Using isometries for computational design and fabrication

CAIGUI JIANG, KAUST, Saudi Arabia
HUI WANG, KAUST, Saudi Arabia
VICTOR CEBALLOS INZA, KAUST, Saudi Arabia
FELIX DELLINGER, TU Wien and TU Graz, Austria
FLORIAN RIST, KAUST and TU Wien, Austria
JOHANNES WALLNER, TU Graz, Austria
HELMUT POTTMANN, KAUST, Saudi Arabia

We solve the task of representing free forms by an arrangement of panels that are manufacturable by precise isometric bending of surfaces made from a small number of molds. In fact we manage to solve the paneling task with surfaces of constant Gaussian curvature alone. This includes the case of developable surfaces which exhibit zero curvature. Our computations are based on an existing discrete model of isometric mappings between surfaces which for this occasion has been refined to obtain higher numerical accuracy. Further topics are interesting connections of the paneling problem with the geometry of Killing vector fields, designing and actuating isometries, curved folding in the double-curved case, and quad meshes with rigid faces that are nevertheless flexible.

CCS Concepts: • Computing methodologies → Shape modeling; Optimization algorithms.

Additional Key Words and Phrases: discrete differential geometry, computer-aided design, computational fabrication, architectural geometry, discrete isometry, isometric registration, Killing vector field

ACM Reference Format:

1 INTRODUCTION

In this work we apply recent progress in isometric mappings to the paneling of freeform architectural skins. The paneling problem, i.e., the decomposition of curved shapes into smaller elements that are more easily manufacturable, is crucial to the actual realization of freeform designs on the architectural scale. The choice of panel geometry is central to visual appearance and thus is an integral part of the artistic aspect of freeform architecture. At the same time this choice has far-reaching consequences on manufacturing and poses big computational challenges. There are at least two reasons for that: one is that a geometric fit of all panels may amount to a global and nonlinear system of constraints that is hopeless to solve by black box methods. Another is the sometimes prohibitive cost of manufacturing panels, in particular double-curved ones. For an overview of this broad topic we refer to [Pottmann et al. 2015].

In the case of an overall smooth skin, the most successful paradigm for reducing costs has been to break up the fabrication process into two stages, namely manufacturing molds first, and afterwards making the panels from them. Since typically molds are more expensive than striking an individual panel from the mold, the idea is to reuse molds. This concept has been used in practice several times already, e.g. for the Arena Corinthians stadium in São Paulo which was completed in 2014.

Beyond its literal meaning, “mold” may refer to any machine or technique used to produce a certain family of panels. One can think of all flat glass panels to be associated with the same mold (which is cheap, of course). Likewise all glass panels that lie on the surface of any right circular cylinder correspond to a single “mold” which in reality is the machine capable of producing cylindrical shapes, cf. [Baldassini et al. 2013].

Our work considers molds in the literal sense. In contrast to previous work on paneling in architecture, panels are put into place not by a rigid transformation, but are bent isometrically. This method
is applicable only to materials that allow isometric deformations like sheet metal, certain plastics and textiles, and it drastically reduces the number of molds that are needed. We are able to accurately model this process thanks to recent progress in the computational treatment of isometric mappings: we employ and improve the ‘checkerboard pattern’ method proposed by Jiang et al. [2020].

A surprising discovery is that we found it sufficient to use a rather small set of molds of high intrinsic symmetry, namely surfaces of constant Gaussian curvature. This means that in most instances we were able to cover free forms by panels that are either developable, or are obtained from a small set of spherical and pseudo-spherical molds. On an abstract mathematical level this procedure corresponds to the approximation of the Gaussian curvature function by a piecewise-constant function. It was unexpected how well this idea works in practice.

It is important to stress the difference between our work and previous progress on non-rigid isometric registration in Computer Graphics and Computer Vision [Bronstein et al. 2008], the greater part of which is concerned with as-isometric-as-possible matching of different shapes, not with exact isometric registration. We would like to add two historical remarks. Firstly, the well-known sails of the Sydney opera house enjoy spherical geometry of a single radius [Lewis 1973] and thus constitute a panelization via a single spherical mold.

Secondly we point to the 19th century invention of French curves useful for drawing a great number of curves for which ruler and compass are not sufficient but which occur in technical drawings. The act of drawing can be thought of as panelization by means of the French curve set as molds. One could say that our experimental results confirm that as far as isometrically bending panels are concerned, we can realize a great number of freeform architectural skins by flat/spherical/pseudospherical molds, and we do not have to resort to more complicated “French curve” molds.

1.1 Contributions
The new contributions of this paper are the following.

- We improve the precision of Jiang et al.’s [2020] checkerboard method for isometric mappings by means of adapted regularizers (§2.2). We also provide new insights on approximation order (§2.1).
- §3.1 presents a structure-preserving discretization of the concept of infinitesimal isometry.
- We compute Killing vector fields, i.e., intrinsic infinitesimal isometries, in a simple way (§3.2). They are employed to detect surfaces that possess advantages in fabrication because they are isometric to surfaces exhibiting symmetries like rotational surfaces.
- In §4 we discuss precise isometric registration, which in §5.1 is used for the paneling problem. We present two computational pipelines for paneling freeform architectural skins with bendable materials — one with constant Gaussian curvature surfaces as molds, and another one with molds of rotational symmetry.
- Further applications (§5.2) include the design of quad meshes which are flexible if the faces are moving rigidly and are connected by the edges as hinges.

