Volume xx (*200y*), *Number z*, *pp. 1–14*

Constrained Modeling of 3-valent Meshes Using a Hyperbolic Deformation Metric

Ronald Richter   Jan Eric Kyprianidis   Boris Springborn   Marc Alexa

TU Berlin

Abstract

Polygon meshes with 3-valent vertices often occur as the frame of free-form surfaces in architecture, in which rigid beams are connected in rigid joints. For modeling such meshes it is desirable to measure the deformation of the joints' shapes. We show that it is natural to represent joint shapes as points in hyperbolic 3-space. This endows the space of joint shapes with a geometric structure that facilitates computation. We use this structure to optimize meshes towards different constraints, and we believe that it will be useful for other applications as well.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Curve, surface, solid, and object representations

1. Introduction

The optimization of meshes for use in architecture has recently received considerable attention. Most of this work considers constraints on the shape of the faces [PLW*07, PSB*08]. For example, asking for planar faces significantly limits the space of possible shapes [CW07, YYPM11, Vax12].

Here we consider the mesh to be a truss of connected beams, i.e. a shell structure representing a surface with faces that may not be planar and may have arbitrary degree. Noting that it is easy to cut the beams to arbitrary length, we focus on constraining the shape of the joints. Reducing the number of edges to save weight and material leads to smaller vertex degree or larger face degree [LLW15]. Consequently, we limit the construction to use 3-valent joints, and then we consider different constraints on the *shape* of the joint. The main motivation for constraining the joints is that manufacturing individual joints for each node is expensive. Like constraining face elements in architecture, it is more economical to restrict the joint shapes.

Figure 1 shows several existing types of joints: there are standard constructions such as T-joints or 90 degree elbows or more flexible space frame nodes, which are often derived from regular polyhedra. Modifying the mesh so that all nodes can be built from such existing joints could be one strategy, albeit quite restrictive. We note that all common joints are manufactured by molding – and it would be easy to generate other forms of 3-way joints by using molds derived from the construction. This means the shape of the joints would be optimized with the frame being given — an approach that is the dual of [EKS*10, FLHCO10, SS10].

We also believe it is easier to connect the beams if the joint is *symmetric*, i.e., all incident beams subtend the same angle. This property can be exploited to simplify the manufacturing process of joints. Moreover, meshes of symmetric joints found interest in architectural geometry in the form of "honeycomb structures" [**?**]. Consequently, we would like to optimize the mesh such that all joint shapes have this property.

While our approach is related and motivated by the mentioned works that optimize the faces of the mesh, there are interesting fundamental differences: limiting the shape of the faces also fixes the scale, while the shape of the joints is independent of the lengths of the beams, thus leaving more flexibility for the possible constructions. Moreover, the constraints on faces are often invariant with respect to to reflections (i.e., faces are required to fall into congruence classes), while for non-planar joints we are necessarily interested in the shape, meaning we allow only invariance with respect to rigid motions. This has implications on the representation of joint shapes which we will discuss in Section 2.

Our work is based on the observation that Möbius trans-

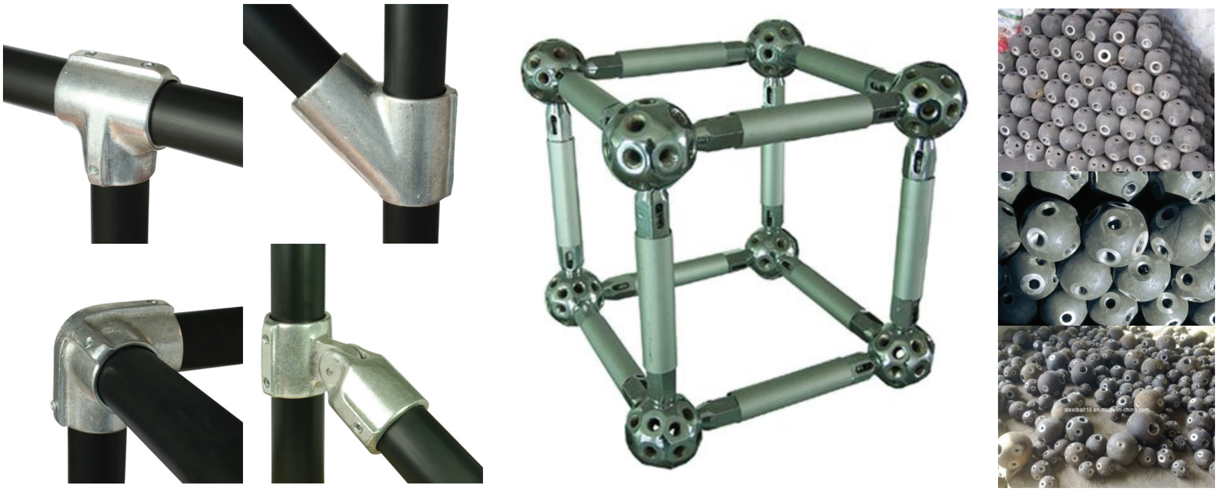

Figure 1: *Real joints*

formations succinctly describe the transformation between joint configurations (see Section 3). Since we are interested in the *deformation*, i.e., in the difference in intrinsic joint angles, but not the rigid transformation between joints, we factor out the rotation from the Möbius transformation. The remaining factor is a Hermitian 2×2-matrix, and these form a model of hyperbolic 3-space. In other words, a rotation-invariant representation of joint shapes is given by points in hyperbolic space. Among other useful properties, this induces a natural metric on the space of joint shapes that allows us to measure the deformation between them (see Section 4).

The hyperbolic 3-space representation of trivalent joints is not just the 'right' description from a purely mathematical point of view, it is also computationally convenient and yields closed-form expressions for all the calculations we need for optimization (Section 6), such as computing means of joint shapes or projecting onto the closest symmetric configuration. By using hyperbolic geometry and complex matrices, the resulting computations are not only elegant, but simple, direct, and fast.

This representation enables us to compute optimized meshes by minimizing distances in hyperbolic space (Section 7). We show several examples of optimized meshes and eventually discuss the inherent limitations of the approach.

2. Shape space of joints: introductory remarks

The shape of a 3-valent joint can be specified by three distinct unit vectors $\mathbf{e}_0, \mathbf{e}_1, \mathbf{e}_2$ pointing in the directions of the outgoing edges, which we may consider as points on the unit sphere:

$$\mathbf{e}_k = \begin{pmatrix} e_{x,k} \\ e_{y,k} \\ e_{z,k} \end{pmatrix} \in S^2 \subset \mathbb{R}^3. \tag{1}$$

This describes, however, not only the shape of the joint but also its orientation (or attitude) in space.

A central question in our work is how to represent the joint without considering its orientation. This representation would allow us to identify when two (or more) joints have the same shape. Moreover, we might use the representation to endow the space of joints with a proper metric, which would facilitate computations such as the *mean* of several joints.

In the following sections we show that a clean way to do this is by considering the relation of two joints as Möbius transformations and then identifying the deformation part of this transformation with a point in hyperbolic space. Choosing an arbitrary joint configuration as "origin" makes shapes of joints correspond to points in hyperbolic space and distance can be measured in this space. (The choice of a reference joint is as irrelevant for this representation as the choice of an origin and a coordinate frame in Euclidean geometry.) It may seem that this machinery is more involved than necessary for practical purposes. For this reason, we briefly discuss two ideas that may come to mind for the purpose of representing the shape of joints or their deformation – and why they fail.

2.1. Angles – spherical geometry

It seems the shape of the joint could be described by the angles between the edge directions, i.e., by $\phi_k = \arccos(\mathbf{e}_k^\top \mathbf{e}_{k+1})$. The arccos-function requires deciding on a

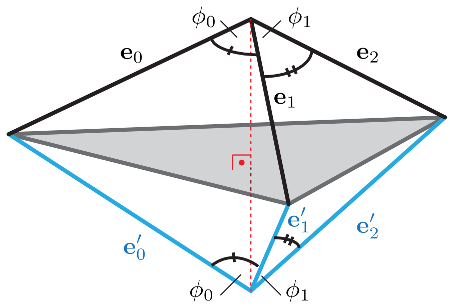

Figure 2: *Mirrored joints* $(\mathbf{e}_0, \mathbf{e}_1, \mathbf{e}_2)$ *and* $(\mathbf{e}'_0, \mathbf{e}'_1, \mathbf{e}'_2)$ *are identified with each other when represented by the corresponding angles* (ϕ_0, ϕ_1, ϕ_2) *between edges*

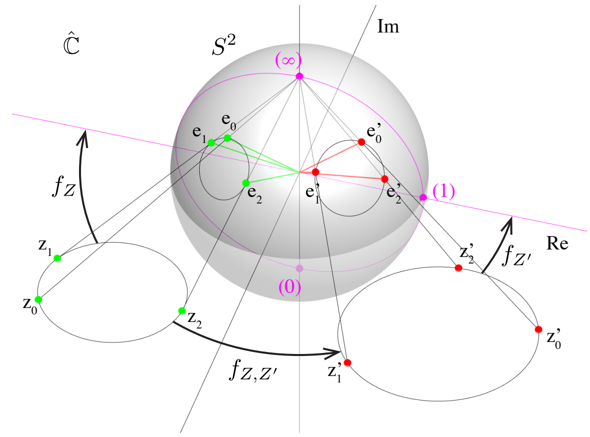

Figure 3: *Illustration of two joints* $(\mathbf{e}_0, \mathbf{e}_1, \mathbf{e}_2)$ *and* $(\mathbf{e}'_0, \mathbf{e}'_1, \mathbf{e}'_2)$ *on* S^2 *projected stereographically on the extended complex plane* $\hat{\mathbb{C}}$ *to* Z *and* Z'*. The Möbius transformations* f_Z *and* $f_{Z'}$ *map the triples* Z *and* Z' *to* $(0, 1, \infty)$ *which is a line on the real axis. The composition* $f_{Z,Z'} = f_{Z'}^{-1} \circ f_Z$ *then directly maps* Z *to* Z'*.*

preferred interval and it seems natural to ask that $\phi_k \in [0, \pi]$. Under this assumption, considering the angles is equivalent to represent the joint shape using the *side lengths* of the *spherical triangle* spanned by the points $\mathbf{e}_k$ on the unit sphere.

