discovered Poisson coordinates [LH13], are piecewise linear along $\partial\Omega$, but giving up this property can be beneficial if conformal mappings are preferred [WBCG09].

Mean value coordinates have also been extended to \mathbb{R}^3 [FKR05, JSW05, LBS06], with a particular focus on their application to shape deformation. Some of the shortcomings of 3D mean value mappings, which result from the coordinates being negative for non-convex polyhedra, have been overcome by using positive mean value [LKCOL07], harmonic [JMD*07], or Green coordinates [LLCO08], but none of these spatial mappings are guaranteed to be bijective and can lead to fold-overs in the deformed shapes.

Image warping has first been mentioned as an application of barycentric mappings in the context of mean value [HF06] and transfinite Wachspress coordinates [WSHD07], but neither these nor any of the follow-up work [JBPS11, MS10, WG10] is guaranteed to avoid fold-overs in the warped image. Modifying the interpolation speed along the boundary of Ω has the potential to overcome this problem [WBCGH11], but it is unclear if this approach works for any arbitrary source and target polygons.

So far, bijectivity can only be guaranteed for the special case of Wachspress mappings between convex polygons [FK10] or by triangulating Ω and computing a piecewise linear mapping which preserves the orientation of each triangle [ACOL00, Lip12], but this solution comes at the price of giving up smoothness.

1.2. Contributions

We introduce the novel concept of *composite barycentric mappings*, which can be used to design smooth and bijective mappings between arbitrary polygons (Section 2). The key observation is that any barycentric mapping is bijective if source and target polygon are sufficiently close (Section 2.1). Hence, by creating sufficiently many intermediate polygons and successively mapping from one to the next, it is possible to get a composite mapping that is bijective.

A similar approach has been described in the context of free-form deformations with tensor-product B-spline mappings [GD01], but we are able to provide sufficient bounds to guarantee bijectivity, which depend on the gradient of the barycentric coordinates used to define the mapping (Section 2.2). For the special case of mean value mappings, these bounds can provably be satisfied for mappings between any

convex polygons and our numerical examples suggest that this also holds for arbitrary polygons (Section 3).

We further discuss how to implement our approach efficiently on the GPU, allowing the user to design *visually bijective* mappings, that is, bijective up to pixel resolution, by interactively modifying the target vertices (Section 4). As an essential ingredient of our implementation, we derive a simple procedure for computing the Jacobian of a mean value mapping (Section 4.2). We briefly mention how to extend the idea to spatial mappings (Section 4.5) and conclude with a conjecture regarding the inverses of composite mean value mapping (Section 5).

2. Theoretical background

Let us denote the *partial derivatives* of the barycentric mapping $f = (f^1, f^2)$ in (1) at $v = (v^1, v^2) \in \Omega$ by $\partial_k f(v) = \partial f / \partial v^k$, $k = 1, 2$, and the *gradient* of f by $\nabla f = (\partial_1 f, \partial_2 f)$. We further denote the *dot product* of two vectors $u, v \in \mathbb{R}^2$ by $u \cdot v$ and the *2D cross product* by $u \times v = \det(u, v)$.

As we consider only source and target polygons without self-intersections and assume that the barycentric coordinates b_i are at least continuously differentiable, it is clear that the barycentric mapping f is always surjective. A sufficient condition for the injectivity of f is that its *Jacobian*,

$$J_f = \begin{vmatrix} \partial_1 f^1 & \partial_2 f^1 \\ \partial_1 f^2 & \partial_2 f^2 \end{vmatrix} \quad (5)$$

is strictly positive in Ω [MO63]. As it follows from (2) that a barycentric mapping between identical source and target polygons is the identity with $J_f(v) = 1$ for all $v \in \Omega$, it is reasonable to expect that a small perturbation of the target vertices keeps J_f positive and the mapping bijective.

2.1. Perturbed target polygons

It turns out that the maximum possible perturbation is closely related to the gradient of the barycentric coordinates b_i . Let us first consider the perturbation of a single vertex.

Lemma 1 Consider the target polygon with vertices $\hat{v}_i = v_i + u$ for some i and $\hat{v}_j = v_j$ for $j \neq i$. The barycentric mapping f in (1) is injective if the displacement vector $u = (u^1, u^2)$ satisfies $\|u\| < 1/M_i$, where

$$M_i = \sup_{v \in \Omega} \|\nabla b_i(v)\|.$$

Proof Substituting the target vertices in (1), the linear precision property (2) implies that

$$f(v) = v + b_i(v)u$$

for any $v \in \Omega$ and

$$J_f(v) = \begin{vmatrix} 1 + \partial_1 b_i(v)u^1 & \partial_2 b_i(v)u^1 \\ \partial_1 b_i(v)u^2 & 1 + \partial_2 b_i(v)u^2 \end{vmatrix} = 1 + \nabla b_i(v) \cdot u.$$

Therefore,

$$J_f(v) \geq 1 - |\nabla b_i(v) \cdot u| \geq 1 - \|u\| \|\nabla b_i(v)\|,$$

which is strictly positive if $\|u\| < 1/M_i$. \square

This result extends nicely to a perturbation of all vertices.

Lemma 2 Consider the target polygon with vertices $\hat{v}_i = v_i + u_i$ for $i = 1, \dots, n$. The barycentric mapping f in (1) is injective if the maximum displacement distance satisfies

$$d = \max_{1 \leq i \leq n} \|u_i\| < \frac{\sqrt{5} - 1}{2M},$$

where $M = M_1 + \dots + M_n$.