1.2 Previous Work

Previous Work in Freeform Architecture. On the architectural scale, smooth shapes can be obtained by different means, e.g. as equilib-rium of a membrane. More often they are an assembly of individual panels that seamlessly join to form the surface. The high cost of manufacturing double-curved panels in the past has led architects to designs with repetitive geometry. This can e.g. take the form of a hidden rotational symmetry which is not visually obvious because panel boundaries attract attention away from it – the 2003 lens-shaped entrance to the St. Lazare metro station in Paris is an instance of this. For such simple surfaces, panels in different places of the surface can be struck from the same mold, and the cost of molds plus panels is significantly reduced. An extension of this idea to true free forms has been performed by Eigensatz et al. [2010] and has been used in practice [Schiftner et al. 2013]. The present paper can be seen as a continuation of this work, but with non-rigid panels that bend isometrically.

As to curved panels originating in a flat rest state, prior contributions are mainly concerned with elastic materials, in which case paneling is modeled by mappings that are not isometric. Examples include flat pre-stretched tilings assuming a desired shape upon release [Guseinov et al. 2017], auxetic micro-mechanisms [Konakovici-Lukovic et al. 2018], micro-mechanisms based on spiral springs [Malomo et al. 2018] and deployable structures of a grid-shell nature [du Peloux et al. 2013; Panetta et al. 2019]. Most recently, and relevant to our work, Gavrili et al. [2020] used learning methods to panel architectural freeform skins by cold-bent glass, taking the restrictions of the material into account.

Previous Work on Isometric Mappings Between Surfaces. Finding correspondences between geometric shapes is one of the most significant and diverse problems in geometry processing. We here focus on isometric mappings, i.e., mappings preserving the intrinsic distances measured inside the surface. Since in general two surfaces cannot be connected by an isometry, mappings that are near-isometric have been studied [Claici et al. 2017; Liu et al. 2009; Pietroni et al. 2010; Sorkine and Alexa 2007]. Relevant work on isometric mappings has been done by Chern et al. [2018]. Based on a discrete theory of spin structures, they treat isometric mappings, particularly immersions. Recently, Sassen et al. [2020] investigated...
This is an old topic and connections to meshes are summarized.

The unit pseudosphere shown at right enjoys $K = -1$ and is generated by rotating the curve $y = \log(\frac{1}{2}(1 + \sqrt{1 - x^2})) - \sqrt{1 - x^2}$ about the $y$ axis.

The flexions of triangle meshes with rigid faces via the manifold of realizations of fixed edge lengths but variable dihedral angles. They formulate the corresponding integrability conditions and solve both exact and approximate reconstruction of meshes, which includes computing isometric mappings.

Our paper is based on a different approach, namely checkerboard patterns as proposed by Peng et al. [2019]. They work with patterns exhibiting rectangles. Extending this idea to the general case of parallelograms, Jiang et al. [2020] developed an isometric deformation model by requiring that the parallelograms in a checkerboard pattern move rigidly. This is an instance of the more general principle to model isometry via keeping the shape of small elements, see e.g. [Bouaziz et al. 2012].

**Isometric Bending.** The differential geometry of time-continuous isometric bending of surfaces is extensively surveyed by [Sabitov 1992]. Relevant to our work are statements on existence of bending determined by initial values.

As to applications in engineering and nature, it is well known that the bending of paper and similar thin-sheeted materials is modeled by isometric mappings. Most of the extensive literature on this subject is concerned with developable surfaces created by bending flat sheets. We do not review it here but instead refer to [Jiang et al. 2020; Liu et al. 2006]. Bending of non-flat surfaces occurs in manifold ways in herbal tissue, and special cases of it have inspired technology: The “Flectofin” shading mechanism is modeled after the natural hinge present in the bending behavior of the flower of *Strelitzia reginae*. See [Lienhard et al. 2011; Masselter et al. 2012] and Figure 2.

**Infinitesimal Isometries.** We also make use of the concept of infinitesimal isometry, which is the time-derivative vector field of a deformation that is isometric up to the first order of differentiation.

This is an old topic and connections to meshes are summarized by Sauer [1970]. Tangential infinitesimal isometries (Killing vector fields) have a flow that is isometric. For the geometry of Killing vector fields we refer to the survey [Myers 1936]. Computation and applications have been demonstrated by [Ben-Chen et al. 2010; Solomon et al. 2011a]. In particular they have been employed to detect parts of surfaces enjoying intrinsic symmetries [Solomon et al. 2011b].

**Constant Gaussian Curvature Surfaces.** In this paper we employ surfaces of constant Gaussian curvature, i.e., surfaces isometric to parts of the plane, or to appropriately scaled copies of the unit sphere $S^2$ resp. the unit hyperbolic plane $H^2$ [Alekseevskij et al. 1993]. An often-used surface exhibiting $K = -1$ is the unit pseudosphere, see Fig. 3. Efforts to build surfaces isometric to a part of $H^2$ such as the one shown by Fig. 3 started as soon as $H^2$ was discovered. They include kinematic metal models [Wunderlich 1951] and even crochet [Henderson and Taimina 2001]. Both the textile and the metal versions are relevant to paneling on the architectural scale.

