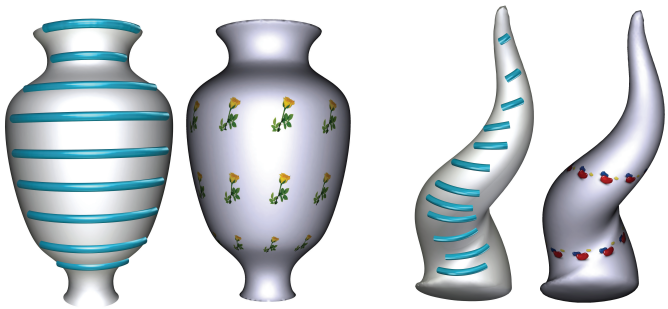
Eurographics Symposium on Geometry Processing 2010
Olga Sorkine and Bruno Lévy
(Guest Editors)

Volume 29 (2010), Number 5

On Discrete Killing Vector Fields and Patterns on Surfaces

Mirela Ben-Chen Adrian Butscher Justin Solomon Leonidas Guibas

Stanford University



Abstract

Symmetry is one of the most important properties of a shape, unifying form and function. It encodes semantic information on one hand, and affects the shape's aesthetic value on the other. Symmetry comes in many flavors, amongst the most interesting being intrinsic symmetry, which is defined only in terms of the intrinsic geometry of the shape. Continuous intrinsic symmetries can be represented using infinitesimal rigid transformations, which are given as tangent vector fields on the surface – known as Killing Vector Fields. As exact symmetries are quite rare, especially when considering noisy sampled surfaces, we propose a method for relaxing the exact symmetry constraint to allow for approximate symmetries and approximate Killing Vector Fields, and show how to discretize these concepts for generating such vector fields on a triangulated mesh. We discuss the properties of approximate Killing Vector Fields, and propose an application to utilize them for texture and geometry synthesis.

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

1. Introduction

Symmetries and symmetric patterns have always fascinated artists and researchers alike, intrigued by the effect they have on our perception of beauty and by the beauty of the underlying mathematical concepts. As the virtual worlds we create mimic our own, the need arises for simple methods for generating symmetric models decorated by symmetric patterns and for automatic methods for extracting such features from existing shapes.

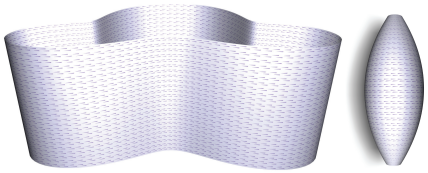
Symmetry can be defined as a structure-preserving transformation from a shape to itself, and we will focus only distance-preserving symmetries. For example, a cylinder has rotational symmetry, since it does not change when rotating around its axis. This is an example of an *extrinsic* symmetry, inherited from the embedding space, as the transformation we applied to the cylinder was defined in $\mathbb{R}^3$. In addition, it is a *continuous* symmetry, as we can rotate the cylinder by any angle. If we endow our shape with more structure, some symmetry is lost. For example, by coloring the cylinder, as in Fig 1(a), the possible transformations which will result in the same shape are only rotations by multiples of $\pi/4$, generating a *discrete* symmetry. A composition of two symmetric transformations is again a symmetric transformation, thus symmetries form a group under composition known as the *symmetry group*. Extrinsic symmetries are well-understood, and many algorithms exist for finding such symmetries in images (see a recent review in [PLC*08]) and some in 3D shapes [PMW*08, BBW*09]. More challenging are *intrinsic* symmetries. Consider for example the shape in Fig 1(b). It is intuitively clear to the human observer that this shape is not substantially different from the colored cylinder, and that there should be a similar notion of symmetric "transformations". However, in this case the symmetry is *intrinsic* to the shape, and not inherited from the embedding space,

hence there is no global rigid transformation which can represent the symmetries of this object. As a result, extrinsic methods for detecting patterns in 3D shapes, such as [PMW*08], are not suitable for this case.

(*a*) (*b*)

Figure 1: *Examples of extrinsic (a) and intrinsic (b) discrete symmetries.*

An alternative way of representing a continuous transformation of a surface is using a tangent vector field: at each point on the surface we are given a velocity vector, and the point moves an infinitesimal amount in the given direction with the given speed. If the geodesic distances between all pairs of points are preserved under the transformation, then the vector field generating this transformation is called a *Killing vector field* (KVF). Fig 2 shows two examples of such vector fields. We show one vector per face, represented using a small arrow whose length is proportional to the norm of the vector. Such vector fields are *intrinsic*, hence the shapes in Fig 1(a) and in Fig 1(b) have the same set of KVFs. Note how the norm of the vector field is larger towards the center of the shape in Fig 2(b), implying points will have to move at a greater speed there as compared to points at the extremities, in order to preserve the geodesic distance between them.



(*a*) (*b*)

Figure 2: *Examples of Killing Vector Fields on simple surfaces. The norm of the vector is important, as it indicates the speed of the movement.*

As KVFs generate intrinsic infinitesimal isometries, they potentially can be used as the underlying mathematical machinery for studying symmetries and symmetric patterns on surfaces. However, exactly symmetric surfaces are quite rare, even more so in the context of noisy 3D models. In fact, it has been known since the 1930's [Mye36] that the only orientable two-manifolds possessing global continuous symmetries are homeomorphic to the sphere, the projective plane, the ordinary plane, the cylinder (and thus also the cone), and the torus. This shows that the existence of a global continuous symmetry is indeed something rather special, if one considers the space of all two-manifolds. On the other hand, in terms of the actual objects that occur in our 3-D world, both natural and man-made, it is almost universal that they possess pieces that are near isometric deformations of planes, spheres, cylinders, cones, tori, etc. Thus, if we can relax the notion of intrinsic symmetries, to allow for *approximate* symmetries, we could potentially detect approximate symmetries in many (parts of) common 3-D models. We show how to relax the symmetry requirement, by reformulating the definition of KVFs as a variational problem, thus allowing for *approximate Killing vector fields* when no exact KVFs exists. Moreover, we show how to define and find *discrete approximate Killing vector fields* on triangular meshes, using a simple operator defined in terms of Discrete Exterior Calculus. Finally, we demonstrate how discrete approximate KVFs can be used to easily generate patterns on simple surfaces.