Proof These target vertices imply that

$$f(v) = v + \sum_{i=1}^n b_i(v) u_i$$

for any $v \in \Omega$ and so, with $u_i = (u_i^1, u_i^2)$,

$$\begin{aligned} J_f(v) &= \begin{vmatrix} 1 + \sum_{i=1}^n \partial_1 b_i(v) u_i^1 & \sum_{i=1}^n \partial_2 b_i(v) u_i^1 \\ \sum_{i=1}^n \partial_1 b_i(v) u_i^2 & 1 + \sum_{i=1}^n \partial_2 b_i(v) u_i^2 \end{vmatrix} \\ &= 1 + \sum_{i=1}^n \nabla b_i(v) \cdot u_i + \sum_{i,j=1}^n \partial_1 b_i(v) \partial_2 b_j(v) (u_i \times u_j). \end{aligned}$$

Therefore,

$$J_f(v) \geq 1 - Md - M^2 d^2,$$

which is strictly positive if $Md < (\sqrt{5} - 1)/2$. \square

2.2. Composite barycentric mappings

Lemma 2 suggests to “split” a barycentric mapping from source to target polygon into a finite number of steps, where each step perturbs the vertices only slightly.

To this end, suppose that $\varphi_i: [0, 1] \rightarrow \mathbb{R}^2$, $i = 1, \dots, n$ are a set of continuous *vertex paths* between each source vertex $v_i = v_i^0 = \varphi_i(0)$ and its corresponding target vertex $\hat{v}_i = v_i^1 = \varphi_i(1)$. Further assume that the gradients of the barycentric coordinates b_i^t associated with any of the polygons Ω^t with vertices $v_i^t = \varphi_i(t)$ are all bounded by a common constant

$$M^* = \max_{1 \leq i \leq n} \sup_{t \in [0, 1]} \sup_{v \in \Omega^t} \|\nabla b_i^t(v)\|. \quad (6)$$

Definition 3 Let $\tau = (t_0, t_1, \dots, t_m)$ with $t_0 = 0$, $t_m = 1$, and $t_k < t_{k+1}$ for $k = 0, \dots, m-1$ be a *partition* of $[0, 1]$ and f_k be the barycentric mapping from Ω^{t_k} to $\Omega^{t_{k+1}}$, based on the barycentric coordinates $b_i^{t_k}$. The mapping

$$f_\tau = f_{m-1} \circ f_{m-2} \circ \dots \circ f_0,$$

is called a *composite barycentric mapping* from Ω^0 to Ω^1 .

Denoting the maximum displacement distance between Ω^{t_k} and $\Omega^{t_{k+1}}$ by

$$d_k = \max_{1 \leq i \leq n} \|v_i^{t_k} - v_i^{t_{k+1}}\|,$$

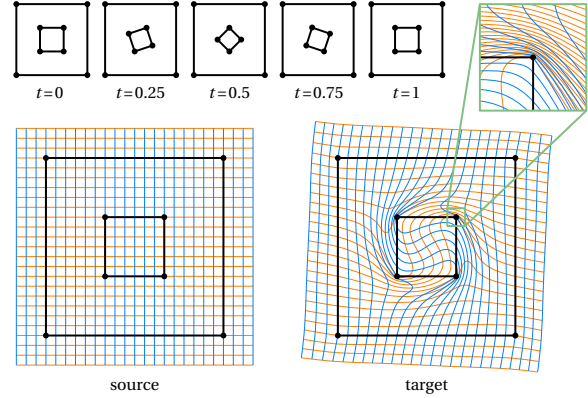


Figure 1: Example of a composite mean value mapping with 1000 uniform steps. The resolution of the grid is increased by a factor of four in the close-up.

it follows from Lemma 2 that f_τ is bijective if

$$d_\tau = \max_{0 \leq k < m} d_k < \frac{\sqrt{5} - 1}{2nM^*}. \quad (7)$$

Theorem 4 Under the assumption (6), there exists some τ , such that f_τ is bijective.

Proof It follows from the continuity of φ_i that there exists a partition τ^i such that $\|\varphi_i(t_k^i) - \varphi_i(t_{k+1}^i)\| < (\sqrt{5} - 1)/(2nM^*)$ for each k . Condition (7) is then satisfied for $\tau = \cup_{i=1}^n \tau^i$. \square

In particular, there exists some $m \in \mathbb{N}$ such that the *uniform partition* τ_m with $t_k = k/m$ gives a bijective composite barycentric mapping f_{τ_m} .

3. Practical considerations

While the construction of composite barycentric mappings is independent of the particular choice of underlying barycentric coordinates, we prefer to use mean value coordinates [Flo03], because they are well-defined for any simple polygon, even sets of nested simple polygons, and they have a simple formula that can be evaluated efficiently [HF06].

Figure 1 shows an example of a composite mean value mapping for two nested squares, with the interior square rotated by 90 degrees in the target configuration. We use linear vertex paths $\varphi_i(t) = (1-t)v_i + t\hat{v}_i$ in this example and the uniform partition τ_{1000} , and we verified numerically that condition (7) is satisfied by this choice. Note that Figure 1 visualizes the deformation induced by f for a domain Ψ that is larger than Ω , which is possible because mean value coordinates are well-defined for any point $v \in \mathbb{R}^2$ and not restricted to Ω . While the mapping is still guaranteed to be injective, global fold-overs can potentially occur if the image $f(\partial\Psi)$ of the boundary of Ψ happens to self-intersect, although we never experienced such a behaviour in our experiments.