**Previous Work on Isometric Registration.** Non-rigid registration is a much studied topic in Vision and Geometry Processing. The special case of isometric registration has been discussed by various authors, see e.g. [Huang et al. 2008; Wand et al. 2007]. Both the initial alignment (global registration) and the later fine tuning through numerical optimization (local registration) have been treated. These methods have been successfully employed for tasks in engineering [Klein et al. 2014; Sacharov et al. 2011]. Most of the work however is not addressing high accuracy applications like the ones we have in mind and thus cannot be used directly. Huang et al. [2008] use features of the target surfaces for registration. This is not suitable for our setting, architectural freeform skins typically lacking prominent features.

## 2 DISCRETE ISOMETRIES

For the computations with surfaces and their isometric mappings, we represent a surface by a quad mesh which is further subdivided into a checkerboard pattern as proposed by Jiang et al. [2020]. Each original face is associated with a quad, formed by the edge midpoints, while each original vertex is associated with a quad, formed by the midpoints of edges emanating from that vertex. In the inset figure, inscribed parallelograms are shown in yellow. Figure 4 illustrates the fact that they are parallel to the diagonals of faces of the original mesh. Following Jiang et al. [2020], we say the surface deforms isometrically, if each inscribed parallelogram moves as a rigid body.

### 2.1 Approximation Order of Discrete Isometries

We need to convince ourselves that discrete-isometric mappings of checkerboard patterns do indeed approximate continuous isometries, and that the discrete-isometric conditions take away just the right number of degrees of freedom.

For that, consider a quad mesh which is the image, under a smooth mapping $\phi$, of the standard lattice $\mathbb{Z}^2$ with stepsize $h$. Focus on the
are discrete counterparts of \(\phi\) by first order differences of vertices, e.g. \(\partial_y \phi = \frac{1}{h} (v_1 - v_0) + O(h)\) and \(\partial_x \phi = \frac{1}{h} (v_3 - v_0) + O(h)\). Actually, these differences properly express the derivatives of \(\phi\) in edge midpoints, since \(\partial_x \phi (m_e) = \frac{1}{h} (v_1 - v_0) + O(h^2)\), for \(m_e = (h/2, 0/2)\). An analogous statement holds for \(\partial_y\) and the midpoint \((0, h/2)\). In order to achieve the higher approximation order \(O(h^2)\) for both first derivatives in the same point, we resort to a trick and apply a change of coordinates. With \(u = x + y\) and \(v = x - y\), derivatives in the face midpoint \(m_f = (h/2, h/2)\) are expressed as

\[
\partial_u \phi (m_f) = \frac{v_2 - v_0}{\sqrt{2h}} + O(h^2), \quad \partial_v \phi (m_f) = \frac{v_3 - v_1}{\sqrt{2h}} + O(h^2),
\]

as can be seen from a Taylor expansion centered in \(m_f\).

As to mappings from one mesh to another, consider meshes \(M = \phi(k\mathbb{Z}^2)\) and \(M' = \phi'(k\mathbb{Z}^2)\), so that the mapping \(\psi : M \rightarrow M'\) obeys \(\psi \circ \phi = \phi'\). With differentials \(d\phi = (\partial_u \phi, \partial_v \phi)\) and \(d\phi' = (\partial_u \phi', \partial_v \phi')\) we have \(d\psi \circ d\phi = d\phi'\). This directly leads to the second order approximation

\[
d\psi : (v_2 - v_0, v_3 - v_1) = (v'_2 - v'_0, v'_3 - v'_1) + O(h^2),
\]

which is valid in the face “midpoint” \(\phi(m_f)\). In terms of edges of the inscribed parallelograms according to Figure 4, this relation between differentials is expressed as

\[
d\psi \cdot (m_{12} - m_{01}, m_{30} - m_{01}) = (m'_{12} - m'_{01}, m'_{30} - m'_{01}) + O(h^2).
\]

One interpretation of this equation is the following: \(\psi\) is isometric if and only if, up to an error of magnitude \(h^2\), the inscribed parallelograms are isometric to each other.

This isometry of parallelograms is exactly our definition of a discrete-isometric mapping. It amounts to 3 scalar constraints per face. The higher approximation order is achieved because the set of constraints respects the mesh’s combinatorial symmetries.

In analogy, \(\psi\) is a continuous isometry, if and only if everywhere on the surface the linear mapping \(d\psi\) is isometric. Thus we expect that discrete-isometric mappings as we defined them faithfully approximate continuous isometries. This expectation has also been confirmed by experiments.

If we had insisted that the faces of the original mesh move rigidly, we would have imposed not 3, but 4 scalar constraints per face. Actually most quad meshes are rigid if faces are kept rigid, so requiring this would not have been a faithful discretization of isometric mapping. We are coming back to this topic in §5.2.2.