1.1. Previous work and overview

Symmetry detection and symmetric pattern generation are well researched subjects, and a thorough review of these topics is beyond the scope of this paper. We will thus concentrate on work most relevant to our approach – in the area of Killing vector fields, and symmetries and patterns on surfaces.

Killing vector fields appear scarcely in the geometry processing literature. As KVFs are tightly connected to isometric deformations, they were first discussed in a modeling paper [KMP07], where they were used for motivating an isometry-preserving deformation method. The paper, however, did not describe how to explicitly find KVFs given a triangular mesh, nor did it consider approximate KVFs. In a completely different context, KVFs were used in [GMDW09] to simplify visualization of concepts from general relativity. They also do not consider approximate KVFs.

In the area of general relativity, KVFs are commonly used as a means for finding symmetries of space-time, as such symmetries can aid in finding exact solutions of Einstein's field equations [Hal04]. Furthermore, approximate symmetries and approximate KVFs are also of interest in that field, see [Zal99] for a review on different definitions of approximate symmetries. The approach closest to ours is the one first suggested by [Mat68] and later re-introduced by [Bee08]. In these papers the authors suggest to find approximate KVFs as solutions of an eigenvalue problem, similarly to the method we propose. The context however is again very different, and the paper did not provide computational results.

In the realm of geometry processing, symmetry is mostly discussed in relation to symmetry extraction and pattern generation. Many solutions have been proposed for extracting patterns and symmetries from three dimensional models. In general, such methods can be divided into *extrinsic* and *intrinsic*. Extrinsic methods, such as [MGP06, PMW*08, BBW*09], utilize the symmetries of the ambient Euclidean space for finding patterns in 3D models. Such methods, while robust, are somewhat restricted, as they cannot recover symmetries and patterns which are not inherited from the embedding space, such as the symmetries of the shape in Fig 1(b). Intrinsic methods, on the other hand, are able to find intrinsic symmetries [OSG08, LTSW09, XZT*09, RBBK10]; however previous methods have focused on discrete symmetries, whereas we consider continuous symmetries. In addition, in the special case that the surface is indeed symmetric, our formulation can be used

for defining an *intrinsic symmetry group* without using the permutation group as in [RBBK10].

Pattern generation is relatively less researched than symmetry extraction. One approach for generating patterns on general surfaces, proposed by [Kap09], is based on tiling a simple domain to which the surface is conformally mapped. Other approaches use texture synthesis techniques for generating semi-regular patterns [ZHW*06, ACXG09]. Finally, [LFZ*10] use a guiding vector field and a grammar for pattern generation. We show in the applications section how one can benefit from using our discrete KVFs to drive a vector-based geometry synthesis method, such as [LFZ*10].

Before diving into the details, we will give a brief overview of our approach. Our main goal is to generate a family of special vector fields given a triangular mesh. These vector fields are the generators of a family of continuous deformations, which are close to intrinsic isometries. First we will repeat some known definitions and theorems to introduce Killing vector fields on smooth surfaces. Then we depart from classic results and extend the definition to allow for approximate KVFs (which we will henceforth refer to as AKVFs), by solving an eigenvalue problem. We proceed by showing how using a simple manipulation AKVFs can be represented as the eigenvectors of an Exterior Calculus operator, which can then be discretized using the existing machinery of Discrete Exterior Calculus. We conclude by showing a possible application for these vector fields.

2. Killing Vector Fields

Killing vector fields and infinitesimal deformations are well known in differential geometry and are widely used in general relativity. For completeness, we will present an intuitive introduction to these concepts, and refer the interested reader to [Pet97, Wal84] for a more detailed treatment.

2.1 Infinitesimal Deformations

A shape is symmetric if it is invariant under a distance-preserving self-mapping. Hence, classifying symmetries is closely related to the parameterization of all possible self-mappings of a shape. When dealing with symmetries of Euclidean space, these mappings are easily defined through global linear transformations. For example, as discussed previously, the cylinder in Fig 1(a) is invariant under rotations around its axis. However, such an approach is not appropriate for intrinsic symmetries, as there might not exist a global linear transformation mapping the surface to itself. Thus, we need an alternative for specifying intrinsic self-mappings of a surface. We will first show how to define the space of continuous self-mappings, and then restrict them to distance preserving ones.

We propose to use *infinitesimal deformations* to represent continuous self-mappings. Intuitively, at each point p on the surface, we prescribe a tangent vector $U(p)$, and treat it as a velocity vector. To find where a point p is mapped, we follow the flow line of the velocity field U, starting at the point p. The amount of "time" t we follow the flow line yields a family of self-mappings ϕ^t (see Fig. 3). This leads to the following definition.

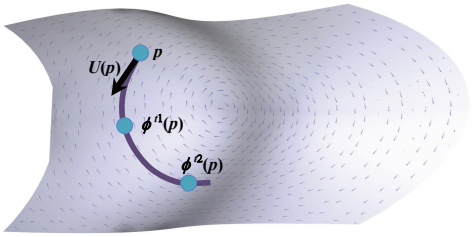


Figure 3: *The mapping of a point p under the deformations ϕ^{t1} and ϕ^{t2}, both belonging to the family of deformations ϕ^t generated by the tangent vector field U. The curve shown is a part of the integral curve of U starting at p.*

***Definition 1*:**

Given a two-manifold M, a smooth tangent vector field U, and $t \in (-\varepsilon, \varepsilon)$ for some $\varepsilon > 0$, the *deformation* generated by U is denoted $\phi^t : M \to M$ and is defined as:

$$\phi^t(p) = \gamma_{p,U}(t) \qquad p \in M$$

where $\gamma_{p,U}(t)$ is the solution of the initial value problem:

$$\gamma_{p,U}(0) = p, \quad \dot{\gamma}_{p,U}(t) = U\left(\gamma(t)\right) \tag{1}$$

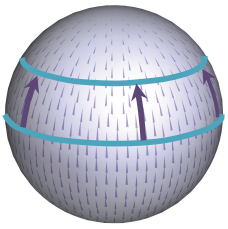
and $\dot{\gamma}(t)$ is the tangent vector of the curve γ at $\gamma(t)$.