The fundamental problem of this representation is that it identifies mirrored joints with each other: a triangle is described by its edge lengths *up to congruence*, which includes reflections (see Figure 2). Reflections, however, are not rigid motions so they represent different joint shapes.

Another potential problem of this representation is that not every triple (ϕ_0, ϕ_1, ϕ_2) identifies a joint shape: Apart from $0 \le \phi_k \le \pi$, the sum of the angles is also bounded from above by the planar configuration $\sum_k \phi_k \le 2\pi$; and side lengths must obey the triangle inequality $\phi_k \le \phi_{k+1} + \phi_{k+2}$. So optimizing an angle-based representation would always have to be subject to these linear constraints.

2.2. Bases – Euclidean geometry

As angles are invariant under reflections it may seem natural to consider the unit vectors as a basis $(\mathbf{e}_0, \mathbf{e}_1, \mathbf{e}_2)$. Since we are interested in measuring the difference of two joint shapes, one could compute the linear transformation between the bases

$$(\mathbf{e}_0, \mathbf{e}_1, \mathbf{e}_2) = \mathbf{A}(\mathbf{e}_0, \mathbf{e}_1, \mathbf{e}_2) \tag{2}$$

and then factor out the special orthogonal part using the polar decomposition, i. e., $\mathbf{A} = \mathbf{RS}$. The deformation would then be captured in $\mathbf{S}$ and could be measured by comparing $\mathbf{S}$ to the identity.

This approach fails for planar joints, for which $(\mathbf{e}_0, \mathbf{e}_1, \mathbf{e}_2)$ is not a basis. In particular, note that cases such as T-joints of the form $\mathbf{e}_0 = -\mathbf{e}_1$ could not be handled.

3. Joint configurations are related by Möbius transformations

In this section, we will explain how Möbius transformations can be employed to describe the difference n shape and orientation between two joints. In particular, we will discuss how to represent an arbitrary joint with respect to a reference configuration. We will also review basic facts about these transformations. Afterwards, we will explain how the desired description of pure shape differences can be extracted from the Möbius description of combined shape/orientation differences.

By stereographic projection, we can conformally identify the unit sphere S^2 with the extended complex plane $\hat{\mathbb{C}} = \mathbb{C} \cup \{\infty\}$. Given a point on the unit sphere the mapping is given by

$$S^2 \ni (x, y, z) \mapsto \frac{x + iy}{1 - z} \in \hat{\mathbb{C}}. \tag{3}$$

Hence, a 3-valent joint, given by three distinct unit vectors $\mathbf{e}_0, \mathbf{e}_1, \mathbf{e}_2$, may be represented as a triple $Z = (z_0, z_1, z_2)$ of distinct points in the extended complex plane. This representation is the basis for all subsequent developments.

A Möbius transformation is an invertible mapping $f : \hat{\mathbb{C}} \to \hat{\mathbb{C}}$ of the extended complex plane $\hat{\mathbb{C}}$ to itself given by

$$f(z) = \frac{az + b}{cz + d}, \quad a, b, c, d \in \mathbb{C}, \quad ad - bc \neq 0. \tag{4}$$

These transformations are important for us because for any triple of distinct points, $Z = (z_0, z_1, z_2)$, there exists a unique

Möbius transformation mapping it to $(0,1,\infty)$, namely

$$f_Z(z) = \frac{z_1 - z_2}{z_1 - z_0}\,\frac{z - z_0}{z - z_2}. \tag{5}$$

Hence, the composition

$$f_{Z,Z'} = f_{Z'}^{-1} \circ f_Z. \tag{6}$$

is the unique Möbius transformation mapping the triple Z to an arbitrary other triple $Z' = (z'_0, z'_1, z'_2)$ (see Figure 3). We hence have:

- If triples Z and Z' represent the shapes and orientations of two joints by Equation (3), then the Möbius transformation $f_{Z,Z'}$ represents the difference of their shapes and orientations.
- With a fixed joint shape and orientation as reference, Möbius transformations can be used to represent the shape and orientation of any other joint: If the reference shape/orientation is given by Z, then a Möbius transformation f represents the shape/orientation given by $f(Z) = (f(z_0), f(z_1), f(z_2))$.

In the following it will be useful to write Möbius transformations, and by the above identification also joints, as complex 2×2-matrices. This is analogous to how real 4×4-matrices are used in computer graphics to describe affine and projective transformations. To this end, points in the extended complex plane $\hat{\mathbb{C}}$ are specified using homogeneous coordinate vectors in $\mathbb{C}^2$:

$$\mathbb{C} \ni z \to \begin{pmatrix} z \\ 1 \end{pmatrix} = \mathbf{z} \in \mathbb{C}^2, \quad \infty \to \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \mathbf{z}, \tag{7}$$

with two homogeneous coordinate vectors $\mathbf{z}, \mathbf{z}' \in \mathbb{C}^2$ representing the same point in $\hat{\mathbb{C}}$ if they are complex multiples of each other:

$$\mathbf{z} \equiv \mathbf{z}' \iff \exists \lambda \in \mathbb{C} \setminus \{0\} : \; \mathbf{z} = \lambda \mathbf{z}'. \tag{8}$$

Using this homogeneous representation, the Möbius transformation in Equation (4) corresponds to multiplication $\mathbf{z} \mapsto \mathbf{M}\mathbf{z}$ with the complex 2×2-matrix

$$\mathbf{M} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}. \tag{9}$$

Since this correspondence is only unique up to a non-zero scalar factor, we normalize the matrix by requiring the determinant $|\mathbf{M}|$ to satisfy

$$|\mathbf{M}| = ad - bc = 1. \tag{10}$$

This makes the matrix representation unique up to sign: $\mathbf{M}$ and $-\mathbf{M}$ still represent the same Möbius transformation. Möbius transformations are thus represented by by elements of the group $SL(2,\mathbb{C})$ formed by complex 2×2-matrices with unit determinant. This description is convenient because a composition of Möbius transformations is represented by the product of the corresponding matrices and the inverse of a transformation is represented by the inverse of the matrix.

The normalized matrix representations of the Möbius transformations f_Z and $f_{Z,Z'}$ defined in Equations (5) and (6) are important for our calculations. These are

$$\mathbf{M}_Z = \frac{1}{\sqrt{\Delta z}} \begin{pmatrix} z_1 - z_2 & -z_0(z_1 - z_2) \\ z_1 - z_0 & -z_2(z_1 - z_0) \end{pmatrix}, \tag{11}$$

with $\Delta z = (z_0 - z_1)(z_1 - z_2)(z_2 - z_0)$ being the determinant of the matrix on the right-hand side of the above expression, and

$$\mathbf{M}_{Z,Z'} = \mathbf{M}_{Z'}^{-1} \mathbf{M}_Z. \tag{12}$$

With the previous results that relate joints and their relative differences to Möbius transformations we can now summarize:

- The matrix $\mathbf{M}_{Z,Z'} \in SL(2,\mathbb{C})$ represents the difference of the shapes and orientations of two joints represented by Z and Z'. (The matrix $-\mathbf{M}_{Z,Z'}$ represents the same difference.)
- With a fixed joint shape and orientation Z' as reference, the matrix $\mathbf{M}_{Z,Z'} \in SL(2,\mathbb{C})$ represents the shape and orientation of a joint Z. (The matrix $-\mathbf{M}_{Z,Z'}$ represents the same shape and orientation.)

Note that we represent a joint configuration Z by the Möbius transformation that maps it to the reference configuration Z'. We do not use the inverse transformation, which maps the reference configuration Z' to Z. At this point, this is just a matter of convention. But it will make a difference in the following section, when we represent shapes and not configurations of joints.

Also note that we have written Equations (3), (4), (5), and (11) without using complex homogeneous coordinates for points in $\hat{\mathbb{C}}$. Hence, they have to be modified if ∞ occurs (or augmented with rules of “canceling infinities”). If homogeneous coordinates are used throughout, one obtains equations that do not require any special treatment of ∞.

4. Joint shape space is hyperbolic 3-space

As we have seen in the previous section, the transformation between two joints can be described by a Möbius transformation. We are interested in measuring how much *deformation* is present in such a transformation. That is, we would like to ignore the rotational part of the mapping and have a metric that only depends on the shape of the joint given by the relative directions of the outgoing edges.

Our construction is based on the fact that Möbius transformations also represent *orientation preserving isometries* of *hyperbolic 3-space* H^3, i.e., rigid motions in hyperbolic space. This leads to a correspondence between the shape space of joints and hyperbolic 3-space. In the following, we will explain the details of this approach.

4.1. Matrix model of hyperbolic 3-space

In the *hyperboloid model* of hyperbolic geometry, the hyperbolic 3-space H^3 is given by

$$H^3 = \{\mathbf{h} \in \mathbb{R}^4 \mid -h_0^2 + h_1^2 + h_2^2 + h_3^2 = -1,\ h_0 > 0\}. \quad (13)$$

Distances in H^3 are measured with the hyperbolic metric d_H, which is defined in terms of the Minkowski inner product $\langle\cdot,\cdot\rangle_{3,1}$ as

$$\cosh d_H(\mathbf{h},\mathbf{h}') = -\langle\mathbf{h},\mathbf{h}'\rangle_{3,1} = h_0 h_0' - \sum_{k=1}^{3} h_k h_k'. \quad (14)$$

Equation (14) is analogous to how distances are measured in spherical geometry, with the usual Euclidean inner product replaced by the Minkowski inner product and the cosine replaced with the hyperbolic cosine.