2.2 Computing Discrete Isometries with High Accuracy

Jiang et al. [2020] use optimization to compute isometries between meshes \(M, M'\). We recapture their method to the extent necessary to explain the improvements we make. The variables in the optimization are the vertices of \(M'\), and the target functional to be minimized has the form

\[
E_{iso} = \sum_{\Delta_{ijk}} |\Delta'_{ijk}|^2, \quad \lambda_{iso} \rightarrow \min.
\]

We thus define

\[
E_{iso, M'} = \sum_{\Delta_{ijk}} |\Delta'_{ijk}|^2, \quad \lambda_{iso, M'} \rightarrow \min.
\]

Summation is over all triples \(i, j, k\) of consecutive vertices of a mesh polyline in \(M'\). If the mesh \(M\) is not fixed during optimization, an analogous energy \(E_{iso, M}\) is used. This fairness energy is based on the interpretation of mesh polylines as discrete versions of curves \(x(t)\) — the 2nd differences occurring in \(E_{iso}\) are discrete counterparts of covariant derivatives \(\frac{d^2 x}{dt^2}\), cf. [do Carmo 1976]. After estimating unit normal vectors \(n_j, n'_j\) in the vertices of \(M, M'\), we let \(\Delta'_{ijk} = \lambda_{ij} = |\Delta_{ijk}| n_j \) and similar for \(\lambda'_{ijk}\).

It is well known that isometries respect covariant derivatives, i.e., we expect that \(d\psi (\Delta'_{ijk}) = \lambda'_{ijk}\) holds. We thus formulate a tangential fairness functional

\[
E_{tang, M'} = \sum_{\Delta_{ijk}} |d\psi_j (\Delta_{ijk}) - \lambda'_{ijk}|^2.
\]
Vector fields representing infinitesimal isometries occur several times in our computations. Firstly a special case (Killing vector fields) of interest we consider the velocity vectors \( \mathbf{v} \) that fit a certain r-field of the kind given by Equ. (5): There are \( \mathbf{v} \)-velocity fields which are as Killing as possible yield information on paneling. The time-derivatives of such isometries are i-fields tangential to surfaces (Killing fields) exist on Fig. 7. Infinitesimal isometries tangential to surfaces (Killing fields) exist on...
any piece of $\Phi$ within $\Phi$ in an isometric way until the boundary is reached. For an r-field, this motion is even a rigid body motion.

Within Geometry Processing, several contributions deal with the computation and applications of Killing vector fields, see e.g. [Ben-Chen et al. 2010]. One notable feature of Killing fields is that they are always tangential to the isolines of Gaussian curvature $K$ (which follows directly from the fact that Gaussian curvature is invariant under isometries, see Fig. 7). The Killing fields of a surface $\Phi$ constitute a linear space $\mathcal{K}_\Phi$ which reveals geometric information [Myers 1936]:

- $\dim \mathcal{K}_\Phi \geq 1$: Simply connected pieces of $\Phi$ can be isometrically matched to a rotational surface, possibly with overlaps. This deformation transforms the Killing field to an r-field tangential to parallel circles. Its orthogonal trajectories are mapped to meridian curves, see Figures 7 and 10.
- $\dim \mathcal{K}_\Phi \geq 2$: Gaussian curvature $K$ is constant. Simply connected pieces of $\Phi$ can be mapped isometrically, possibly with overlaps, to a plane if $K = 0$, to a sphere if $K > 0$, and to a pseudosphere if $K < 0$. If $\Phi$ is simply connected, we have $\dim \mathcal{K}_\Phi = 3$, see Fig. 8.

One should be aware that topological features of surfaces typically restrict existence of Killing fields. To obtain local information it is therefore advisable to make surfaces simply connected by cutting them open before computing Killings fields on them. We are going to use Killing vector fields to find an isometry or near-isometry of a surface to a surface of revolution, thus enabling paneling with a small number of molds (see §5.1.2 and Fig. 10).

3.2.2 Computing Near-Killing Vector Fields. Vector fields can be approximately Killing in different ways: An i-vector field might be only approximately tangential; or a tangential vector field might satisfy the constraints (4) only approximately. The following optimization procedure provides access to both versions.

Assume the surface of interest is given by a quad mesh $(V, E, F)$, and in each vertex $v_i$, $i = 1, \ldots, |V|$ a unit normal vector $n_i$ has been estimated. We let $n = (n_1, n_2, \ldots) \in \mathbb{R}^{|V|}$. A Killing vector field $w = (w_1, w_2, \ldots) \in \mathbb{R}^{|V|}$ consists of a vector $w_i$ per vertex. It is subject to three constraints: Firstly, $w$ is tangential, i.e.,

$$E_{\text{orth}}(w) = \sum_i (w_i, n_i)^2 = w^T (nn^T) w = 0.$$ 

Secondly, it obeys the orthogonality constraint (4), which can be written as $Cw = 0$, with a matrix $C \in \mathbb{R}^{3|F|\times|V|}$. Equivalently,

$$E_{\text{orth}}(w) = (Cw)^T (Cw) = 0.$$ 

Thirdly, a fairness energy of the form (3) is used, which is quadratic and can therefore be expressed as $E_{\text{fair}}(w) = w^T M w$. Summing up, we require $w$ to minimize $E_{\text{Killing}} = E_{\text{tang}} + \lambda_{\text{orth}} E_{\text{orth}} + \lambda_{\text{fair}} E_{\text{fair}}$.

$$E_{\text{Killing}}(w) = w^T K w \rightarrow \min, \quad K = nn^T + \lambda_{\text{orth}} C^T C + \lambda_{\text{fair}} M, \quad (7)$$

under the normalization constraint $\|w\| = 1$. The parameter $\lambda_{\text{orth}}$ is used to adjust the relative weight of constraints, while $\lambda_{\text{fair}}$ is small. The solution consists of an eigenvector of $K$ corresponding to the smallest eigenvalue. This procedure is very similar to the computation of r-vector fields by [Pottmann et al. 2001]. The parameter $\lambda_{\text{orth}}$ in (7) determines whether we want more emphasis on the tangential property or the i-vector field property. §5.1.2 treats the more difficult problem of modifying a mesh so that it has an exact Killing field.