It is well known from the theory of ordinary differential equations, that given continuity conditions on U, the solution of (1) exists and is unique. The curves γ are called the *integral curves*, or the *flow lines*, of the vector field U.

As is the case with Euclidean linear transformations, infinitesimal deformations form a group, with composition as the group action. Specifically, it is easy to show [Boo02, Chapter 4] that:

$$\phi^t\left(\phi^s(p)\right) = \phi^{s+t}(p) \quad \text{and} \quad \phi^0(p) = p.$$

As a simple example of a vector field that generates a family of mappings, consider the vector field: $U(p = x, y, z) = (0,0,1)$ projected onto the unit sphere. This vector field represents the mappings ϕ^t which map the curves $z = c$ to the curves $z = c + f(t)$ for some function f (see the inset figure). Note that this is not a distance preserving mapping, as all points get closer to the north pole of the sphere after the mapping.



2.2 Killing Vector Fields

Now that we have a definition for the mappings, we would like to characterize the distance-preserving ones. As these mappings are given in terms of tangent vectors fields, we can express our problem as follows: which are the vector fields U such that the infinitesimal deformations generated by U are distance-preserving. These vector fields are called *Killing Vector Fields* [Pet97, Chapter 7].

To explain the properties of KVFs, we will need a few concepts from Riemannian geometry. As these definitions are somewhat lengthy, we will only give intuition on their geometric meaning; see [doC92] for a full exposition.

Given a two dimensional manifold M, a *metric* provides a way of measuring angles and lengths on M as follows. Given a point $p \in M$, the tangent plane to M at p is denoted by T_pM. The metric g takes two vectors $X,Y \in T_pM$, and returns a real number $g_p(X,Y)$. Lengths of curves on the surface are defined using the metric. Given a curve $\gamma: I \subset R \to M$, its length is $L(\gamma) = \int_I \sqrt{g_{\gamma(t)}(\dot{\gamma}(t),\dot{\gamma}(t))}dt$. The geodesic distance between two points on the surface is the length of the shortest curve between them. Hence, to find a distance-preserving mapping, we need to find deformations which preserve the metric.

The metric g is defined by its action on vectors in the tangent plane, so first we must explain how the deformation ϕ affects tangent vectors. Given a point $p \in M$ and a tangent vector $X \in T_pM$, there is a unique curve $c(t)$, such that $c(0) = p$, and $\dot{c}(0) = X$. The deformation ϕ maps the curve c to a new curve γ via $\gamma(t) = \phi(c(t))$, and thus maps X to $\dot{\gamma}(0)$, the tangent vector of γ at $\phi(p)$. Hence, for any smooth map $\phi: M \to M$, we can define the *differential* of ϕ at a point $p \in M$ as the mapping $d\phi_p: T_pM \to T_{\phi(p)}M$ which performs this action. The differential is the linear map which prescribes the way tangent vectors change. The application of $d\phi$ to a vector X is also called the *pushforward* of X by ϕ. See Fig 4 for the notations.

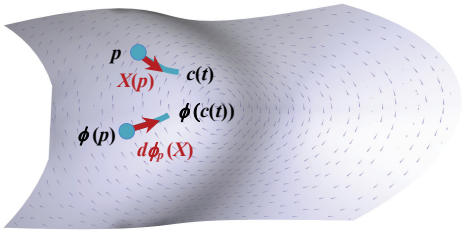


Figure 4: *The differential of ϕ at p maps vectors in the tangent plane of p to vectors in the tangent plane of $\phi(p)$. $d\phi_p(X)$ is called the pushforward of X.*

Equipped with these definitions, we can finally state the condition that a vector field U is a Killing vector field.

***Definition 2*:**

Given a two-manifold M, a smooth tangent vector field U, and $t \in (-\varepsilon, \varepsilon)$ for some $\varepsilon > 0$ then U is a *Killing vector field* if and only if for any $p \in M$ and $X,Y \in T_pM$ we have:

$$g_p(X,Y) = g_{\phi(p)}(d\phi_p(X), d\phi_p(Y)),$$

where $\phi = \phi^t$ is the deformation generated by U.

Since this is true for any t, we can take the limit and get an equivalent definition. Thus, U is a KVF if and only if:

$$(2) \qquad \mathcal{L}_U g \equiv \lim_{t \to 0} \frac{g_{\phi^t(p)}\left(d\phi_p^t(X), d\phi_p^t(Y)\right) - g_p(X,Y)}{t} = 0.$$

This expression is known as the *Lie derivative* of g with respect to U, and denoted $\mathcal{L}_U g$. The Lie derivative is a generalization to curved surfaces of the planar directional derivative – it provides the rate of change of quantities on the surface when following the flow of a given vector field. Since KVFs are vector fields whose flows preserve the metric, it is very natural to define KVFs using the Lie derivative. However, to specify in more concise terms the conditions for a vector field to be a KVF (and eventually to define the discretization of these conditions on a triangular mesh), we need an additional type of generalized derivative – the *covariant derivative*.