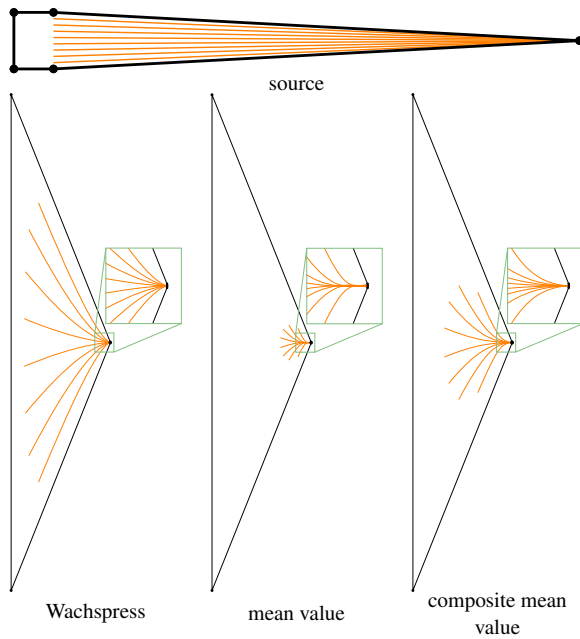


Figure 2: Example of a convex source (top) and a convex target polygon (bottom), taken from [FK10], for which the mean value mapping (middle) is not bijective, but the Wachspress (left) and the uniform composite mean value mapping $f_{\tau_{10}}$ (right) are.

3.1. Choosing the vertex paths

The vertex paths ϕ_i must be chosen such that all intermediate polygons Ω^k are simple, because mean value coordinates are not well-defined for polygons with self-intersections. Linear vertex paths work well for the example in Figure 1, but they are not guaranteed to avoid such artefacts in general.

Instead, we use the method by Sederberg et al. [SGWM93], which constructs Ω^t by linearly interpolating the signed turning angles at v_i and \hat{v}_i , and by finding edge lengths that are as close as possible to the linearly interpolated edge lengths. As both quantities are invariant under translation and rotation, the interpolated polygon is unique only up to a rigid body transformation. In our implementation, we specify the translation such that the barycentre of Ω^t aligns with the linear interpolation of the barycentres of source and target polygon, and we fix the rotation by linearly interpolating the angles between the horizontal axis and the first edges of Ω and $\hat{\Omega}$, as suggested by Sederberg et al. We further scale Ω^t such that the area of the polygon varies linearly between source and target. Overall, this provides a smooth deformation which is intuitive for the user, but we should point out that the composite mapping f_τ does not depend on this particular choice, because mean value coordinates are invariant with respect to similarity transformations [HF06].

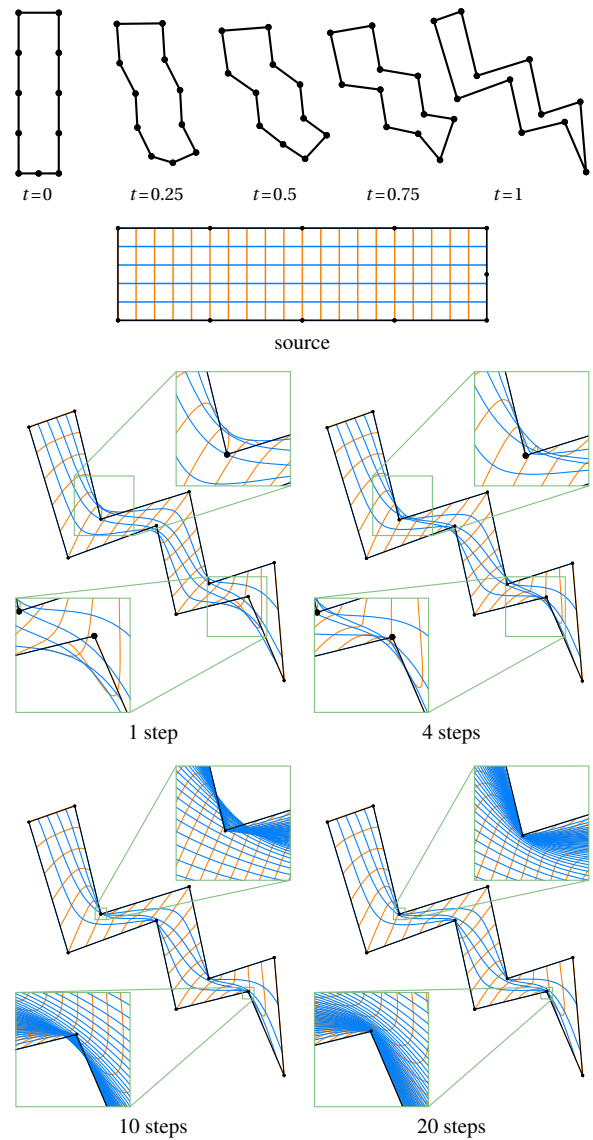


Figure 3: Examples of uniform composite mean value mappings for different numbers of uniform steps.

Another advantage of Sederberg et al.'s method is that it preserves convexity by construction. Thus, if source and target polygon are both convex, it is possible to derive an upper bound on M^* in (6) explicitly by following the considerations in [RGB12, Theorem 4.3], and hence deduce a partition τ that guarantees the bijectivity of f_τ . We carried out these calculations for the example in Figure 2 and found that a uniform composite mean value mapping with $m = 10$ steps is provably bijective, although we admit that it is probably preferable to simply use a bijective Wachspress mapping in this example.

In practice, however, we suggest to start with a small number of uniform steps, to test for the bijectivity of the mapping numerically, and to uniformly refine the partition if needed. A result of this strategy can be seen in Figure 3. An even more efficient and adaptive alternative is described in the next section.

A potential drawback of Sederberg et al.'s method is that Ω' may self-intersect, invalidating the overall construction. While this never occurred in all our examples, this does not affect the theoretical considerations in Section 2, and other methods of choosing vertex paths can be used in practice.

4. Efficient implementation

On the one hand, a sufficiently large number of m steps guarantees the bijectivity of a composite mean value mapping, as shown in the previous section, but on the other hand, it is important to choose m small, because the computations scale linearly with m . To find such a small m that still gives mappings which are bijective up to pixel accuracy, we adopt the following strategy, which relies on the minimum of the Jacobian

$$J_{\min} = \min_{v \in \Omega} (J_f(v)).$$

We first explain how to exploit the condition on the positivity of J_{\min} to construct a binary partition τ that guarantees bijectivity (Section 4.1). We then describe how to efficiently compute J_{\min} in order to have a real-time application (Section 4.2), and finally illustrate how to use the two previous ingredients to efficiently compute the composite mapping (Section 4.3).