The most well known models of hyperbolic space are the the *Cayley–Klein model*, the *Poincaré ball model*, and the *Poincaré half-space model*. In the Cayley–Klein model, hyperbolic 3-space is represented by the open unit ball in $\mathbb{R}^3$, and hyperbolic lines and planes are represented by intersections of Euclidean lines and planes with the unit ball. The Poincaré ball model also represents hyperbolic space as the unit ball, but hyperbolic lines and planes are represented by circular arcs or diameters, and by spherical caps or equatorial disks, respectively, all meeting the unit sphere orthogonally. The Poincaré half-space model represents hyperbolic space as the upper half-space of $\mathbb{R}^3$, and hyperbolic lines and planes correspond to half-circles or vertical lines, and to hemispheres or vertical half-planes, respectively. The Poincaré ball and half-space models are conformal, i.e., hyperbolic angles are represented correctly. A point $\mathbf{h} \in H^3$ of the hyperboloid model corresponds to the point $\frac{1}{h_0}(h_1,h_2,h_3)$ in the Cayley–Klein model, to the point $\frac{1}{h_0+1}(h_1,h_2,h_3)$ in the Poincaré ball model, and to the point $\frac{1}{2(h_0+h_3)}(h_1,h_2,1+h_0+h_3)$ in the half-space model.

For our purposes, a less well known model of hyperbolic 3-space is more convenient. One can identify elements $\mathbf{h} = (h_0,h_1,h_2,h_3) \in H^3$ in the hyperboloid model with positive definite Hermitian 2×2-matrices with unit determinant. In components, the corresponding matrix $\mathbf{H}$ is given by

$$\mathbf{H} = \begin{pmatrix} h_0+h_3 & h_1+ih_2 \\ h_1-ih_2 & h_0-h_3 \end{pmatrix}. \quad (15)$$

This identification is known as the *matrix model* of hyperbolic 3-space. Note how the unit determinant property is equivalent to the hyperboloid equation:

$$\begin{aligned} |\mathbf{H}| &= (h_0+h_3)(h_0-h_3) - (h_1+ih_2)(h_1-ih_2) \\ &= h_0^2 - h_3^2 - h_1^2 - h_2^2 = -\langle\mathbf{h},\mathbf{h}\rangle_{3,1} = 1\,. \end{aligned} \quad (16)$$

Moreover, it follows that the determinant in the matrix model is equivalent to the quadratic form induced by the Minkowski inner product in the hyperboloid model. This enables computing the Minkowski inner product from the determinant using the polarization identity. Let $\mathbf{H} \cong \mathbf{h}$ and $\mathbf{H}' \cong \mathbf{h}'$ be Hermitian matrices with their corresponding points in the hyperboloid model, then the polarization identity yields:

$$\begin{aligned} \langle\mathbf{h},\mathbf{h}'\rangle_{3,1} &= \tfrac{1}{2}(|\mathbf{H}+\mathbf{H}'| - |\mathbf{H}| - |\mathbf{H}'|) \\ &= \tfrac{1}{2}|\mathbf{H}+\mathbf{H}'| - 1\,. \end{aligned} \quad (17)$$

Furthermore, this shows that linear transformations of $\mathbf{H}$ that preserve the determinant of Hermitian matrices are isometries, because they preserve the Minkowski inner product.

Now, given any Möbius transformation represented by the matrix $\mathbf{M}$, it easy to see that the map

$$\mathbf{H} \mapsto \mathbf{M}\mathbf{H}\mathbf{M}^* \quad (18)$$

induces an isometry on H^3: the map sends Hermitian matrices to Hermitian matrices and, because $|\mathbf{M}| = 1$, the determinant and thus the Minkowski inner product are preserved.

In fact, one can show that the isometry (18) is orientation preserving, and that every orientation preserving isometry of H^3 can be represented in this way by an $SL(2,\mathbb{C})$-matrix. As with Möbius transformations, the representation is two-to-one: $\mathbf{M}$ and $-\mathbf{M}$ represent the same rigid motion in hyperbolic space. Thus, there is a one-to-one correspondence between Möbius transformations and hyperbolic rigid motions.

As the 2-sphere is the infinite boundary of hyperbolic 3-space, the action of $SL(2,\mathbb{C})$ on H^3 also provides a shortcut to calculate the action of an $SL(2,\mathbb{C})$-matrix M on the unit sphere. Instead of projecting a point $\mathbf{e} \in S^2$ stereographically to a point $z \in \hat{\mathbb{C}}$, applying the Möbius transformation $f(z)$ with matrix M, and then projecting back to S^2, one can achieve the same result as follows: In homogeneous coordinates (x_0,x_1,x_2,x_3), the equation for S^2 is $-x_0^2 + \sum_1^3 x_k = 0$, so a point in S^2 is homogeneously represented by a Hermitian matrix H with determinant zero. The image point is MHM^*.

4.2. Rotations are represented by special unitary matrices

If we identify the sphere S^2 with the extended complex plane $\hat{\mathbb{C}}$ via stereographic projection (3), then the rotations of S^2 correspond to the Möbius transformations with special unitary matrices $\mathbf{U} \in SU(2)$, i.e., matrices of the form

$$\mathbf{U} = \begin{pmatrix} \alpha & -\bar{\beta} \\ \beta & \bar{\alpha} \end{pmatrix}, \quad \alpha,\beta \in \mathbb{C}, \quad |\alpha|^2 + |\beta|^2 = 1, \quad (19)$$

which satisfy $\mathbf{U}\mathbf{U}^* = \mathbf{I}$. This can be seen directly from (3): First, note that a ϕ-rotation around the z-axis corresponds to the Möbius transformation $w \mapsto e^{i\phi}w$ with normalized matrix

$$\mathbf{U}_3(\tfrac{\phi}{2}) = \begin{pmatrix} \exp(\frac{i\phi}{2}) & 0 \\ 0 & \exp(-\frac{i\phi}{2}) \end{pmatrix}.$$

A little calculation using $x^2+y^2+z^2=1$ shows that a $\frac{\pi}{2}$-rotation around the y-axis corresponds to the Möbius transformation $w \mapsto \frac{w-1}{w+1}$ with normalized matrix

$$\mathbf{U}_2(\tfrac{\pi}{4}) = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}.$$

Hence, a θ-rotation around the x-axis corresponds to the Möbius transformation with matrix

$$\mathbf{U}_1(\tfrac{\theta}{2}) = \mathbf{U}_2(\tfrac{\pi}{4})\mathbf{U}_3(\tfrac{\theta}{2})\mathbf{U}_2(\tfrac{\pi}{4})^{-1} = \begin{pmatrix} \cos\frac{\theta}{2} & i\sin\frac{\theta}{2} \\ i\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix}.$$

Finally, note that every special unitary matrix (19) can be represented as a product of the form

$$\mathbf{U}_3(\tfrac{\phi}{2})\mathbf{U}_1(\tfrac{\theta}{2})\mathbf{U}_3(\tfrac{\psi}{2}) = \begin{pmatrix} e^{\frac{i}{2}(\phi+\psi)}\cos\frac{\theta}{2} & ie^{\frac{i}{2}(\phi-\psi)}\sin\frac{\theta}{2} \\ ie^{\frac{i}{2}(-\phi+\psi)}\sin\frac{\theta}{2} & e^{-\frac{i}{2}(\phi+\psi)}\cos\frac{\theta}{2} \end{pmatrix}.$$

This corresponds to the representation of an arbitrary rotation in terms of the Euler angles (ϕ,θ,ψ).

This $SU(2)$-representation of rotations is closely related to the better known quaternionic representation: in the matrix representation of the quaternions,

$$\mathbf{i} = \begin{pmatrix} 0 & i \\ i & 0 \end{pmatrix}, \quad \mathbf{j} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}, \quad \mathbf{k} = \begin{pmatrix} i & 0 \\ 0 & -i \end{pmatrix}, \tag{20}$$

the unit quaternions correspond to $SU(2)$-matrices.

4.3. From joint shapes to points in hyperbolic 3-space (and back)

We are now ready to explain the representation of joint shapes as points in hyperbolic 3-space. As explained in Section 3, we choose an arbitrary but fixed joint configuration Z' as reference, and we represent any joint *configuration* Z by the $SL(2,\mathbb{C})$-matrix $M_{Z,Z'}$. The *shape* of the joint Z is then represented by the positive definite unimodular Hermitian matrix

$$\mathbf{H}_{Z,Z'} = \mathbf{M}_{Z,Z'}\mathbf{M}^*_{Z,Z'}, \tag{21}$$

i.e., by a point $\mathbf{h}_{Z,Z'} \cong \mathbf{M}_{Z,Z'}$ in hyperbolic 3-space H^3.

To see that the point $\mathbf{h}_{Z,Z'}$ really represents the shape, consider joint configurations Z_1, Z_2 that differ only by a rotation. Then

$$\mathbf{M}_{Z_2,Z'} = \mathbf{M}_{Z_1,Z'}\mathbf{M}_{Z_2,Z_1},$$

where $\mathbf{M}_{Z_2,Z_1} \in SU(2)$, as explained in Section 4.2. Now $\mathbf{M}_{Z_2,Z_1}\mathbf{M}^*_{Z_2,Z_1} = \mathbf{I}$ implies $\mathbf{H}_{Z_2,Z'} = \mathbf{H}_{Z_1,Z'}$. Thus, configurations of joints with the same shape correspond to the same point in hyperbolic space.