Before introducing the definition of the covariant derivative, we would like to motivate the discussion using an example from the 2D plane. In 2D, rotations are distance-preserving deformations. For example, for a given point $p = (x_0, y_0)$ consider the curve ϕ^t:

$$\phi^t(x_0, y_0) = (u(t), v(t)) = \begin{pmatrix} \cos(t) & -\sin(t) \\ \sin(t) & \cos(t) \end{pmatrix} \begin{pmatrix} x_0 \\ y_0 \end{pmatrix}.$$

Computing the tangent vector to the curve we get:

$$(u'(t), v'(t)) = \begin{pmatrix} -\sin(t) & -\cos(t) \\ \cos(t) & -\sin(t) \end{pmatrix} \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} = (-v(t), u(t)).$$

Therefore, the deformation ϕ^t is generated by the vector field $U = (-y, x)$, which is thus a KVF. We are interested in the differential properties of U, which we will later mimic on a surface. We can consider the directional derivative of U in the direction of an arbitrary vector V. This is given by $J(U)V$, where $J(U)$ is the Jacobian matrix of U. We have:

$$J(U)V = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} v_x \\ v_y \end{pmatrix} = \begin{pmatrix} -v_y & v_x \end{pmatrix} = R^{90}V.$$

Thus, the Jacobian of U rotates V counter clockwise by $\pi/2$, and we get that for *any* vector V we have:

$$(3) \qquad \langle J(U)V, V \rangle = 0.$$

It is easy to check that (3) also holds trivially for translations, where $\phi^t(p) = p + (t,0)$ and $U = (1,0)$, and therefore J is 0. Note that equation (3) implies that J should be an anti-symmetric matrix. This condition induces an over-determined system of three differential equations ($u_x = 0$, $v_y = 0$, $u_y = -v_x$) in two variables (u and v). Thus, trying to solve this system directly will usually lead to no solution, which is the mathematical reason for the rareness of KVFs on general surfaces. The variational approach we propose later can also be seen as the best solution, in the least squares sense, to the system of equations induced by the anti-symmetry condition.

Equation (3) is usually viewed as the *defining equation* for Killing Vector Fields on curved surfaces, where $J(U)V$ is replaced by the appropriate directional derivative of U in the direction of V on the surface.

A vector field on the surface is given in terms of its components in some local coordinates, which can change from point to point. Thus, it does not make sense to compute the derivative of a vector simply by taking the derivative of its components, since this does not take into account how the

local coordinates change. To be able to take derivatives, we need to prescribe a way to *transport* a vector from a point to a nearby point, and then compare the two vectors in the tangent plane of the second point. The Lie derivative transports vectors using the flow of another vector field, and thus requires some additional information. The *covariant derivative*, on the other hand, transports vectors using parallel transport. Intuitively, a vector is *parallel transported* along a curve if it is dragged along the curve without rotating or stretching. Specifically, if we parallel transport a vector U along a geodesic curve, the length of U and the angle between U and the tangent to the curve is preserved. The covariant derivative is defined as follows.

***Definition 3*:**

Given a two-manifold M, a smooth tangent vector field U, a point $p \in M$, and a vector $V(p) \in T_pM$, the *covariant derivative* of U w.r.t V at p is:

$$(\nabla_V U)(p) = \lim_{t \to 0} \frac{\Gamma(c,t,0)U_{c(t)} - U_{c(0)}}{t}$$

where $c(t)$ is a curve starting at p with $\dot{c}(0) = V$ and $\Gamma(c,a,b)U$ is the parallel transport of the vector U along c from $c(a)$ to $c(b)$ (See Fig. 5). The result does not depend on the choice of the curve c. Note that the covariant derivative of a scalar function $f : M \to R$ is defined similarly:

$$(\nabla_V f)(p) = \lim_{t \to 0} \frac{f(c(t)) - f(c(0))}{t}.$$

But for a scalar function there is no need to use parallel transport, as we can directly compare the values of the function at two points on the surface.

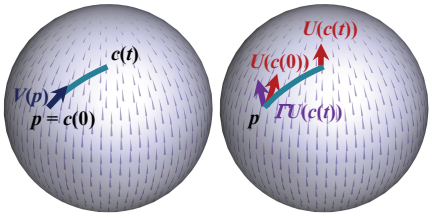


Figure 5: *Notations for the covariant derivative of the vector field U in the direction of the vector V at p.*

Finally, we can formulate a condition guaranteeing that a vector field is a KVF.

***Lemma 1*:**

Given a two-manifold M, a smooth tangent vector field U is a Killing vector field if and only if, for any point $p \in M$ and any vector $V \in T_pM$ we have:

$$\langle \nabla_V U, V \rangle_p = 0 \tag{4}$$

where $\langle , \rangle_p$ is given by the metric g at p.

The proof is based on (2), the definition of the Lie derivative of the metric, and the connection between the Lie and the covariant derivatives. The proof is provided in the supplemental material for completeness, although this is a known result.

Fig. 6 shows a few examples of Killing vector fields on some intrinsically symmetric surfaces. These were computed using the methods discussed in the next section. In addition, the figure shows a pattern generated using such a KVF by applying the methods discussed in Section 4.

Killing vector fields provide a way of describing all possible continuous intrinsic symmetries on a surface. KVFs form a linear subspace of the space of tangent vector fields, as any linear combination of KVFs is also a KVF. This follows directly from (4) and the linearity of the inner product and the covariant derivative. A surface can have at most 3 linearly independent KVFs, and this occurs only on the sphere [Mye36]. Other surfaces have different number of KVFs, depending on the intrinsic symmetries they support. For example, the cylinder has 2, and a "generic" surface of revolution has one.