4.1. Creating binary partitions

Suppose we start with the initial partition $\tau = (0, 1)$ and consider the classical mean value mapping $f_\tau: \Omega^0 \rightarrow \Omega^1$. We can now compute J_{\min} for this mapping and check for positivity. If J_{\min} is negative, and hence the mapping is not bijective, we can try to restore bijectivity by inserting 0.5 into the partition τ and considering the intermediate polygon $\Omega^{0.5}$. We then recalculate J_{\min} for the two mappings

Listing 1: Pseudo-code of the algorithm for finding a binary partition with a small number of steps.

```

function checkJacobian( $a, b$ )
     $J_{\min} \leftarrow \text{computeJmin}(a, b)$ 
    if  $J_{\min} \leq 0$  and  $|b - a| > \epsilon$  then
         $c \leftarrow (a + b)/2$ 
         $\tau \leftarrow \tau \cup c$ 
        checkJacobian( $a, c$ )
        checkJacobian( $c, b$ )
    end
end

```

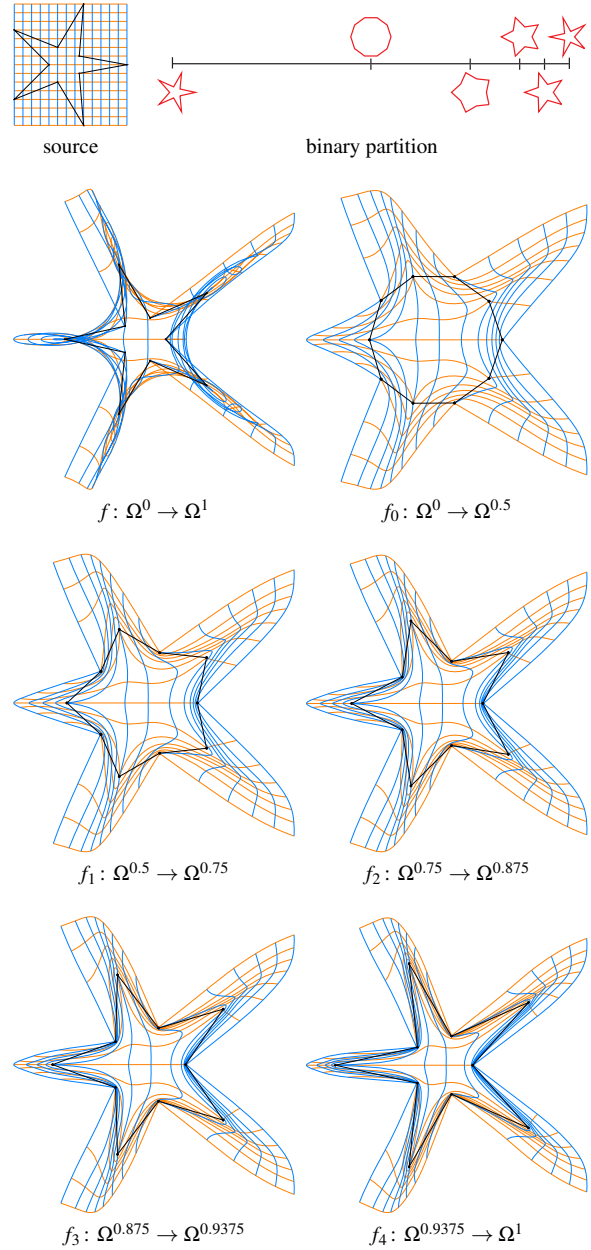


Figure 4: Example of a composite mean value mapping with a 5-step binary partition.

$f_1: \Omega^0 \rightarrow \Omega^{0.5}$ and $f_2: \Omega^{0.5} \rightarrow \Omega^1$ and test both for positivity. We continue this binary refinement of τ until J_{\min} is positive for all mappings f_k or if the distance between two steps becomes numerically too small, that is, smaller than some threshold $\epsilon > 0$. We used $\epsilon = 0.01$ in our examples. The pseudo-code in Listing 1 illustrates a recursive algorithm for finding such a binary partition. Given two domains Ω^a and Ω^b for $0 \leq a < b \leq 1$, the algorithm's core consists of

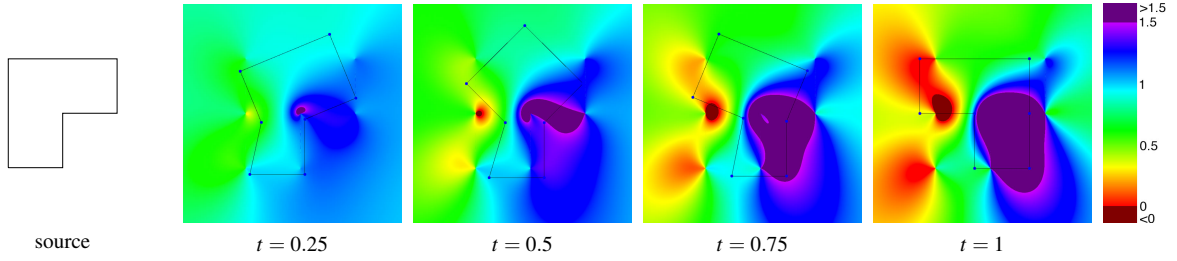


Figure 5: Colour-coded plots of J_f for the mean value mapping $f: \Omega^0 \rightarrow \Omega^t$ for different values of t .