Conversely, any point in hyperbolic 3-space determines a unique joint shape, i.e., the point determines a joint configuration up to rotation. This is easily seen using polar decomposition: Given a positive definite Hermitian 2×2-matrix $\mathbf{H}$ with unit determinant, we are interested in finding $SL(2,\mathbb{C})$-matrices $\mathbf{M}$ satisfying

$$\mathbf{H} = \mathbf{M}\mathbf{M}^*. \tag{22}$$

Since any $SL(2,\mathbb{C})$-matrix $\mathbf{M}$ has a unique polar decomposition $\mathbf{M}=\mathbf{P}\mathbf{U}$, where $\mathbf{P}$ is a positive definite Hermitian matrix with unit determinant and $\mathbf{U}\in SU(2)$, and since $\mathbf{M}\mathbf{M}^* = \mathbf{P}^2$, the solutions of (22) are precisely the matrices

$$\mathbf{M} = \mathbf{H}^{\frac{1}{2}}\mathbf{U}, \tag{23}$$

where $\mathbf{U}$ is an arbitrary special unitary matrix. The solutions of (22) therefore represent a class of joint configurations that are related by rotations, i.e., they represent a unique joint shape.

If we are just interested in finding some joint configuration corresponding to a given point in hyperbolic space, we may as well choose $\mathbf{U}=\mathbf{I}$ in (23), i.e., $\mathbf{M}=\mathbf{H}^{\frac{1}{2}}$. Taking the square root is particularly easy in this case: Elementary computations show that for any 2×2 matrix $\mathbf{X}$ we have that $(\mathbf{X}+|\mathbf{X}|\mathbf{I})^2$ is a scalar multiple of $\mathbf{X}$. Since $\mathbf{H}$ has unit determinant by construction, we can compute the desired Möbius transformation $\mathbf{M}$ from $\mathbf{H}$ as

$$\mathbf{M} = \frac{1}{\sqrt{|\mathbf{H}+\mathbf{I}|}}(\mathbf{H}+\mathbf{I}). \tag{24}$$

4.4. Distance in shape space, invariance with respect to the choice of reference configuration

Since we represent joint shapes as points in hyperbolic space, it is a natural idea to measure the distance between the shapes of two joints Z_1, Z_2 by the hyperbolic distance between the corresponding points in H^3:

$$d(Z_1,Z_2) = \cosh^{-1}\left(-\langle \mathbf{h}_{Z_1,Z'},\mathbf{h}_{Z_2,Z'}\rangle_{3,1}\right). \tag{25}$$

At first sight, it may seem that this shape space metric depends on the choice of reference configuration Z', but in fact it does not. If we use a different reference configuration Z'' instead of Z', the shape of a joint Z is represented by $\mathbf{H}_{Z,Z''}$ instead of $\mathbf{H}_{Z,Z'}$ (see (21)). Since $\mathbf{M}_{Z,Z''} = \mathbf{M}_{Z',Z''}\mathbf{M}_{Z,Z'}$, the representations are related by

$$\mathbf{H}_{Z,Z''} = \mathbf{M}_{Z',Z''}\mathbf{H}_{Z,Z'}\mathbf{M}^*_{Z',Z''}.$$

But the map $\mathbf{H} \mapsto \mathbf{M}_{Z',Z''}\mathbf{H}\mathbf{M}^*_{Z',Z''}$ is an isometry of hyperbolic space (see (18)). Thus, a different choice of reference frame changes the representation by a hyperbolic isometry. Distances remain the same. Moreover, angles are also preserved, as well as are straight lines and planes. This will be important when we consider the spaces of planar and symmetric joints.

Since the hyperbolic measure of distance between the shapes of two joints Z, Z' does not depend on the choice of reference configuration, we may as well choose one of the joints, say Z', as reference. Then the configuration Z' is represented by the identity matrix $\mathbf{I}$, so the shape is represented

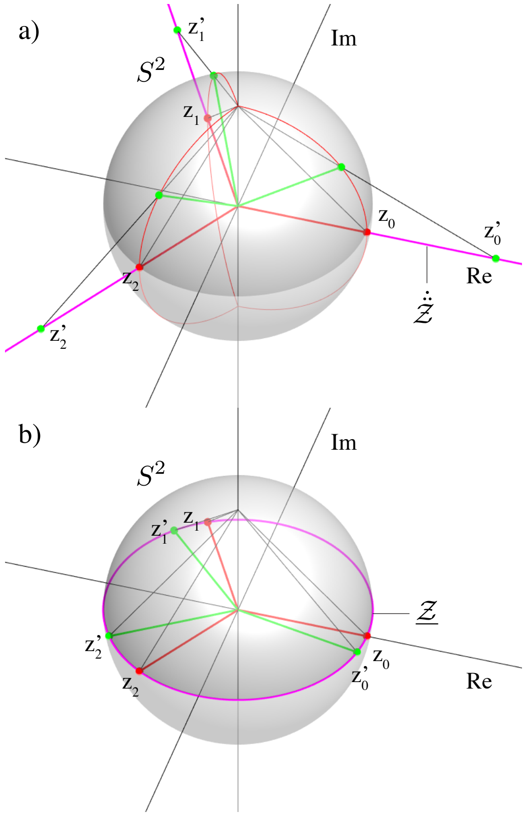

Figure 4: *Two subspaces of joint configurations in the stereographic projection (magenta): a) the subspace of symmetric configurations $\overset{\therefore}{\mathcal{Z}}$ and b) the subspace of flat configurations $\underline{\mathcal{Z}}$. The reference frame $\overset{\therefore}{\underline{Z}} = (z_0, z_1, z_2)$ (red) is by definition flat and symmetric. An instance (z'_0, z'_1, z'_2) of each subspace is shown in green.*

by $(1,0,0,0) \in H^3$, and the distance between the shapes of Z and Z' is

$$\begin{aligned} d(Z,Z') &= \cosh^{-1}\left(-\langle(1,0,0,0),\mathbf{h}_{Z,Z'}\rangle_{3,1}\right) \\ &= \cosh^{-1} h_0. \end{aligned} \tag{26}$$

5. The subspaces of symmetric and flat joint shapes

As mentioned in the introduction, we believe it is useful to constrain the joint shapes to some specific set of shapes. For example we might want all joints to be *threefold rotationally symmetric* (three equal angles between edges). Up to rotation, we may denote the set of symmetric joint shapes as

$$\overset{\therefore}{\mathcal{Z}} = \{(z_0, z_1, z_2), \quad \bar{z}_0 z_1 = \bar{z}_1 z_2 = \bar{z}_2 z_0\}. \tag{27}$$

Or, we may want to ask that all joint shapes are *flat*, i. e., all edges lie in a common plane. We denote the set of flat joint shapes as

$$\underline{\mathcal{Z}} = \{(z_0, z_1, z_2), \quad \bar{z}_0 z_0 = \bar{z}_1 z_1 = \bar{z}_2 z_2 = 1\}. \tag{28}$$

We wish to characterize such joint shapes in a way that allows us to compute the distance of an arbitrary joint shape to the closest symmetric or flat one. In other words, we are interested in the subspaces of symmetric and flat joints; and an orthogonal projection onto these subspaces.

As it turns out, both spaces can be described succinctly relative to the *flat symmetric* reference joint

$$\overset{\therefore}{\underline{Z}} = \left(-1, \frac{1+\sqrt{3}i}{2}, \frac{1-\sqrt{3}i}{2}\right). \tag{29}$$

In Figure 4 the reference joint $\overset{\therefore}{\underline{Z}}$ is shown in red color and the subspaces in magenta. The subspace of symmetric joints $\overset{\therefore}{\mathcal{Z}}$ is a linear subspace as illustrated by lines in $\hat{\mathbb{C}}$ (4a). The unit circle in 4b marks the subspace of all flat configurations.

Effectively, the flat symmetric reference frame lends a geometric meaning to the coefficients of $\mathbf{h}$: as we show next, h_3 can be identified with the non-flat symmetric configurations, while h_1, h_2 span the subspace of flat configurations.

5.1. Symmetric configurations

Equation (27) implies that up to rotations, any element of $\overset{\therefore}{\mathcal{Z}}$ is an isotropic scale of the flat symmetric configuration $\overset{\therefore}{\underline{Z}}$ or, in other words, up to rotation each element of $\overset{\therefore}{\mathcal{Z}}$ is the result of applying a Möbius transformation of the form

$$\mathbf{M}_s = \begin{pmatrix} s & 0 \\ 0 & s^{-1} \end{pmatrix} \tag{30}$$

to $\overset{\therefore}{\underline{Z}}$. Now let us represent a joint configuration Z relative to $\overset{\therefore}{\underline{Z}}$, i.e., by $\mathbf{M}^{-1}_{\overset{\therefore}{\underline{Z}}}\mathbf{M}_Z$. If $\overset{\therefore}{Z} \in \overset{\therefore}{\mathcal{Z}}$ is symmetric, then this transformation consists of a rotation (i.e., the unitary factor) and the isotropic scale shown above, which necessarily corresponds to $\mathbf{H}$. Thus, for symmetric configurations $\overset{\therefore}{Z} \in \overset{\therefore}{\mathcal{Z}}$ we have

$$\mathbf{h}_{\overset{\therefore}{Z},\overset{\therefore}{\underline{Z}}} = (h_0, 0, 0, h_3) \in H^3, \quad \overset{\therefore}{Z} \in \overset{\therefore}{\mathcal{Z}}. \tag{31}$$