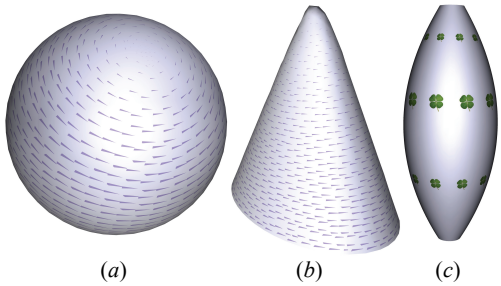


Figure 6: *KVFs on special surfaces which support them – (a) sphere, (b)* cone *with non-circular cross section. (c) a pattern generated using the KVF as explained in Section 4.*

2.3 Approximate Killing Vector Fields

Most surfaces are not *exactly* symmetric, even more so when dealing with noisy sampled surfaces. However, many surfaces do exhibit some kind of approximate symmetry which a human observer will easily notice. We would like to relax condition (4) to allow *Approximate Killing Vector Fields* (AKVF) such that surfaces which are "almost" symmetric will have AKVFs.

To do that, we first rewrite (4) using local coordinates, which are easier to manipulate. Given two vectors $E_1(p)$, $E_2(p)$ which span the tangent plane at the point $p \in M$, the metric g is given by a 2x2 matrix whose entries are $g_{ij} = \langle E_i, E_j \rangle$, where $\langle , \rangle$ is the dot product in R^3. Using the same coordinates, U and V can be written as $U = u^1E_1 + u^2E_2$ and $V = v^1E_1 + v^2E_2$ while $\nabla_V U$ becomes simple matrix multiplication

$$\nabla_V U = (\nabla U)\begin{pmatrix} v^1 \\ v^2 \end{pmatrix} \tag{5}$$

where ∇U is a matrix whose components are $(\nabla U)_{ij} = \nabla_{E_i} u^j \equiv \nabla_i u^j$. These can be computed from the partial derivatives of u^i and the derivatives of the metric g.

Plugging (5) into (4), we get that a vector field U is a KVF if and only if for any vector V we have:

$$\langle \nabla_V U, V \rangle_p = \begin{pmatrix} v^1 & v^2 \end{pmatrix} g \cdot \nabla U \begin{pmatrix} v^1 \\ v^2 \end{pmatrix} = 0$$

Since this should be true for *any* vector $V = (v^1, v^2)$, we get that the matrix $J \equiv g \cdot \nabla U$ is anti-symmetric. Note the similarity to the case of planar rotation, where we found that the Jacobian matrix of U is anti-symmetric. Although we defined J using local coordinates, the anti-symmetry property does not depend on the chosen coordinates – if J is anti-symmetric in one coordinate system, it will be anti-symmetric in *any* coordinate system. This leads us to the following definition.

Definition 4:

Given a two-manifold M with metric g, and a smooth tangent vector field U, the *Killing operator* is the linear differential operator K taking vectors to symmetric tensors, given by $KU := J + J^T$ where $J = g \cdot \nabla U$.

As we have seen, for a Killing vector field U, we have $(KU)(p) = 0$ for all points $p \in M$. To measure how close a vector field is to being a KVF we introduce the Killing energy functional which integrates the square norm of KU (also a coordinate-independent quantity) over the entire surface.

Definition 5:

Given a two-manifold M and a smooth tangent vector field U, the *Killing energy* is:

$$E_K(U) = \int_M \|KU\|^2 \, dv \, .$$

KU is a tensor, and its norm is given with respect to the metric g. Since E_K is positive definite, we have that $E_K(U) = 0$ if and only if U is a KVF. Thus for a surface which has Killing vector fields, they are the minimizers of the Killing energy. More interesting is the situation on non-symmetric surfaces.

Definition 6:

Given a two-manifold M, and a smooth tangent vector field U, then U is *an approximate Killing vector field* of M if it is the solution to the following optimization problem:

$$(6) \qquad \min_U E_K(U) \quad s.t. \quad \int_M \|U\|^2 \, dv = 1 \, .$$

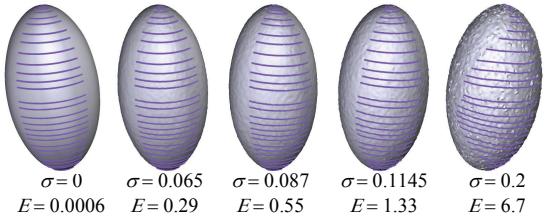


Figure 7: *The mesh of an ellipsoid is perturbed by adding Gaussian noise with standard deviation σ times the average edge length in the normal direction. We show the flow lines of the AKVF for a fixed elapsed time from a few selected vertices, in addition to σ and the resulting Killing energy.*

Fig 7 shows the approximate KVF of a perfect rotationally symmetric ellipsoid, as it gets progressively deformed by noise, and its matching Killing energy. This ellipsoid has one exact KVF, whose E_K is 0. As noise is added by perturbing the surface with Gaussian noise in the normal direction, the surface ceases to be symmetric, and there is no U such that $E_K(U) = 0$ anymore. However, for a small amount of noise, we can still see an approximate symmetry, generated by the approximate KVF. In our case, since we are working with discrete surfaces, even the "exact" ellipsoid has a non-zero (albeit small) Killing energy.

Noise, however, is not the main cause of symmetry loss, as there are many smooth surfaces which are only "close" to being symmetric and not exactly symmetric. Fig. 8 shows a series of deformations of a closed cone (i.e. a cone whose open end has been capped off). The perfect closed cone is a surface of revolution, and thus has one exact KVF. When squashing it, the bottom of the cone deforms in a non-isometric way and the cone is no longer perfectly symmetric. Nevertheless, the approximate KVF still exists and gives rise to the type of symmetry we would expect from such a shape. The AKVFs for both figures were computed using the method described in the next section, and the Killing energy is the minimal eigenvalue of (11).

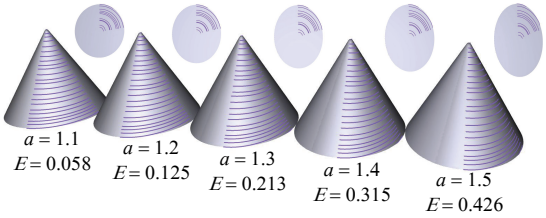


Figure 8: *A closed cone given by $(a \cdot h \cdot \cos(\theta), h, h \cdot \sin(\theta))$ with the respective Killing energy, and the flow lines of the AKVF for a fixed elapsed time from a few selected vertices.*

To solve the optimization problem, we will assume for simplicity that the surface does not have boundary, and discuss the boundary case later on. In this case, using standard calculus of variations, (6) can be formulated as an eigenvalue problem.