3.2.3 Computing the Dimension of $\mathcal{K}_\Phi$. It may happen that the matrix $K$ of (7) has more than one small eigenvalue. The meaning of ‘small’ has to be clarified: For any normalized eigenvector $w$ corresponding to the eigenvalue $\omega$ we have $E_{\text{Killing}}(w) = w^T K w = \omega w^T w = \omega$. So $\omega$ is small if, seen as an energy expressing the Killing property, it is small. The occurrence of two or three such small eigenvalues is evidence for $\dim \mathcal{K}_\Phi \geq 2$, which can only happen if the Gaussian curvature is constant. Our numerical experiments confirmed this relation. Figures 7 and 10 show the case of a single smallest eigenvalue, while the example of Fig. 8 has three.

Remark 4. There is an alternative to the energy defined by (7) that does not require estimating normal vectors. Suppose $f = (v_p, v_q, v_r, v_s)$ is a face, and the inscribed parallelogram has normal vector $n_f$ which is thought to be valid for the center $m_f = \frac{1}{4} (v_p + v_q + v_r + v_s)$. The vector field assigns to $m_f$ the vector $\frac{1}{2}(w_p + w_q + w_r + w_s)$, and tangency is expressed by $E_{\text{tang}} = \sum_f (w_p + w_q + w_r + w_s)^2 \rightarrow \min$. This modified energy $E_{\text{tang}}$ yields a modified energy $E_{\text{Killing}}$.

4 ISOMETRIC REGISTRATION

For the purpose of paneling, we treat the following problem. Given are two surfaces $\Phi_1$ (a panel) and $\Phi_2$ (the mold as target surface). Isometrically deform $\Phi_1$ so that the resulting deformed surface $\Phi_1'$ matches $\Phi_2$. We do not require alignment of boundaries here. This process is called nonrigid, isometric registration. Algorithmically, one has to solve for an initial alignment first — this is a global registration problem and will depend on the specific application. In a second step one fine-tunes the alignment by numerical optimization — this is a local registration problem.

For local registration, we will be able to work within the framework of i-vector fields presented in §3. The surface $\Phi_1$ is represented as a quad mesh $M_1$, while $\Phi_2$ is represented in any way which allows us to compute closest points and distances.

The Distance Field of a Surface. We here build on prior work by Pottmann et al. [2006] on rigid registration. For simplifying the distance field of the target surface, we use the following property: If a point $y$ is not far from its closest point $y^* \in \Phi_2$ (not far compared to curvature radii of $\Phi_2$), then locally around $y$ we may replace the squared distance from the target by the squared distance from the
Fig. 9. Paneling with surfaces of constant Gaussian curvature $K$. 5 positive curvature molds corresponding to red panels and negative curvature molds corresponding to blue panels are used. The remaining panels are developable and correspond to one single flat virtual mold. The color coding indicates the value of Gaussian curvature. Subfigures (b) and (c) show actual architectural freeform skins, namely a 2009 glass roof by Asymptote Architecture in Abu Dhabi and a design by Zaha Hadid architects realized 2012 in concrete in Baku. See detailed statistics in Fig. 17.

The local registration procedure itself is based on iterative optimization. Its aim is to find a mesh $M'_1$ isometric to the given mesh $M_1$ (the panel), but close to the target $\Phi_2$ (the mold). We even allow the panel to change a little in the process so as to better fit the mold. Conceptually, we linearize the evolution of the mesh by computing an i-vector field and move vertices in this direction. Isometry is destroyed by this procedure (but not by much) and has to be restored using optimization. These two steps are combined into one, as described.

Registration by Optimization. The local registration procedure is used during iterative optimization, closest points and normal vectors are recomputed before every round of optimization.

Registration by Optimization. The local registration procedure is based on iterative optimization. Its aim is to find a mesh $M'_1$ isometric to the given mesh $M_1$ (the panel), but close to the target $\Phi_2$ (the mold). We point out that the well known ICP algorithm works on the basis $d(x, \Phi_2) \approx ||x - y'||$, which is a good approximation only far away from the target [Pottmann et al. 2006].

We initialize by $M'_1 = M_1$ and update $M'_1$ in every round of optimization. Collecting the vertices of $M'_1$ in $v' = (v'_1, \ldots) \in \mathbb{R}^{3|V|}$, the update is via $v' \leftarrow v' + w'$, where $w'$ is an i-vector field minimizing the energy $E_{orth}$ according to Eq. (6). This vector field linearizes the isometric evolution of $M'_1$ over time.

Based on (8), proximity of the updated mesh $M'_1$ to the target is governed by the penalty term

$$E_{close} = \sum_i \left( v'_i + w'_i - v'_i, n'_i \right)^2 + \varepsilon \sum ||v'_i + w'_i - v'_i||^2.$$  \hfill (9)

It is formulated in terms of closest point projections $v'_i$ of vertices $v'_i$ onto the target. It includes a regularizer with small weight $\varepsilon$ which might not be necessary, depending on the application. If the original panel $M_1$ is also a variable, an analogous energy $E_{close}$ penalizes deviation of $M_1$ from the reference surface $\Phi_1$.