Hence, by Equation (25) the inner product between an arbitrary Möbius transformation $H^3 \ni \mathbf{h} \cong Z$ to $\overset{\therefore}{\underline{Z}}$ and a symmetric $\overset{\therefore}{\mathbf{h}} \cong \overset{\therefore}{Z} \in \overset{\therefore}{\mathcal{Z}}$ is given by

$$\cosh\left(d(Z, \overset{\therefore}{Z})\right) = -\langle \mathbf{h}, \overset{\therefore}{\mathbf{h}} \rangle_{3,1} = \overset{\therefore}{h}_0 h_0 - \overset{\therefore}{h}_3 h_3. \tag{32}$$

Thus, to find the symmetric joint closest to Z, we need to vary h_0, h_3 so that this inner product is minimal, subject to the hyperboloid condition $h_0^2 - h_3^2 = 1$. Elementary computations show that the solution is to set $h_1 = h_2 = 0$ and to rescale h_0, h_3 to satisfy the hyperboloid condition, i.e.,

$$\underset{\overset{\therefore}{\mathbf{h}} = (\overset{\therefore}{h}_0, 0, 0, \overset{\therefore}{h}_3) \in H^3}{\arg\min} \quad d(\mathbf{h}, \overset{\therefore}{\mathbf{h}}) = (h_0^2 - h_3^2)^{-1/2}(h_0, 0, 0, h_3). \tag{33}$$

The distance from an arbitrary $Z \cong \mathbf{h} \in H^3$ to the closest symmetric configuration $\overset{\therefore}{\mathbf{h}} \in H^3$ is therefore given by

$$d(\mathbf{h}, \overset{\therefore}{\mathbf{h}}) = \cosh^{-1}\left(\sqrt{h_0^2 - h_3^2}\right). \tag{34}$$

Note that this distance is the same for any cyclic permutation.

5.2. Flat configurations

The reasoning for flat configurations is similar to the situation for symmetric configurations, we therefore omit redundant details. The symmetric flat configuration $\dot{\underline{\mathcal{Z}}}$ is taken to any other flat configuration, up to rotation, by Möbius transformations of the form

$$\mathbf{M}_f = \begin{pmatrix} a & b \\ \bar{b} & a \end{pmatrix} = \begin{pmatrix} h_0 & h_1 + ih_2 \\ h_1 - ih_2 & h_0 \end{pmatrix}. \tag{35}$$

This means, flat configurations $\underline{Z} \in \underline{\mathcal{Z}}$ can be are represented (relative to the symmetric flat configuration) as

$$\mathbf{h}_{\underline{Z},\dot{\underline{\mathcal{Z}}}} = (h_0, h_1, h_2, 0) \in H^3, \quad \underline{Z} \in \underline{\mathcal{Z}}. \tag{36}$$

In analogy to the situation for symmetric configurations, the projection onto the closest flat configurations is performed by setting h_3 to zero and lifting back to the hyperboloid, also giving the distance to the closest flat configuration as

$$d(\mathbf{h}, \underline{\mathbf{h}}) = \cosh^{-1}\left(\sqrt{h_0^2 - h_1^2 - h_2^2}\right). \tag{37}$$

Concluding we note how the joint representation relative to $\dot{\underline{\mathcal{Z}}}$ spans the space of joint shapes in an intuitive way: The axes h_1 and h_2 describe the deformation away from symmetric configurations and the axis h_3 represents deformation away from flat configurations.

6. Computations for k-means

The hyperbolic 3-space representation of trivalent joints developed in the last two sections is not just natural from a mathematical standpoint but also beneficial for the computations closest joints and, more particularly, k-means.

6.1. Computing means

It is useful to be able to compute the mean shape of a set of joints efficiently. We can do this based on the representation of the joints in H^3. Let m joints be given by $\{\mathbf{H}_i\}$ or, respectively, $\{\mathbf{h}_i\}$, and denote the mean by $\bar{\mathbf{H}}$, or $\bar{\mathbf{h}}$. In analogy with Euclidean spaces, a widely adopted way to generalize the mean to Riemannian manifolds is to define it as the minimizer of squared distances to the m points; this view and its origins are nicely discussed by Karcher [Kar14].

Karcher's analysis leads directly to a gradient descent algorithm for computing the mean (which has appeared in several guises in the graphics literature) that employs the exponential map to account for the true distances along the manifold. Interestingly, however, Karcher mentions that for surfaces with constant (sectional) curvature, such as the hyperboloid in our case, one may also use the (Minkowski) inner product as the measure of distance in the space, as the resulting definition of the mean is isometric invariant and that is all one can reasonably expect. For a sphere, this would result in embedding the sphere in Euclidean space, then taking the mean in ambient space and projecting back to the sphere. As we know well the results of this procedure are very similar to taking means on the sphere, especially for points close to each other.

For the hyperboloid model of hyperbolic space this leads to the following definition of the mean:

$$\bar{\mathbf{h}} = \underset{\mathbf{h}' \in H^3}{\arg\min} \sum_i -\langle \mathbf{h}', \mathbf{h}_i \rangle_{3,1}. \tag{38}$$

This minimization is easily solved by interpreting it as a constrained minimization in $\mathbb{R}^4$. Let $\mathbf{x}_i = \mathbf{h_i}$ be points with Euclidean mean $\bar{\mathbf{x}}$ in $\mathbb{R}^4$. The desired solution $\mathbf{x}$ should additionally satisfy $x_0^2 - \sum_{k=1}^3 x_k^2 = 1$. This yields the Lagrangian

$$L(\mathbf{x}, \lambda) = \sum_i \left(x_0 x_{i_0} - \sum_{k=1}^{3} x_k x_{i_k} \right) + \lambda \left(x_0^2 - \sum_{k=1}^{3} x_k^2 - 1 \right). \tag{39}$$

Setting the gradient

$$\nabla L = \sum_i \begin{pmatrix} x_{i_0} \\ -x_{i_1} \\ -x_{i_2} \\ -x_{i_3} \end{pmatrix} + 2\lambda \begin{pmatrix} x_0 \\ -x_1 \\ -x_2 \\ -x_3 \end{pmatrix} \tag{40}$$

to zero shows that a necessary condition for the solution is $\mathbf{x} = \mu \bar{\mathbf{x}}$, i.e., a scalar multiple of the arithmetic mean of the points in $\mathbb{R}^4$. Then the hyperboloid constraint identifies the solution as

$$\mathbf{x} = \left(-\langle \bar{\mathbf{x}}, \bar{\mathbf{x}} \rangle_{3,1}\right)^{-\frac{1}{2}} \bar{\mathbf{x}}. \tag{41}$$

Thus, we take the mean of points in H^3 by taking the arithmetic mean in the ambient $\mathbb{R}^4$ and then normalizing the result using the Minkowski inner product.

6.2. Computing the closest joint

In the definition of the hyperbolic joint deformation distance in Equation (26) we have assumed a fixed identification of the outgoing edges of the joints. In practice this will usually be unnecessary and we would like to allow cyclic permutations of the edges when comparing between joints (see Figure 5). To this end, let

$$Z^c = (z_c, z_{c+1}, z_{c+2}) \tag{42}$$

where indices are taken modulo 3. The distance we will often be interested in is then

$$\hat{d}(Z, Z') = \min\left(d\left(Z^0, Z'\right), d\left(Z^1, Z'\right), d\left(Z^2, Z'\right)\right). \tag{43}$$

Note that this distance function is, by taking the minimum of continuous functions, itself continuous.

Given joints in a mesh we would like to efficiently compute for each of them the closest joint from a fixed set of N

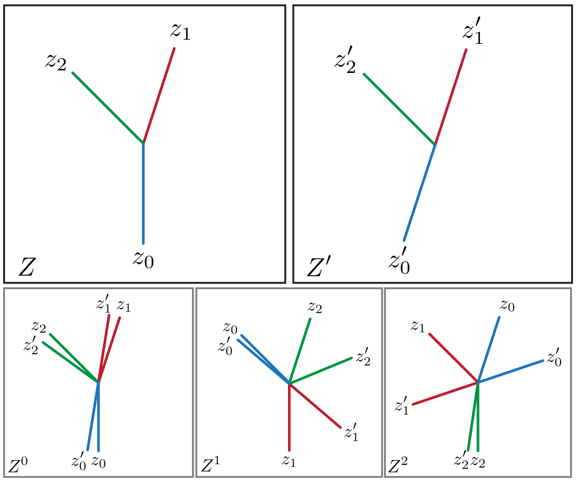

Figure 5: *The joint deformation distance depends on the identification of the outgoing edges between both joints Z and Z′ (top row). In practice, this identification is unnecessary and we therefore define the distance as the smallest distance to all cyclic permutations* (Z^0, Z^1, Z^2) *of Z. In this example the best fits between Z′ and the cyclic permutations of Z are illustrated in the bottom row. The permutation exhibiting the smallest distance is* Z_0 *in this case.*

configurations $\{Z_0, \ldots, Z_N\}$, for example generated in a k-means approach. For each mesh joint we need to determine distances to $\{Z_0^0, Z_0^1, Z_0^2, Z_1^0, \ldots\}$ over all N joints in the set and all cyclic permutations.

This may seem expensive, however, we can precompute the vectors $\{\mathbf{h}_0^0, \mathbf{h}_0^1, \mathbf{h}_0^2, \mathbf{h}_1^0, \ldots\}$ for the set of (cyclic permutations of) the given joint configurations. Then for each mesh joint $\bar{Z}$ its hyperbolic representation $\mathbf{h}_{\bar{Z}}$ needs to be determined to compute the distances using Equation (25). Note that $\cosh^{-1}$ is monotone and hence finding the closest joint in fact only requires computing the Minkowski inner product.