Lemma 2:

Given a two-manifold M, and a smooth tangent vector field U, then U is an approximate KVF of M if and only if:

$$(7) \qquad K^* K U = \lambda U \, ,$$

where K^* is the formal adjoint operator of K, and λ is the minimal such value.

The operator K^*K is semi-positive definite and thus $\lambda \geq 0$. When $\lambda = 0$ then U is an exact KVF and lies in the kernel of K^*K. As we have formulated this as an eigenvalue problem, we can consider not only the vector field minimizing E_K, but also "second best" vector fields.

Definition 7:

Given a two-manifold M and a smooth tangent vector field U, then U is a λ-*approximate Killing vector field* of M if it is an eigenvector of K^*K with eigenvalue λ.

The benefit of considering not only the "best" AKVF can be seen when analyzing the deformation of a sphere to an ellipsoid. A sphere has 3 exact KVFs, and when smoothly stretched into a rotationally symmetric ellipsoid, only 1 exact KVF remains and a gap is generated between $\lambda_0 = 0$ and $\lambda_1 = \lambda_2$. The more the sphere is stretched, the larger the gap. The deformation from a sphere to an ellipsoid is shown in Fig. 9, together with the three smallest eigenvalues of K^*K.

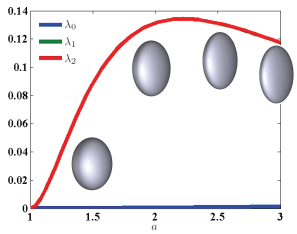


Figure 9: *A sphere deformed into an ellipsoid with radius a. The first 3 eigenvalues of the Killing operator split, generating a gap between $\lambda_0 = 0$ and $\lambda_1 = \lambda_2$.*

3. Discrete Killing Vector Fields

3.1 The Exterior Calculus Approach

Exact and approximate KVFs would not be of much use if there were no concrete way of computing them. An obvious approach would be to discretize equation (6) by discretizing the covariant derivative. Unfortunately, this is not trivial to do on a triangulated mesh because the covariant derivative involves the derivative of the metric, which on a piecewise flat mesh is zero. Of course, as is done for the computation of other higher-order properties, one could approximate the underlying surface using a quadratic fit, and calculate the covariant derivative on the fitted surface. This method is, however, somewhat cumbersome and potentially computationally heavy.

We opt instead for a simpler approach based on Discrete Exterior Calculus. In this setting, we can avoid defining the covariant derivative altogether and instead reformulate the problem in terms of the well known Hodge Laplacian for one-forms [FSDH07]. This yields an extremely simple implementation for the K^*K operator.

First, we reformulate (7) on a smooth manifold, in terms of one-forms and exterior calculus. Then, we can use existing definitions of the discrete variants involved, such as the divergence and the curl of a one-form on a mesh [Hir03], to get the discrete analog of (7). Again, we will assume for now the surface does not have boundary, and discuss the boundary case later on.

The connection between the Killing energy and the Hodge Laplacian for one-forms is given by the following Theorem. This connection is well known, and is an example of the so-called "Bochner Technique" [Pet97, Chapter 7] in differential geometry. We repeat the derivation here to provide some insight into the geometry behind the formula.

***Theorem 1*:**

Given a two-manifold M without boundary and a vector field U, the following holds:

$$E_K(U) = \int_M \|KU\|^2 \, dv = 2\int_M \left(\|d\omega\|^2 + 2(\delta\omega)^2 - 2k\|\omega\|^2 \right) dv \\ = 2\int_M \langle \Delta\omega + d\delta\omega - 2k\omega, \omega \rangle dv \tag{8}$$

where ω is the one-form corresponding to U, and d and δ are the exterior derivative and co-differential operator respectively. For a one-form, δ is the divergence operator, taking a one-form and returning a scalar, whereas d is the curl operator, taking a one-form and returning a two-form. The lengths and inner products are measured with respect to the metric g, k is the Gaussian curvature, and Δ is the Hodge Laplacian for one-forms.

To see why the theorem is true, consider a vector field on a planar domain. In this case J is just the Jacobian matrix of U, and simple algebra shows that:

$$\frac{1}{2}\left\| J + J^T \right\|_F^2 = \|\nabla \times U\|^2 + 2(\nabla \cdot U)^2 - 4\det(J), \tag{9}$$

where $\nabla\times$ and $\nabla\cdot$ are the curl and divergence operators, respectively, and the last factor is the determinant of J. A closer look shows that

$$\det(J) = \nabla \cdot F \tag{10}$$

for some vector field F, and thus the integral of (10) over the whole domain is equal to the flux of F through the boundary of the domain. If this flux is 0, (for example if F is tangential to the boundary, or if the domain is a closed surface), we get that $\int det(J) dv$ vanishes, and we are left with the curl and divergence terms only.

On a curved surface we get an extra term when computing the norm of $J + J^T$. This happens because to derive (10) one needs to assume that covariant derivatives commute, which is true in the plane (where covariant derivatives are the usual partial derivatives) but fails on a curved surface. In fact, curvature is the *reason* that covariant derivatives do not commute, and we have $\nabla_1\nabla_2\boldsymbol{\omega} - \nabla_2\nabla_1\boldsymbol{\omega} = k\boldsymbol{\omega}^\perp$, where $\boldsymbol{\omega}^\perp$ is the counter clockwise rotation of ω by 90 degrees [doC92]. Incorporating this fact into our derivation yields the last term in (8). Using the fact that d and δ are formal adjoints, we get:

$$\int \langle d\omega, d\omega \rangle + \langle \delta\omega, \delta\omega \rangle = \int \langle \delta d\omega, \omega \rangle + \langle d\delta\omega, \omega \rangle = \int \langle \Delta\omega, \omega \rangle$$

and the second part of (8) follows. The full proof of Theorem 1 is given in Appendix A.