The energy $E_{iso}$ of (2) expresses isometry of $M_1$ and the updated mesh $M'_1$, if applied to $v' + w'$. Combining the constraints listed so far with fairness, we create an energy which is to be minimized:

$$E_{reg} = \lambda_{iso} E_{iso}(v' + w') + \lambda_{close} E_{close} + \lambda_{mold} E_{mold} + \lambda_{orth} E_{orth}(w') + \lambda_{fair} E_{fair} \rightarrow \min .$$  \hfill (10)

The variables in this optimization are the vector field $w'$ and, depending on the application, also the vertices of the original panel $M_1$. Fairness applies to all meshes that are variables, namely $M'_1$ and possibly also $M_1$. Therefore we do not use the fairness energy $E_{fair}$ (which is based on a comparison of these two meshes), but stick with $E_{fair}$.

After updating we repeat until either the optimization goal or the maximum number of iterations has been reached. A summary of the procedure is given by Alg. 1. We found that for our main application (paneling) it was not necessary to change the weights occurring in (10).

Isometric Registration up to Scale. It turned out that a very successful method of paneling is to use surfaces of constant Gaussian curvature $K$, see e.g. Fig. 9. We found it useful to have at our disposal a way to match a given surface patch with a sphere or pseudosphere, but where the value of $K$ is still a variable. This is done by deciding the sign of $K$ first, and subsequently registering to the unit sphere, or unit pseudosphere, in a way that is isometric up to scale.

Algorithm 1: Isometrically register mesh $M_1$ to target surface $\Phi_2$.

Data: Mesh $M_1$ with vertices $v$
$\mathbf{v'} \leftarrow v'$;
Fix energy threshold $E_{reg,min}$ and maximum iterations;
Choose weights $\varepsilon$, $\lambda_{orth}$, $\lambda_{fair}$, $\lambda_{close}$, $\lambda_{mold}$ for (9) and (10);
repeat
For all vertices $v_i$, compute closest point $v'_i \in \Phi_2$;
Estimate unit normal vectors $n'_i = (v_i - v'_i) / ||v_i - v'_i||$
Determine $v$, $w'$ such that $E_{reg} \rightarrow \min$;
Update $v' \leftarrow v' + w'$;
Update weights;
until $E_{reg} \leq E_{reg,min}$ or max. iterations reached;
return Vertex coordinates $v$, $v'$.
For that we modify Alg. 1, by replacing the energy $E_{iso}$ by $E'_{iso} = \sum_f \sum_{j=0}^c c_{iso,j}(f)$. The constraints $c_{iso,j}$ are based on the isometry constraints (1) and express isometry up to scaling by a factor $\kappa$:

\[
\begin{align*}
  c_{iso,0}(f) &= \kappa^2 ||v_0 - v_2||^2 - ||v'_0 - v'_2||^2 = 0, \\
  c_{iso,1}(f) &= \kappa^2 ||v_1 - v_3||^2 - ||v'_1 - v'_3||^2 = 0, \\
  c_{iso,2}(f) &= \kappa^2 (v_0 - v_2, v_1 - v_3) - (v'_0 - v'_2, v'_1 - v'_3) = 0.
\end{align*}
\]

Since $\kappa^2$ enters these constraints linearly, we use $\kappa^2$ as a variable. If meshes $M, M'$ obey these constraints and $M$ represents the unit sphere, then $M'$ is isometric to a sphere of radius $1/\kappa$ and enjoys Gaussian curvature $K = \kappa^2$. In the pseudosphere case, $K = -\kappa^2$.

5 APPLICATIONS

5.1 Isometric Paneling

The main motivation for our work is paneling of architectural freeform skins by materials that allow isometric bending (sheet metal, certain plastics, and textiles). We start from a given panel layout on a surface and compute a set of molds for it, so that this set is as small as possible and the predefined panels can be manufactured, up to tolerances, by these molds. Quality criteria include deviation from the reference surface, and gaps and kink angles at seams. A second task is to guide the panel layout by geometric information. Note however that the panel layout, if at all visible on the finished architectural design, in a real-world application is going to be a design decision.

5.1.1 Paneling Algorithm for Constant Gaussian Curvature Molds.

In the course of our work on the paneling problem we were surprised to find out to which extent it is possible to achieve a smooth paneling of freeform shapes by surfaces of constant Gaussian curvature. Figures 1, 9a–c, 15, and 19 show results obtained by this method. The panelization procedure starts with a given reference surface plus panel boundary layout, and it consists of the following steps.