7. Optimizing meshes

Let a dual triangle mesh be defined by vertex positions $\mathbf{V} = \{\mathbf{v}_i \in \mathbb{R}^3\}$ and edges $\mathbf{E} = \{(i,j) \in \{1,\ldots,|\mathbf{V}|\}^2\}$. We denote the *oriented* edge vector between vertices i and j as $\mathbf{e}_{ij} = \mathbf{v}_i - \mathbf{v}_j$. Normalized edge vectors can be used to compute Möbius transformation as described in the preceding sections. We denote by $\mathbf{h}_i \in H^3$ the joint shape representation for vertex i so that there is a unique mapping from $\mathbf{V}$ to $\{\mathbf{x}_i\}$.

With the formulation we detailed in the last sections, a dual triangle mesh with desired joint configurations can be generated by minimizing the distances to target joint shapes. We may further constrain the vertex positions to be on or near a desired surface, and we usually fix the combinatorics of the mesh.

Now, independent of what the desired shape of a joint should be, in any instance we can prescribe the edge vectors of a joint, up to their individual lengths and a common rigid transformation. This is a setup that fits the strategy used in ARAP [SA07] / ShapeUp [BDS*12]: we consider the rotation in a joint and the lengths of edges as additional variables. Then we minimize the distance of the joint shapes to the desired ones as a function of vertex positions, rotations, and edge lengths. In a local step, we update the edge vectors, based on the constraint of the joint shape, and then by matching the edges with a rotation and scale to the current vertex positions. In a global step, we optimize the vertex positions such that the squared distance to the desired edge vectors is minimal. Since the energy is minimized in each step, this approach often yields a local minimum. More general justification for this approach is given by Bouaziz et al. [BDS*12].

A minor difference is that we compute the optimal rotation and edge lengths based on the ideal joint shape and the *vertex positions in the last iteration*: given the current edge vectors $\mathbf{e}_{ij}$ pointing from vertex i to any neighbor j, we first project the joint shape $\mathbf{h}_i$ onto the constraint set yielding updated edge vectors $\mathbf{e}'_{ij}$ – the details of this step are discussed in the following sections. Note that this step will generally lead to different oriented edge vectors $\mathbf{e}'_{ij} \neq -\mathbf{e}'_{ji}$. Now we find scaling factors so that the updated edges have the same length as the old edge vectors, i. e.

$$s_{ij} = \frac{\|\mathbf{e}_{ij}\|}{\|\mathbf{e}'_{ij}\|}. \tag{44}$$

Then we align the old and the new frame by a rotation $\mathbf{R}_i \in \mathbb{R}^{3\times 3}$ in euclidean space that minimizes the squared differences

$$\underset{\mathbf{R}_i^\mathsf{T}\mathbf{R}_i = I}{\arg\min} \sum_{(i,j)\in E} \|\mathbf{e}_{ij} - \mathbf{R}_i s_{ij} \mathbf{e}'_{ij}\|^2, \tag{45}$$

which can be computed using SVD [SA07]. Based on these edge vectors we can then solve the linear system describing the minimizer of

$$\begin{aligned} \underset{\mathbf{V}}{\arg\min} \sum_{(i,j)\in E} &\|\mathbf{v}_i - \mathbf{v}_j - R_i s_{ij} \mathbf{e}'_{ij}\|^2 + \\ &\|\mathbf{v}_j - \mathbf{v}_i - R_j s_{ji} \mathbf{e}'_{ji}\|^2. \end{aligned} \tag{46}$$

The optimization towards the different constraints on the joints only differs in how the edge vectors $\mathbf{e}'_{ij}$ are computed to reflect the desired joint shapes. We describe this next.

7.1. Symmetric and flat joints

Projecting the joint shapes onto the closest symmetric or flat joint shape follows directly from the derivation in Section 5. In each iteration of the optimization, we compute the closest constrained joint shape for each of the joints. Given edges $\mathbf{e}_{ij}$

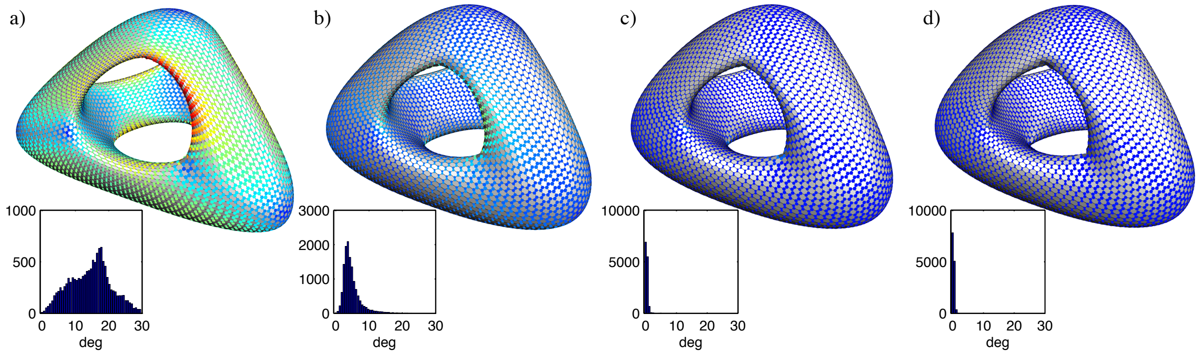

Figure 6: *Optimization for symmetric joints: Hexagonal input mesh with colorized joints indicating the distance to the nearest symmetric configuration (red corresponds to high distances). a) Input mesh b-d) Result after 1,10 and 100 iterations. The distribution of angular deviation w.r.t. the nearest symmetric joint is shown in the histograms.*

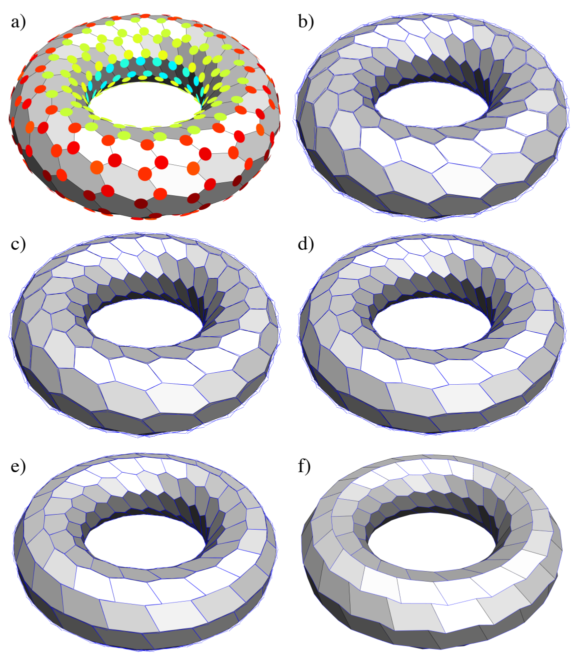

Figure 7: *Optimization for planar joints: a) Hexagonal input mesh with colorized joints indicating the distance to the nearest planar configuration (red corresponds to high distances). b-f) The joints are optimized iteratively toward planar configurations (number of iterations from b to f: 0,10,20,30,100).*

for vertex i we normalize the edge vectors, perform stereographic projection and then $\mathbf{M}_{Z_i}$ (Equations (3) and (9)). Using the precomputed $\mathbf{M}_{\dot{\underline{Z}}}^{-1}$ we generate the hyperbolic point relative to the symmetric flat configuration:

$$\mathbf{h}_i = (h_{i_0}, h_{i_1}, h_{i_2}, h_{i_3}) \cong \mathbf{H}_i = \mathbf{M}_{\dot{\underline{Z}}}^{-1} \mathbf{M}_{Z_i} \mathbf{M}_{Z_i}^{*} (\mathbf{M}_{\dot{\underline{Z}}}^{-1})^{*} \quad (47)$$

Then we get the updated joint shape representation by taking either

$$\mathbf{h}'_i = \sqrt{h_{i_0}^2 - h_{i_3}^2}\,(h_{i_0}, 0, 0, h_{i_3}), \quad (48)$$

or

$$\mathbf{h}'_i = \sqrt{h_{i_0}^2 - h_{i_1}^2 - h_{i_2}^2}\,(h_{i_0}, h_{i_1}, h_{i_2}, 0) \quad (49)$$

for the case of symmetric, resp. flat joint shapes. This yields $\mathbf{M}'_i = \mathbf{H}_i + \mathbf{I}$, which represents a Möbius transformation that maps the flat symmetric frame $\dot{\underline{Z}}$ to the symmetric configuration we are interested in. Note that the matrix representation is *not* normalized (compare to Equation (24)), but this is irrelevant since we are computing with homogeneous coordinates. We apply this transformation to $\dot{\underline{Z}}$ and get the homogeneous coordinates of the desired symmetric frame, which we turn into (unit) vectors of $\mathbb{R}^3$ back projecting onto the sphere.

Together with the procedure described above this is all we need to optimize meshes to consist only of symmetric or flat joints. Examples are shown in Figures 6, 7, and 8. Figure 6 shows the optimization of a mesh without boundary towards symmetric joint shapes. The histograms and color coding how the joints converge to be perfectly symmetric, while only moderately deviating from the original shape. In Figure 7 we demonstrate successfully modifying the joint shapes of a torus to be all planar up to numerical precision (while the faces are necessarily *not* planar). Note the interesting duality to the torus of planar hexagons (but non-planar joint shapes) of Li et al. [LLW15, Figure 12].