3.2 Discrete Approximate KVFs

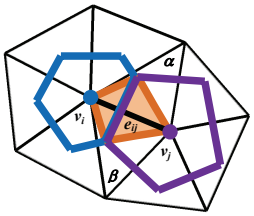
Given a triangulated mesh $M = (V,F,E)$, we would like to find a discrete version of equation (8). The quantities in Eq. (8) have discrete analogues, given by Discrete Exterior Calculus [Hir03]. We choose the same formulation as in

[FSDH07] for the definition of one-forms, the operators δ, d, and the Hodge Laplacian of one-forms. In this setup, a one-form is given by a scalar on each edge, and there is a correspondence mapping tangent vector fields on the surface to one-forms and vice versa. The exterior calculus operators are simply matrices. The exterior calculus boundary operator is a matrix which maps each element to its boundary (e.g., maps an edge to its endpoints, and so on). The operator d is the transpose of the boundary operator, and $\delta = \star^{-1} d \star$, where $\star$ is the Hodge operator given as a diagonal matrix. See the supplemental material for the explicit expressions for these operators, and the mapping between one-forms and vector fields. Combining these matrices, we get $\Delta = \delta d + d\delta$ simply by matrix multiplication. The size of the matrix Δ is $n_e \times n_e$, where $n_e = |E|$.

The only missing quantity is the Gaussian curvature on the edge. In the usual setting, when solving for scalar functions defined on the vertices, the Gaussian curvature is 0 everywhere except at the vertices, where it is defined to be the integral of the curvature over the vertex' Voronoi region. Since we are interested in one-forms which live on the edges, we redistribute the curvature to the edges using:

$$k_{ij} = \frac{1}{2} A_{ij} \left(\frac{k_i}{|v_i^*|} + \frac{k_j}{|v_j^*|} \right) \text{ with } \mathrm{A}_{ij} = \frac{|e_{ij}|^2}{4} \left(\cot\alpha + \cot\beta \right),$$

where e_{ij} is the edge (v_i, v_j) with length $|e_{ij}|$, and α and β are the angles opposite to e_{ij}. In addition, k_i and k_j are the discrete Gaussian curvatures at v_i and v_j respectively (defined as 2π minus the sum of angles around the vertex), and $|v_i^*|$ is the Voronoi area of v_i. See the inset figure for the notations. We can interpret this as "splitting" each vertex into a few vertices (depending on the vertex' degree), and then taking the curvature on the edge to be the sum of curvatures at its "split" endpoints. Since we did not add or remove curvature, the sum of the curvature over the whole surface is preserved, and we still have: $\sum_{(i,j)\in E} k_{ij} = \sum_i k_i = 2\pi\chi$.



Finally, we can define the discrete analog of approximate KVFs by plugging in the discrete analogues in eq. (8).

Definition 8:

Given a triangulated mesh $M = (V,F,E)$, let R be the matrix given by $R = \Delta + d\delta - 2BG$, where G is the diagonal matrix whose entries are k_{ij} / A_{ij} for every edge e_{ij} (we need point-wise curvature, as opposed to the integrated quantity), and B is the diagonal Hodge operator for one-forms. A tangent vector field U is a *discrete λ-approximate KVF* if it is the vector field corresponding to a one-form ω, and:

$$R\omega = \lambda B\omega . \quad (11)$$

Note that the definition of the discrete AKVF is intrinsic, as we only use the edge lengths, and not the actual embedding.

Figures 6-8 show examples of discrete approximate KVFs computed this way, and Fig. 10 shows the "best" AKVF for different surfaces.

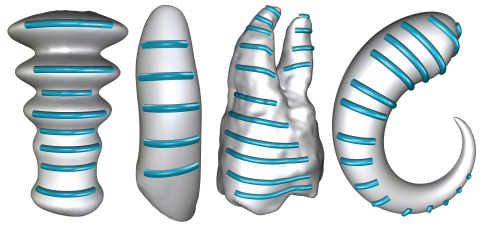


Figure 10: *The figure shows the flow lines of the discrete AKVF from a few selected vertices for a fixed elapsed time.*

Even when the surface does not possess an obvious global intrinsic symmetry, the eigenvectors of R are still of interest. The general behavior appears to be that some of the possible *local* symmetries are captured by different eigenvectors. For example, in Figure 11, the shape has a few possible local intrinsic symmetries, for each of the extrusions. In this case, the first eigenvector is localized around the right limb, and its norm is close to zero on the rest of the surface. The second eigenvector is localized on the left limb, and so on. Figure 11 shows the color coding of the norm of the vector fields which match the first four eigenvectors and some flow lines. Note, though, that as the eigenvector computation is global, most likely not all possible local symmetries are captured this way.

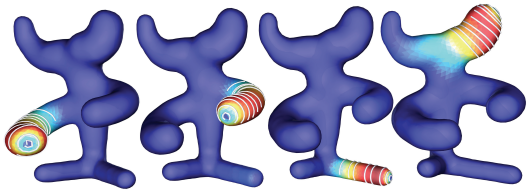


Figure 11: *From left to right, the first to fourth eigenvector of R. The figure shows the color coding of the norm of the vector field, and a few flow lines.*

3.3 Experimental Validation

We would like to check empirically that the vector fields computed using (11) are in fact approximate Killing vector fields, as given by the continuous definition (8).