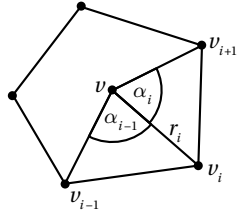


Figure 6: Notation used for the definition of mean value coordinates.

computing J_{\min} for the mapping between Ω^a and Ω^b . Then, by checking the positivity of the minimum, the algorithm decides whether to introduce an intermediate step. Overall, this produces a binary partition which guarantees that the corresponding composite mapping is bijective and consists of a small number of intermediate steps. Figure 4 shows an example of the result of this algorithm.

4.2. Efficiently computing J_{\min}

In order to compute J_{\min} , we need a formula for the partial derivatives of the mean value coordinates

$$b_i(v) = \frac{w_i(v)}{\sum_{i=1}^n w_i(v)},$$

where

$$w_i = \frac{\tan(\alpha_{i-1}/2) + \tan(\alpha_i/2)}{r_i} \quad (8)$$

and α_i is the angle between $v_i - v$ and $v_{i+1} - v$ and r_i is the distance between v and v_i ; see Figure 6. To efficiently evaluate expression (8), we express the tangent as

$$t_i = \tan(\alpha_i/2) = \frac{\sin \alpha_i}{1 + \cos \alpha_i} = \frac{s_i \times s_{i+1}}{r_i r_{i+1} + s_i \cdot s_{i+1}},$$

where $s_i = v_i - v$.

The gradient of b_i can now be derived from the gradients of the cross product,

$$\nabla(s_i \times s_{i+1}) = (v_{i+1} - v_i)^\perp = \begin{pmatrix} v_i^2 - v_{i+1}^2 \\ v_{i+1}^1 - v_i^1 \end{pmatrix},$$

the distance,

$$\nabla r_i = \frac{-s_i}{r_i},$$

and the dot product

$$\nabla(s_i \cdot s_{i+1}) = -(s_i + s_{i+1}),$$

and by using the chain rule, which first gives

$$\nabla t_i = \frac{\nabla(s_i \times s_{i+1}) - t_i(\nabla r_i r_{i+1} + r_i \nabla r_{i+1} + \nabla(s_i \cdot s_{i+1}))}{r_i r_{i+1} + s_i \cdot s_{i+1}},$$

then

$$\nabla w_i = \frac{\nabla t_{i-1} + \nabla t_i - w_i \nabla r_i}{r_i},$$

and finally

$$\nabla b_i = \frac{\nabla w_i - b_i \nabla \sum w_i}{\sum w_i}. \quad (9)$$

In order to efficiently compute (9), we opt to evaluate it on the GPU using a vertex shader. This shader receives the vertices of the source and target polygons, computes the Jacobian, and writes it as a colour. Figure 5 shows some examples of the values of J_f computed this way. The major drawback of this approach is that the output of the shader is a full resolution image, which must be read back and analysed by the CPU, which should be avoided if possible.

The vertex shader knows only the position of the currently processed vertex, hence finding the global minimum is impossible. To circumvent this problem we exploit the *z-buffer*. This particular OpenGL-buffer is used to determine which object is closer to the camera. Using this trick we modify the shader to “collapse” all the points to the same (x, y) position and we artificially set the z -component to the value of the Jacobian (clamped to zero if negative). In this way the generated image contains only one pixel which is the one closest to the camera and the colour of this pixel corresponds to J_{\min} . At the end, the CPU reads only one pixel and produces a sequence of domains Ω^i , $i = 1, \dots, n$ such that the mapping between two successive domains is visually bijective.

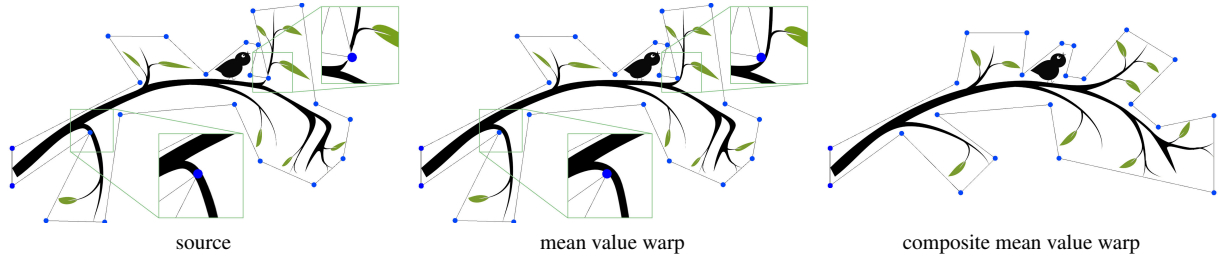


Figure 7: Comparison of image warping using a non-bijective mean value mapping and a visually bijective composite mean value mapping with 9 steps from a binary partition.

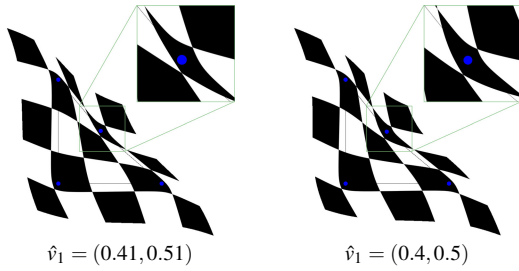


Figure 8: Composite mean value mappings from a unit square $[0, 1]^2$ to a deformed square with upper right vertex \hat{v}_1 . The slight change of position induces a change in the binary partition and thus gives different composite mappings (see close-up).

4.3. Computing composite mean value mappings

For computing the overall composite mapping, we decide again to implement the algorithm on the GPU using a vertex shader. The shader receives two successive domains Ω^k and Ω^{k+1} and moves the vertex position according to the mapping f_k . We then exploit the *transform feedback buffer*, a technique that allows to alter the rendering pipeline of OpenGL such that the primitives, in our case the vertex positions, are written into a buffer object. Hence the algorithm becomes a “ping pong” between two *transform feedback buffers*. We use the first buffer as output (where the vertex position is written) and the second as input (from where the positions are read). In the next step the role of the two buffers is inverted and so on. At the end, the last output buffer contains the vertex positions of the composite mapping. Figure 4 shows the warping of the star for different choices of intermediate steps. When the partition is sufficiently refined, the morphing is visually bijective and prevents fold-overs. The real-world example in Figure 7 shows how the non-bijectivity of the direct mean-value mapping is avoided by our composite approach with just 9 intermediate steps.