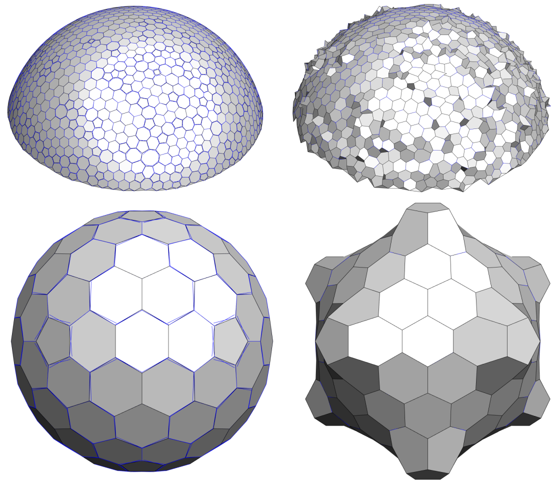

Figure 8: *If all joint shapes are symmetric and faces are nearly planar, face degree takes the role of discrete curvature. We show this here by starting with trivalent meshes of varying face degree (left). After optimization for symmetric joint shapes irregularity of face degree is clearly exhibited in the geometry (right).*

We wish to stress that near-planar faces and symmetric joint shapes result in heavy dependence on face degree (see Figure 8, which then represents a discrete measure of curvature. A similar effect can be observed if we optimized triangles in a triangle mesh to be equilateral [IGG01].

7.2. Discrete joint shapes

Given a set of k joint shapes $\{\bar{\mathbf{h}}_j\}$ (for example, the joints shown in Figure 1), our goal is to optimize the mesh such that it consists of these joints only. For each vertex, we compute the joint shape representation $\mathbf{h}_i$ and then find the joint shape $\bar{\mathbf{h}}_j$ that minimizes the distance $\hat{d}(\mathbf{h}_i^c, \bar{\mathbf{h}}_j)$, i. e. the distance *considering all cyclic permutations* $c \in \{0,1,2\}$ *of the joints.* As explained in Section 6 we precompute the cyclic permutations for all target joint shapes. Then we set the edge vectors of each vertex to the corresponding joint shape of the nearest element in $\{\bar{\mathbf{h}}_j\}$.

However, we can also follow recent work in the literature that aims at representing a surface using a small set of representative facets [EKS*10,FLHCO10,SS10], optimized for the given surface, yet for the joint shapes.

To generate the set $\bar{\mathbf{H}}_k = \{\bar{\mathbf{h}}_j\}$ of representative joints $\bar{\mathbf{h}}_j$ we cluster the individual mesh joints and compute representative joint shapes for each group by k-means: the computations alternate between two basic steps: (1) assigning each joint to the nearest representative joint in $\bar{\mathbf{H}}_k$, and (2) computing the mean of each group and updating $\bar{\mathbf{H}}_k$. All computations take place in hyperbolic space H^3: for nearest neighbor computations we employ the hyperbolic distance and computing means is done as detailed in section 6.1.

In our implementation we repeat (re-)assignment and cluster updating typically 10 times. In most cases this is sufficient to reach a local minimum. However, we observed that the solutions are sensitive to the choice of initial data. Instead of the standard random initialization we employ k-means++ [AV07] sampling, which yields more robust solutions.

The results of the optimization for the dual of the Lilium tower roof model for different k is shown in Figure 9. In a) the colored vertices indicate the closest reference joint $\bar{\mathbf{h}}_j \in \bar{\mathbf{H}}_k$ computed on the input mesh. To visualize the joints and their assignments to the closest mean we project all $\mathbf{h}_i \in H^3$ (and their cyclic permutations) to the Poincaré ball model of hyperbolic space (Figure 9c,d). The assignments resulting from computing k-means on the initial mesh joints is shown in c) with the same colors used in a). The cyclic permutations of each joint are displayed in the same color while only the closest of the three possible permutations determines the assignment. The results after iteratively optimizing vertex positions in order minimize the distances to the reference set $\bar{\mathbf{H}}_k$ are shown in b) and d). In b) the colored vertices indicate the distance to the assigned reference joint. Note that the distances to the optimal joint shapes decreases with an increasing number of reference joints. In d) one can see that the optimized joints $\mathbf{x}_i$ form clusters in H^3.

8. Discussion

We have derived a rotation-invariant representation of 3-valent joints in hyperbolic 3-space H^3 that endows their shape space with a natural deformation metric. The hyperbolic representation facilitates computation and, in particular, yields simple expression for important quantities such as the distance to the closest rotationally symmetric configuration.

The hyperbolic joint space metric enables us to optimize dual triangle meshes for joint constraints using the ShapeUp optimization framework. Another option might be to start with an appropriately constrained mesh and then explore the space of admissible meshes [YYPM11]. We have demonstrated that it is possible to optimize meshes to use a discrete set of joint shapes as well as symmetric or flat joints (see Figures 10 and 11 for additional results). The orientation of the joint shape unconstrained has been left unconstrained in our experiments. Restricting flat joints to lie in a common plane would directly lead to a parameterization approach [LZX*08].

We feel that the central contribution of our work lies in identifying hyperbolic geometry as a powerful representation for the shape of frames. We believe that this can be applied to scenarios beyond the one considered here such as:

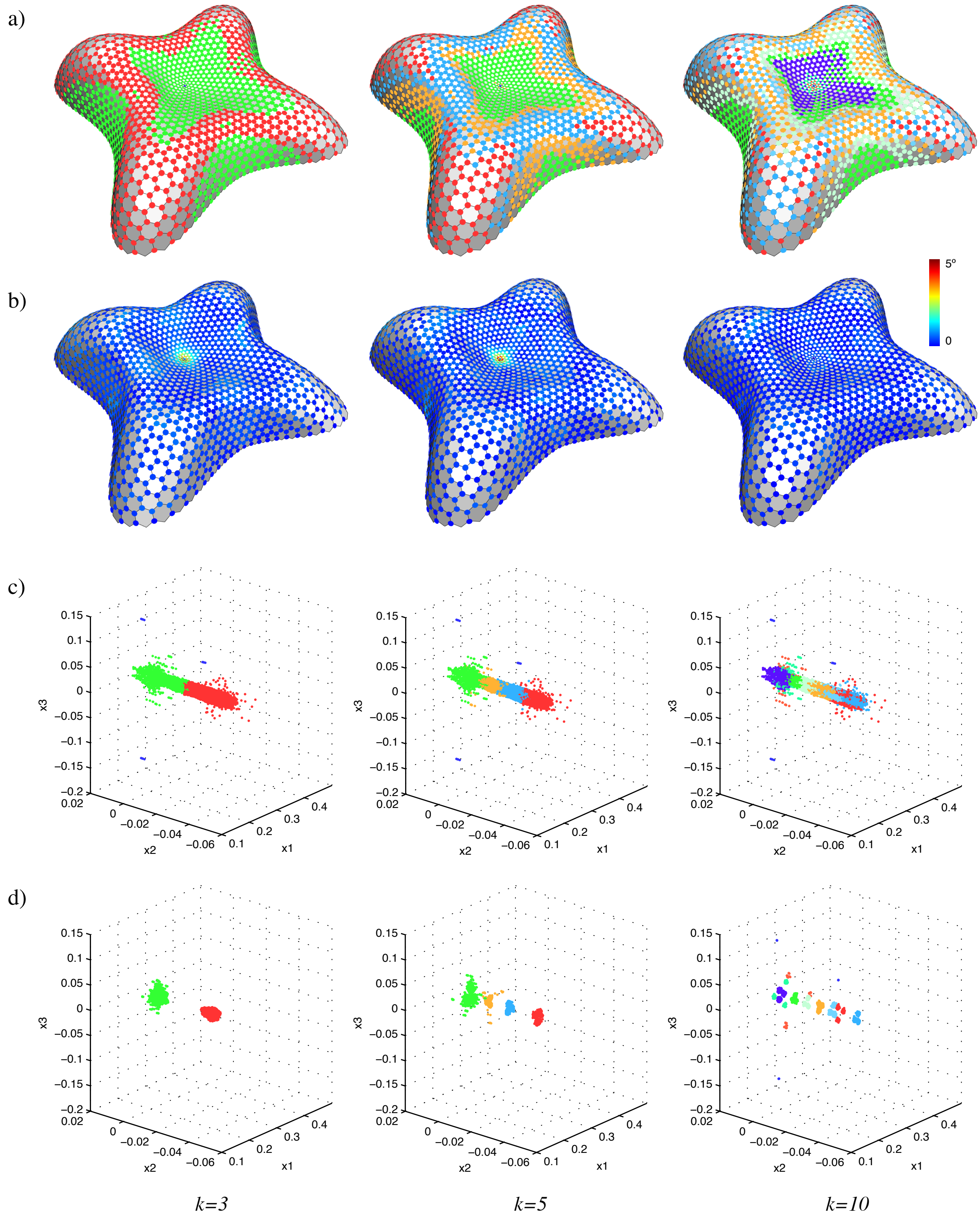

Figure 9: *Optimization for discrete joint shapes with different number of reference joints ($k = 3,5,10$): a) Initial mesh and discrete joint assignment, b) Optimized vertex positions w.r.t. E_k. Color indicated angular deviation to assigned reference joint at each vertex (red: high deviation). c-d) Visualization of joints $\mathbf{h}_i \in H^3$ in the Poincaré ball model with initial joints (c) and optimized joints (d).*