- **Intrinsic patch analysis.** For this application, this means computing Gaussian curvature $K$, for which we use the method of osculating jets [Cazals and Pouget 2003].
- **Clustering.** This means assigning a type of mold to each panel. Panels where the Gaussian curvature does not exceed a threshold are declared intrinsically flat, exhibiting zero Gaussian curvature. On the other panels we employ $k$-means clustering on the average value of $K$ of the panel, using the algorithm of [Lloyd 1982]. This leads to a small number of clusters centered around values $K_j$ (we simply chose to use 5 clusters). $K_j$ is the Gaussian curvature of the mold assigned to all panels in that cluster.
- **Construction of molds.** For each nonzero value $K_j$ we construct a mold by scaling the unit sphere resp. pseudosphere with the factor $|K_j|^{-1/2}$. It is very convenient to have spheres as molds since the distance field is known. Similarly, the pseudospheres have rotational symmetry and their distance field is manageable. The third kind of mold is a flat plane, it is associated with zero Gaussian curvature.
- **Local panel optimization.** For a first alignment between panels and molds, rigidly move molds of Gaussian curvature $K_j$ to each panel contained in the corresponding cluster. This global registration task in case $K_j > 0$ consists of a standard least square fitting of a sphere of known radius to the panel. In case $K_j = 0$ we use PCA to fit a plane to the panel. In case $K_j < 0$ we find a pseudospherical piece of the same size as the panel and likewise use PCA for the first alignment. We subsequently for each panel $M_j$ find an isometric panel $M'_j$ that is close to the mold which currently is positioned close to $M_j$. This local registration procedure is initialized as $M'_j = M_j$ and is performed according to §4. In order to achieve a better fit between the design surface and the molds we extend the registration procedure by considering also the vertices of the original panels as variables except for the boundaries which are kept fixed.
- **Final optimization.** While this was not necessary for most of our examples, in order to achieve a smoother appearance (smaller kink angles at panel boundaries) one can perform a few steps of optimization of several panels (or even all panels) simultaneously. The target functional of optimization is the sum of the ones used in the local panel optimization above, cf. Fig. 14.

5.1.2 Paneling Algorithm Based on Rotational Surfaces. Paneling a surface of revolution $\Psi$ needs only a small number of molds – choose molds as pieces of $\Psi$ along a meridian and ensure that each panel can be rotated into at least one mold (see Fig. 10d). Any surface $\Phi$ admitting a near-Killing vector field can be modified in small ways to become isometric to a surface of revolution $\Psi$, so also $\Phi$ has a paneling that needs only a small number of molds. Figure 10 illustrates the procedure we apply to achieve this modification.

We start with the cross field determined by the near-Killing vector field and vectors orthogonal to it. We use the libigl implementation of mixed-integer quadrangulation [Bommes et al. 2009; Jacobson et al. 2018] to find a quad mesh $M$ aligned with this cross field, see Fig. 10b. $M$ becomes exactly isometric to a rotational surface, if any pair of faces that are neighbours in the Killing field direction become
isometric. These constraints are expressed analogous to Equ. (1) and are met by way of optimization. The result is a mesh $M'$ (Fig. 10c) isometric to a mesh $M''$ with rotational symmetry (Fig. 10d). The construction of $M''$ is elementary and is described e.g. by [Wang et al. 2019]; it is unique after choice of an initial value.

5.2 Designing and Actuating Isometries

5.2.1 Flexible Quad Meshes with Planar Faces. Certain special quad meshes with planar faces admit flexions where all faces move as rigid bodies and edges act as hinges. They have been a topic of interest already in the earlier works on discrete differential geometry [Sauer 1970]. In case of regular combinatorics, counting degrees of freedom reveals that flexibility is equivalent to flexibility of each 3 meshes is rather elementary and is described e.g. by [Wang 2016].

Remark 5. An easily understood class of flexible quad meshes are discrete Voss surfaces where mesh polylines enjoy a discrete geodesic property [Sauer 1970]. In each vertex the four angles $\omega_1, \ldots, \omega_4$ of faces obey $\omega_1 - \omega_3 = \omega_2 - \omega_4 = 0$. For Fig. 11, average and maximum of these differences equal 1.3° resp 10°. This example therefore is not a Voss surface.

Remark 6. When trying to understand the isometric bending $\psi$ of a surface $\Phi$ into another shape $\Phi' = \psi(\Phi)$, we might ask for directions where the normal curvature changes the most, or changes the least. This normal curvature in direction of a unit tangent vector is expressed as $\Pi(\nu, v)$. With $\nu' = d\psi(\nu)$ the normal curvature after bending is $\Pi'(\nu', v')$. When we ask for extremal values of the quotient, we are led to the very same generalized eigenvalue problem as above. Fig. 12 shows an example. Note that asking for extremal values does not make sense in case of negative Gaussian curvature, because if $\nu'$ points in an asymptotic direction, we have $\Pi'(\nu', v') = 0$ and the ratio will be infinite [do Carmo 1976, p. 148ff].

5.2.2 Isometric Bending Guided by Trajectories. Figure 6 shows the time-continuous isometric bending of a surface such that selected vertices $v_{ik}, k = 1, \ldots, r$ run on prescribed trajectories $T_{ik}$. This is simulated by computing an i-vector field $w_{ik}$ where $w_{ik}$ is tangential to $T_{ik}$, updating vertices by $v_{i} \leftarrow v_{i} + w_{i}$ and restoring isometry to the mesh we started with by invoking optimization according to §2. The movement of vertices along the prescribed trajectories is ensured by penalizing distances by an energy $E_{close}$ analogous to (9). This small example illustrates the many degrees of freedom that are present in isometric bending.