4.4. Continuous binary partitions

Our initial tests showed that visual artefacts can appear when interactively moving a target vertex leads to the creation of

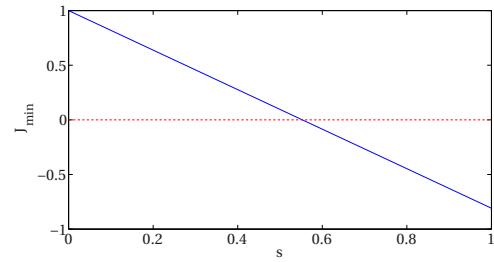


Figure 9: J_{\min} for different warps of the unit square to a unit square with the upper right vertex moved to the position $(1-s, 1-s)$ for $s = 0.005$.

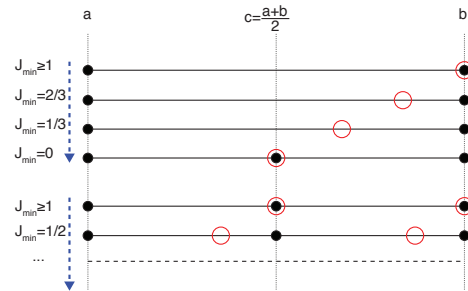


Figure 10: Partitions with additional intermediate steps (red circles), placed according to the value of J_{\min} .

an intermediate step in the binary partition between t_k and t_{k+1} . Figure 8 shows an example of such an artefact. Even though the positions of the upper right target vertex differ only slightly, they lead to different partitions and different composite mappings. If we consider a unit square and linearly move its upper right corner from $(1, 1)$ on the diagonal to the position $(0.005, 0.005)$, J_{\min} changes linearly as shown in Figure 9.

In order to avoid visual artefacts, we introduce an artificial intermediate step \tilde{c} between a and b as shown in Figure 10, where \tilde{c} is indicated by a red circle. The position of \tilde{c} moves smoothly from a to $c = (a+b)/2$ (the position of the new step) according to J_{\min} . Figure 10 shows the movement of \tilde{c} .

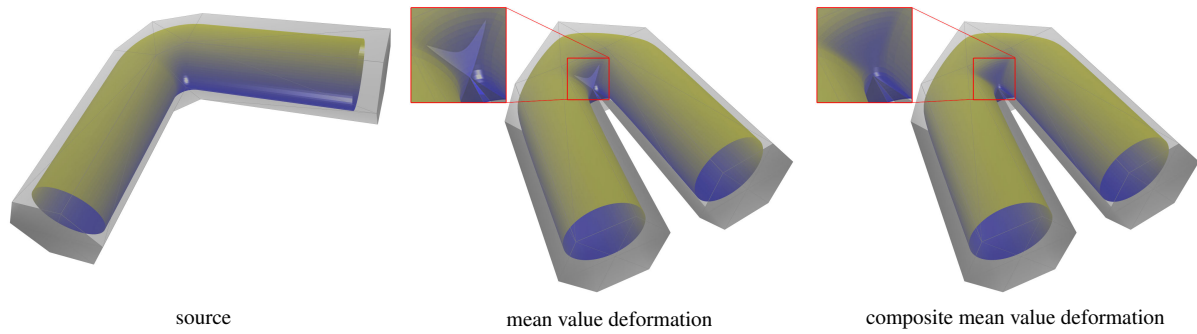


Figure 11: Comparison of shape deformation using a non-bijective mean value mapping and a bijective composite mean value mapping with 100 uniform steps. Note how the deformed shape intersects the target polyhedron in the close-up for the direct mean value deformation.

We see that \tilde{c} moves to c as J_{\min} becomes less positive. When the step c is introduced, two additional steps are created (the two red circles in the second part of the image) which again move according to J_{\min} from their original position to the next intermediate steps.

4.5. Extension to 3D

The concept of composite mean value mappings can be extended easily to 3D to generate mappings between a source and a target polyhedron. It is beyond the scope of this paper to investigate any theoretical guarantees, but our numerical examples show that a sufficiently large number of steps guarantees bijectivity even for extreme deformations.

Figure 11 shows an example, where mean value mappings are used to deform a simple shape that is embedded in the source polyhedron. Even though both deformed shapes are free of self-intersections and are valid meshes in that sense, note that the result of the mean value deformation leaves the target polyhedron (see close-up), which evidences that the mapping is not bijective. In contrast, the composite mean value mapping with 100 uniform steps keeps the shape inside the target polyhedron and we verified numerically that it is indeed bijective. In this example we used the method of Winkler et al. [WDAH10] to generate the interpolated polyhedra, which is not guaranteed to prevent self-intersections of the latter, but works well in practice. Otherwise, the interference algorithm by Harmon et al. [HPSZ11] can be used to avoid self-intersections.

5. Conclusions

Composite mean value mappings provide an efficient tool for designing bijective mappings between arbitrary polygons, which in turn can be used for user-controlled image warping. Barycentric mappings are by no means the only way to warp images and many other approaches exist, based for example on radial basis functions [RM95], B-splines

[LCSW95], moving least squares [SMW06], Killing vector fields [SBCBG11], partial differential equations [JWSH12], or finite elements [KWSH*13]. However, to the best of our knowledge, composite mean value mappings are the only way so far to guarantee a bijective mapping that is piecewise linear along the boundaries of source and target polygon, which might be preferred in certain situations. Moreover, the concept extends straightforwardly to 3D, inheriting all advantages from the 2D setting.