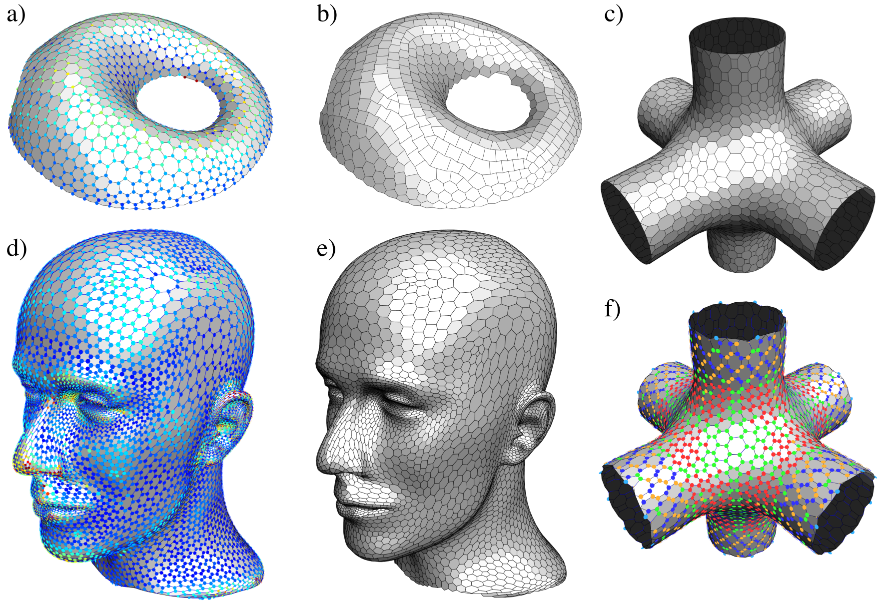

Figure 10: *Joint optimization: Input meshes a) and d) are optimized toward flat joint configurations (see Sec 7.1). The colored vertices indicate deviation from flatness (red: high deviation). The resulting meshes are shown in b) and e). The mesh f) is the result of optimizing input mesh c) to consist of joints from a discrete set only (see Sec 7.2). The color in f) identifies the assignment to the respective joint in the set.*

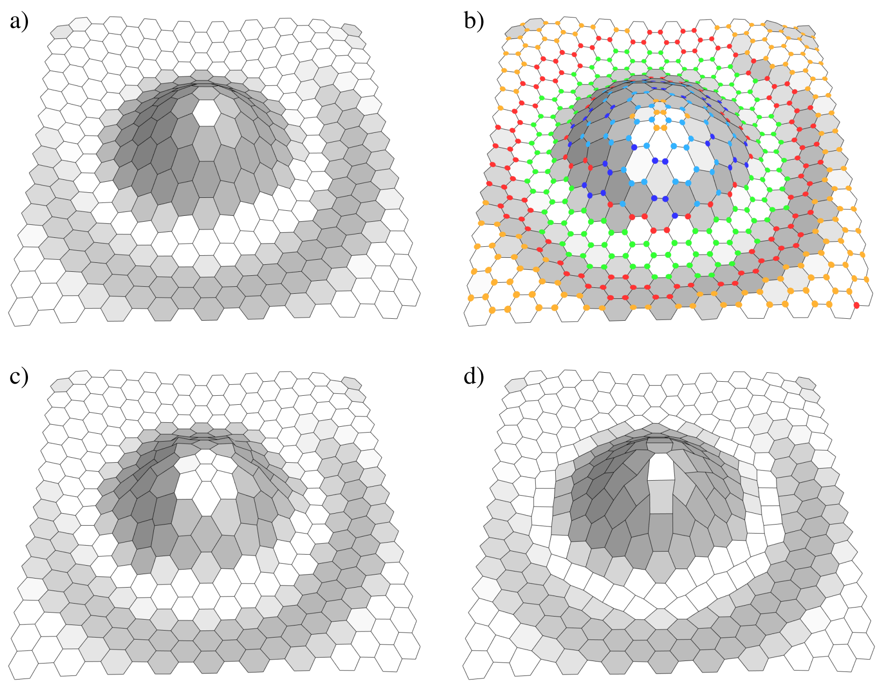

Figure 11: *Optimization results: a) Input: dual triangle mesh. b) Optimized toward discrete joint set: Colors indicate assignment of vertex joints to reference joint in the set (Sec 7.2). c) Optimized for symmetric joint configurations (Sec 7.1). d) Optimized toward flat joint configurations (Sec 7.1).*

- A triangle mesh can be used as a free-form surface in architecture for so called point-folding structures where a pyramidal shape is supported over each triangle. Manufacturing costs for such surfaces can be reduced by restricting the possible pyramidal shapes. As shown in previous work [ZCBK12], this can be done by clustering the shapes. Our hyperbolic 3-space representation can be employed for pyramid shapes and the algorithm for k-means clustering of joint shapes described in Section 7.2 is applicable.
- Instead of representing three edges incident to a vertex in hyperbolic 3-space, one may also represent three normal vectors. The normals could, for example, describe the local neighborhood of the point, and hence provide a local descriptor of the surface in hyperbolic 3-space. This representation would again be invariant to rotation, translation, *and scale*, and be minimal for this invariance. In fact, Möbius geometry has already been used successfully for symmetry detection [KLCF10] or establishing correspondence [LF09].

References

[AV07] ARTHUR D., VASSILVITSKII S.: k-means++: The advantages of careful seeding. In *ACM-SIAM symposium on Discrete algorithms* (2007), pp. 1027–1035. 11

[BDS*12] BOUAZIZ S., DEUSS M., SCHWARTZBURG Y., WEISE T., PAULY M.: Shape-up: Shaping discrete geometry with projections. *Computer Graphics Forum 31*, 5 (Aug. 2012), 1657–1667. doi:10.1111/j.1467-8659.2012.03171.x. 9

[CW07] CUTLER B., WHITING E.: Constrained planar remeshing for architecture. In *Proceedings of Graphics Interface 2007* (New York, NY, USA, 2007), GI '07, ACM, pp. 11–18. doi:10.1145/1268517.1268522. 1

[EKS*10] EIGENSATZ M., KILIAN M., SCHIFTNER A., MITRA N. J., POTTMANN H., PAULY M.: Paneling architectural freeform surfaces. *ACM Trans. Graph. 29*, 4 (July 2010), 45:1–45:10. doi:10.1145/1778765.1778782. 1, 11

[FLHCO10] FU C.-W., LAI C.-F., HE Y., COHEN-OR D.: K-set tilable surfaces. *ACM Trans. Graph. 29*, 4 (July 2010), 44:1–44:6. doi:10.1145/1778765.1778781. 1, 11

[IGG01] ISENBURG M., GUMHOLD S., GOTSMAN C.: Connectivity shapes. In *Proceedings of the Conference on Visualization '01* (Washington, DC, USA, 2001), VIS '01, IEEE Computer Society, pp. 135–142. URL: http://dl.acm.org/citation.cfm?id=601671.601691. 11

[Kar14] KARCHER H.: Riemannian center of mass and so called karcher mean, 2014. arXiv:1407.2087. 8

[KLCF10] KIM V., LIPMAN Y., CHEN X., FUNKHOUSER T.: Möbius transformations for global intrinsic symmetry analysis. *Computer Graphics Forum (Symposium on Geometry Processing) 29*, 5 (July 2010). 14

[LF09] LIPMAN Y., FUNKHOUSER T.: Möbius voting for surface correspondence. *ACM Trans. Graph. 28*, 3 (July 2009), 72:1–72:12. doi:10.1145/1531326.1531378. 14

[LLW15] LI Y., LIU Y., WANG W.: Planar hexagonal meshing for architecture. *Visualization and Computer Graphics, IEEE Transactions on 21*, 1 (Jan 2015), 95–106. doi:10.1109/TVCG.2014.2322367. 1, 11

[LZX*08] LIU L., ZHANG L., XU Y., GOTSMAN C., GORTLER S. J.: A local/global approach to mesh parameterization. In *Proceedings of the Symposium on Geometry Processing* (Aire-la-Ville, Switzerland, Switzerland, 2008), SGP '08, Eurographics Association, pp. 1495–1504. URL: http://dl.acm.org/citation.cfm?id=1731309.1731336. 11

[PLW*07] POTTMANN H., LIU Y., WALLNER J., BOBENKO A., WANG W.: Geometry of multi-layer freeform structures for architecture. *ACM Trans. Graph. 26*, 3 (July 2007). doi:10.1145/1276377.1276458. 1

[PSB*08] POTTMANN H., SCHIFTNER A., BO P., SCHMIEDHOFER H., WANG W., BALDASSINI N., WALLNER J.: Freeform surfaces from single curved panels. *ACM Trans. Graph. 27*, 3 (Aug. 2008), 76:1–76:10. doi:10.1145/1360612.1360675. 1

[SA07] SORKINE O., ALEXA M.: As-rigid-as-possible surface modeling. In *Proceedings of the Fifth Eurographics Symposium on Geometry Processing* (Aire-la-Ville, Switzerland, Switzerland, 2007), SGP '07, Eurographics Association, pp. 109–116. URL: http://dl.acm.org/citation.cfm?id=1281991.1282006. 9

[SS10] SINGH M., SCHAEFER S.: Triangle surfaces with discrete equivalence classes. *ACM Trans. Graph. 29*, 4 (July 2010), 46:1–46:7. doi:10.1145/1778765.1778783. 1, 11

[Vax12] VAXMAN A.: Modeling polyhedral meshes with affine maps. *Comp. Graph. Forum 31*, 5 (Aug. 2012), 1647–1656. doi:10.1111/j.1467-8659.2012.03170.x. 1

[YYPM11] YANG Y.-L., YANG Y.-J., POTTMANN H., MITRA N. J.: Shape space exploration of constrained meshes. *ACM Trans. Graph. 30*, 6 (Dec. 2011), 124:1–124:12. doi:10.1145/2070781.2024158. 1, 11

[ZCBK12] ZIMMER H., CAMPEN M., BOMMES D., KOBBELT L.: Rationalization of triangle-based point-folding structures. *Comp. Graph. Forum 31*, 2pt3 (May 2012), 611–620. doi:10.1111/j.1467-8659.2012.03040.x. 14