5.2.3 Continuous Folding of Doubly-Curved Surfaces. Geometric folding has produced a vast amount of literature. Specifically the folding of flat sheets along curves has been treated from the viewpoint of geometry processing only more recently. Double-curved smooth surfaces likewise admit isometric deformations to surfaces.
that are only piecewise smooth. Interestingly, time-continuous deformations of this kind are rather restricted. The reason for this is a well known geometric property occurring at the foldline. The foldline’s osculating plane must be a local supporting plane [Rabinovich et al. 2019], which in the moment we start folding, where the surface is still smooth, means that the osculating plane equals the tangent plane. Unfortunately in non-flat surfaces such curves are rare. In case Gaussian curvature $K$ is positive there are none, for developable surfaces there are only the rulings. In the case $K < 0$ we have the network of asymptotic curves, with two curves passing through each point. Curved folding is only possible along these asymptotic curves. Fig. 13 shows an example. We simulated folding by computing isometries under the side-condition that two selected points move towards each other, while not enforcing fairness along selected foldlines.

6 DISCUSSION

Implementation and Properties of the Optimization. For optimization, we employ a Levenberg-Marquardt method according to [Madsen et al. 2004, §3.2], using a damping parameter of $10^{-6}$, but we could just as well have used other optimization methods. Our implementation in C++ uses the data structures of OpenMesh [Botsch et al. 2002] and the Taucs library for sparse linear solvers [Toledo 2003]. Detailed statistics on panelization examples are provided by the table in Fig. 17. The computation times refer to an Intel Xeon E5-2687W 3.0GHz processor without parallel processing or other acceleration techniques. The result can be summarized by saying that registering a single panel with 1024 vertices takes about 0.3 to 0.4 seconds. In comparison, computing an isometric mapping according to §2 takes only about 20% of that time.

The panelization procedure is essentially local; each panel is registered and optimized to a mold separately. We found that we can achieve a surface quality sufficient for applications in this way. However, a final round of global optimization, as described in the last paragraph of §5.1.1, can still improve surface quality. Fig. 14 shows an example where panels are optimized together with their immediate neighbourhood. Such a procedure is not difficult but rather time-consuming.

Convergence. We also want to add a comment on convergence. Convergence is not guaranteed, as is usual in nonlinear and nonconvex situations like ours, so a good initialization is very important. Our experiments revealed the optimization problems we solved as rather benign, with typically 5-10 iterations necessary to achieve convergence.

As to the order of convergence, in particular of optimization of isometries according to §2, we replaced the energy $E_{\text{fair}}$ by $E_{\text{tang}}$ not just because of geometric considerations. This was done also because we were concerned about $E_{\text{fair}}$ witnessing extrinsic curvature and thus never achieving zero residual. $E_{\text{tang}}$ can do that but in practice does not. In consequence, the order of convergence was not improved from one to two by switching to $E_{\text{tang}}$. This switch nevertheless improves the behaviour of optimization. This is because
$E_{\text{tang}}$ regularizes like $E_{\text{fair}}$ while having significantly smaller value, thus not dominating other energies as easily, see Fig. 16.

**Accuracy.** We distinguish two kinds of accuracy, one being the quality of approximation of a reference surface by the paneling. It is expressed by the distances $\delta_{\text{avg}}, \delta_{\text{max}}$, and the number of molds. The number of iterations of our optimization procedure is 10 throughout, the time (in seconds) is shown in the table. The data verifying the quality of the result are the distance between patches and design surface, the distance between isometrically deformed patches and mold surface. Distances are normalized such that a patch bounding box has length 1 in average. The last number is the $L^2$ error of edge lengths, defined as $\|L'-L\|_2/|L|_2$, where $L$, $L'$ refer to the vector of all edge lengths occurring in panels, resp. deformed panels.

**Practical Feasibility.** The materials and processes employed in freeform architecture do not scale well to the kind of small models we are able to build for experimental verification. We nevertheless checked the feasibility of our paneling procedure by means of molds created by CNC machining and panels made from high impact polystyrene. Molds and panels are shown by Figures 18, 19.

**Limitations.** Our method of representing isometries has some limitations. E.g. the importance of fairness makes it unsuitable for modeling wrinkles and creases that might occur in isometric bending. This might lead to unrealistic behaviour in applications like the one shown in Fig. 6. However this is not an issue with paneling.

**Future Research.** Several directions of research can build on our work. Especially interesting are those having to do with continuous flexions. Flexible meshes with rigid faces have been a source of deep Mathematics and also actual transformable designs, but have hardly been explored algorithmically. Also mechanisms consisting of nonrigid parts are an interesting direction, maybe in connection with bionic technology. We may even go one step further and leave the domain of isometric mappings. mappings which obey constraints, e.g. on length distortion, have already been noticed to play a role in our understanding of axiety and other nonconforming micromaterials which are actually mechanisms. Mappings subject to length-related constraints are obviously relevant for the modeling of the behaviour of more general sheet-like materials. We believe that in this area, well-informed geometric modeling can significantly speed up simulations.

**ACKNOWLEDGMENTS**

This work was supported by the Austrian Science Fund via grants I2978 (SFB-Transregio programme Discretization in geometry and dynamics), W1230 (DK programme Discrete Mathematics) and F77 (SFB grant Advanced Computational Design). V. Ceballos, C. Jiang, F. Rist, and H. Wang were supported by KAUST baseline funding.

**REFERENCES**


ACM Trans. Graph., Vol. 40, No. 4, Article 42. Publication date: August 2021.


Sivan Toledo. 2003. Taucs, A Library of Sparse Linear Solvers. Tel Aviv University. www.tau.ac.il/~stoledo/taucs


Fig. 19. Paneling free forms with an isometrically deforming material. This is an actual model built with high impact polystyrene panels. For the design surface and a photo of the molds see Fig. 18.