We admit that the guaranteed bijectivity of composite mean value mappings depends on the existence of the upper bound M^* in (6), which is proven only in the convex setting up to now [RGB12]. It remains future work to establish similar bounds for the gradients of mean value coordinates with respect to arbitrary polygons, but all our numerical tests show that such a bound exists and that bijective mappings can be designed by using a sufficiently large number of steps. Actually, this number turns out to be rather small if bijectivity is required only up to pixel accuracy, which in turn enables applications at interactive speed.

Another interesting question for future research regards the following observation. The inverse of a barycentric mapping is usually not a barycentric mapping itself, but it seems as if the inverse of an *infinite mean value mapping* $f_\infty = \lim_{m \rightarrow \infty} f_{\tau_m}$ is again an infinite mean value mapping. Figure 12 shows the result of mapping the star from Figure 4 with a uniform composite mapping f_{τ_m} to the target polygon and back to the original shape by using another uniform composite mapping $\tilde{f}_{\tau_m} : \hat{\Omega} \rightarrow \Omega$ with the same number of steps. Computing the maximum distance $\|v - \tilde{f}_{\tau_m}(f_{\tau_m}(v))\|$ for one million random points $v \in \Omega$ reveals that this distance is on the order of $O(1/m)$, so that $\tilde{f}_{\tau_m} = f_{\tau_m}^{-1}$ as $m \rightarrow \infty$.

Finally, it will be interesting to study the distortion properties of composite mean value mappings and to investigate how the quality of a mapping depends on the particular choices of barycentric coordinates, vertex paths, and partitions.

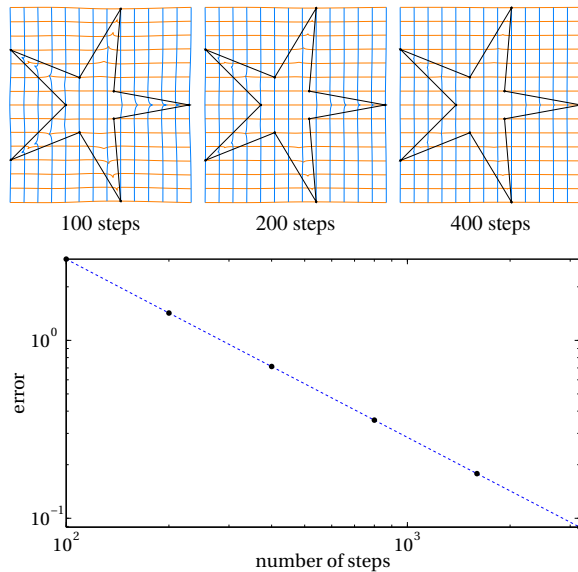


Figure 12: Composing the mapping f_{τ_m} and the backward mapping \bar{f}_{τ_m} converges to the identity as the number of m uniform steps increases.

Acknowledgements

This work was supported by the SNF under project number 200021_140583. We thank the anonymous reviewers for their insightful comments.

References

- [ACOL00] ALEXA M., COHEN-OR D., LEVIN D.: As-rigid-as-possible shape interpolation. In *Proceedings of SIGGRAPH* (New Orleans, July 2000), pp. 157–164. [2](#)
- [EDD*95] ECK M., DERÖSE T., DUCHAMP T., HOPPE H., LOUNSBERRY M., STUETZLE W.: Multiresolution analysis of arbitrary meshes. In *Proceedings of SIGGRAPH* (Los Angeles, Aug. 1995), pp. 173–182. [2](#)
- [FK10] FLOATER M. S., KOSINKA J.: On the injectivity of Wachspress and mean value mappings between convex polygons. *Advances in Computational Mathematics* 32, 2 (Feb. 2010), 163–174. [2, 4](#)
- [FKR05] FLOATER M. S., KÓS G., REIMERS M.: Mean value coordinates in 3D. *Computer Aided Geometric Design* 22, 7 (Oct. 2005), 623–631. [2](#)
- [Flo03] FLOATER M. S.: Mean value coordinates. *Computer Aided Geometric Design* 20, 1 (Mar. 2003), 19–27. [2, 3](#)
- [GD01] GAIN J. E., DODGSON N. A.: Preventing self-intersection under free-form deformation. *IEEE Transactions on Visualization and Computer Graphics* 7, 4 (Oct.–Dec. 2001), 289–298. [2](#)
- [HF06] HORMANN K., FLOATER M. S.: Mean value coordinates for arbitrary planar polygons. *ACM Transactions on Graphics* 25, 4 (Oct. 2006), 1424–1441. [2, 3, 4](#)
- [HPSZ11] HARMON D., PANOZZO D., SORKINE O., ZORIN D.: Interference-aware geometric modeling. *ACM Transactions on Graphics* 30, 6 (Dec. 2011), Article 137, 10 pages. *Proceedings of SIGGRAPH Asia*. [8](#)
- [HS08] HORMANN K., SUKUMAR N.: Maximum entropy coordinates for arbitrary polytopes. *Computer Graphics Forum* 27, 5 (July 2008), 1513–1520. *Proceedings of SGP*. [2](#)
- [Jac12] JACOBSON A.: *Bijective mappings with generalized barycentric coordinates: a counterexample*. Tech. rep., Department of Computer Science, ETH Zurich, 2012. [1](#)
- [JBPS11] JACOBSON A., BARAN I., POPOVIĆ J., SORKINE O.: Bounded biharmonic weights for real-time deformation. *ACM Transactions on Graphics* 30, 4 (July 2011), Article 78, 8 pages. *Proceedings of SIGGRAPH*. [2](#)
- [JMD*07] JOSHI P., MEYER M., DERÖSE T., GREEN B., SANOCKI T.: Harmonic coordinates for character articulation. *ACM Transactions on Graphics* 26, 3 (July 2007), Article 71, 9 pages. *Proceedings of SIGGRAPH*. [2](#)
- [JSW05] JU T., SCHAEFER S., WARREN J.: Mean value coordinates for closed triangular meshes. *ACM Transactions on Graphics* 24, 3 (July 2005), 561–566. *Proceedings of SIGGRAPH*. [2](#)
- [JWSH12] JACOBSON A., WEINKAUF T., SORKINE-HORNUNG O.: Smooth shape-aware functions with controlled extrema. *Computer Graphics Forum* 31, 5 (Aug. 2012), 1577–1586. *Proceedings of SGP*. [8](#)
- [KWSH*13] KAUFMANN P., WANG O., SORKINE-HORNUNG A., SORKINE-HORNUNG O., SMOLIC A., GROSS M.: Finite element image warping. *Computer Graphics Forum* 32, 2 (May 2013), 31–39. *Proceedings of Eurographics*. [8](#)
- [LBS06] LANGER T., BELYAEV A., SEIDEL H.-P.: Spherical barycentric coordinates. In *Proceedings of the Fourth Symposium on Geometry Processing* (Cagliari, June 2006), pp. 81–88. [2](#)
- [LCSW95] LEE S.-Y., CHWA K.-Y., SHIN S. Y., WOLBERG G.: Image metamorphosis using snakes and free-form deformations. In *Proceedings of SIGGRAPH* (Los Angeles, Aug. 1995), pp. 439–448. [8](#)
- [LH13] LI X.-Y., HU S.-M.: Poisson coordinates. *IEEE Transactions on Visualization and Computer Graphics* 19, 2 (Feb. 2013), 344–352. [2](#)
- [Lip12] LIPMAN Y.: Bounded distortion mapping spaces for triangular meshes. *ACM Transactions on Graphics* 31, 4 (July 2012), Article 108, 13 pages. *Proceedings of SIGGRAPH*. [2](#)
- [LKCOL07] LIPMAN Y., KOPF J., COHEN-OR D., LEVIN D.: GPU-assisted positive mean value coordinates for mesh deformations. In *Proceedings of the Fifth Symposium on Geometry Processing* (Barcelona, July 2007), pp. 117–123. [2](#)
- [LLCO08] LIPMAN Y., LEVIN D., COHEN-OR D.: Green coordinates. *ACM Transactions on Graphics* 27, 3 (Aug. 2008), Article 78, 10 pages. *Proceedings of SIGGRAPH*. [2](#)
- [MO63] MEISTERS G. H., OLECH C.: Locally one-to-one mappings and a classical theorem on the schlicht functions. *Duke Mathematical Journal* 30, 1 (Mar. 1963), 63–80. [2](#)
- [MS10] MANSON J., SCHAEFER S.: Moving least squares coordinates. *Computer Graphics Forum* 29, 5 (July 2010), 1517–1524. *Proceedings of SGP*. [2](#)
- [PP93] PINKALL U., POLTHIER K.: Computing discrete minimal surfaces and their conjugates. *Experimental Mathematics* 2, 1 (1993), 15–36. [2](#)
- [RGB12] RAND A., GILLETTE A., BAJAJ C.: Interpolation error estimates for mean value coordinates over convex polygons. *Advances in Computational Mathematics* (2012). To appear. [4, 8](#)

- [RM95] RUPRECHT D., MÜLLER H.: Image warping with scattered data interpolation. *IEEE Computer Graphics and Applications* 15, 2 (Mar. 1995), 37–43. [8](#)
- [SBCBG11] SOLOMON J., BEN-CHEN M., BUTSCHER A., GUIBAS L.: As-killing-as-possible vector fields for planar deformation. *Computer Graphics Forum* 30, 5 (Aug. 2011), 1543–1552. Proceedings of SGP. [8](#)
- [SGWM93] SEDERBERG T. W., GAO P., WANG G., MU H.: 2-D shape blending: an intrinsic solution to the vertex path problem. In *Proceedings of SIGGRAPH* (Anaheim, Aug. 1993), pp. 15–18. [4](#)
- [SMW06] SCHAEFER S., MCPHAIL T., WARREN J.: Image deformation using moving least squares. *ACM Transactions on Graphics* 25, 3 (July 2006), 533–540. Proceedings of SIGGRAPH. [8](#)
- [Wac75] WACHSPRESS E. L.: *A Rational Finite Element Basis*, vol. 114 of *Mathematics in Science and Engineering*. Academic Press, New York, 1975. [2](#)
- [WBCG09] WEBER O., BEN-CHEN M., GOTSMAN C.: Complex barycentric coordinates with applications to planar shape deformation. *Computer Graphics Forum* 28, 2 (Apr. 2009), 587–597. Proceedings of Eurographics. [2](#)
- [WBCGH11] WEBER O., BEN-CHEN M., GOTSMAN C., HORMANN K.: A complex view of barycentric mappings. *Computer Graphics Forum* 30, 5 (Aug. 2011), 1533–1542. Proceedings of SGP. [2](#)
- [WDAH10] WINKLER T., DRIESEBERG J., ALEXA M., HORMANN K.: Multi-scale geometry interpolation. *Computer Graphics Forum* 29, 2 (May 2010), 309–318. Proceedings of Eurographics. [8](#)
- [WG10] WEBER O., GOTSMAN C.: Controllable conformal maps for shape deformation and interpolation. *ACM Transactions on Graphics* 29, 4 (July 2010), Article 78, 11 pages. Proceedings of SIGGRAPH. [2](#)
- [WSD07] WARREN J., SCHAEFER S., HIRANI A. N., DESBRUN M.: Barycentric coordinates for convex sets. *Advances in Computational Mathematics* 27, 3 (Oct. 2007), 319–338. [2](#)