Approximate KVFs on The Sphere

In general, it is hard to solve (8) analytically as it boils down to a system of coupled differential equations. However, in the case of the sphere, by applying the Hodge decomposition we can show that: $\lambda \in \{2\lambda_s - 4k, 4\lambda_s - 4k\}$, where λ_s are the non-zero eigenvalues of the Laplace-Beltrami operator (of scalar functions) on the sphere, and k is the constant Gaussian curvature (see Appendix B for the proof). Since the non-zero values of λ_s are known to be $m(m+1)k$ for positive integers $m > 0$, we can check the eigenvalues of R to see if they achieve their expected values. Indeed, when

computing the eigenvalues of R on a mesh of the unit sphere with 20,000 vertices, we get:

$$\sqrt{\sum_i \left(\lambda_i^{computed} - \lambda_i\right)^2} \Big/ \sqrt{\sum_i \lambda_i^2} \approx 0.0016 ,$$

for the first 180 eigenvalues of R. In general, the multiplicity of the eigenvalues is the same as the multiplicity of the eigenvalues of the spherical Laplace-Beltrami operator, with the special case of $\lambda_i = 20k$ (see Appendix B).

Harmonic vs. Killing Vector Fields

There exists an interesting relationship between harmonic and Killing vector fields. If a surface has both a harmonic vector field and a KVF, then their inner product is constant [Yan52]. Since discrete harmonic vector fields are easy to compute as the kernel of the Hodge Laplacian, we can check that this property holds, and thus check that our KVF is compatible with other entities on the surface. Figure 12 shows one pair of harmonic and Killing vector fields. In this case, the vector fields are parallel, but their norm is different, so that one is shorter when the other is longer, the total effect yielding a close to constant inner product.

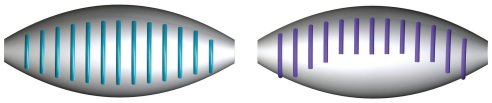


Figure 12: *A Killing vs. a harmonic vector field (HVF). The figure shows the flow lines from a few selected vertices for a fixed elapsed time. The KVF and HVF are parallel, yet with different norms, one speeding up when the other slows down, yielding an almost constant dot product.*

An Isometric Deformation

An important test that a vector field is indeed an exact KVF is that its flow generates an isometric deformation. To check that, we generated a sequence of meshes, whose embeddings are $\left\{X_t = \phi^t(X_0) \mid t = m\varepsilon, m \in \mathbb{N}\right\}$, where ϕ is the flow of U and X_0 is the embedding of the original mesh.

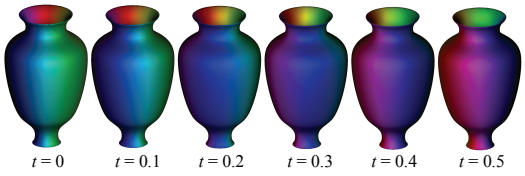


Figure 13: *The sequence of meshes generated by following the flow lines of the KVF from all the vertices, for the specified elapsed times. The color coding for all the vertices is the same as in the original model (t = 0).*

Figure 13 shows this sequence of meshes for a symmetric model. To visualize the change between the models, we choose a color coding for the original model, and reuse the same colors for all the meshes in the sequence. As is evident from the figure, the deformations are indeed close to isometric. For a more qualitative test, we compute the relative mean squared error of the edge lengths:

$$Err_i = \frac{1}{n_e}\sqrt{\sum_{(i,j)\in E}\left(l_{ij}^0 - l_{ij}^i\right)^2} \Big/ \sqrt{\sum_{(i,j)\in E}\left(l_{ij}^0\right)^2} ,$$

where l_{ij}^k is the length of the edge (i,j) in the k-th mesh. The resulting errors are quite small, of the order of 10^{-6}.

3.4 Surfaces with Boundary

On a surface with boundary, eq. (8) is not correct anymore since the integral of the determinant of the covariant derivative does not vanish and thus the eigenvectors we find using (11) are not the minimizers of (8). The correct expression (see Appendix A) is:

$$\int_M \|KU\|^2 dv = 2\int_M \langle \Delta\omega + d\delta\omega - 2k\omega, \omega\rangle dv + 4\int_{\partial M} \langle \nabla_T \omega, \omega^\perp \rangle ds$$

where T is the tangent vector to the boundary. By taking $\omega = a \cdot dt + b \cdot dn$, where dt is the unit tangent vector, and dn is the normal vector at the boundary, we get:

$$\langle \nabla_T \omega, \omega^\perp \rangle = b\frac{\partial a}{\partial t} - a\frac{\partial b}{\partial t} - k_g \|\omega\|^2 , \tag{12}$$

where $\partial/\partial t$ is the derivative in the tangent direction, and k_g is the geodesic curvature of the boundary. In the special case that the boundary is a geodesic (i.e. $k_g = 0$), and ω is either tangential ($b = 0$) or normal ($a = 0$) to the boundary, (12) vanishes (as is the case for the models in Figures 1, 2 and 7), and we get the same expression as in the boundaryless case. We do not currently handle the general case; however as a discrete counterpart for all the quantities exists, the extension to this case is quite straightforward.

4. Application – Intrinsic Pattern Generation

To demonstrate possible uses of Killing vector fields, we decorate almost symmetric surfaces by restricting the continuous symmetry defined by the AKVF to a discrete one.

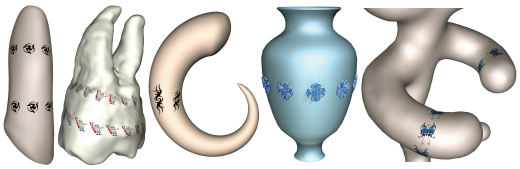
Given a surface which has continuous intrinsic symmetries, we can define discrete *Intrinsic Symmetry Groups*, and use them to generate patterns. If we are given a continuous symmetric surface, after we endow it with a pattern, it will be discretely symmetric. We will concentrate only on 1-parameter group of intrinsic symmetries, appearing in isometric deformations of surfaces of revolution. Note that unlike other pattern generation methods, due to the special nature of our vector fields, the patterned surface will be "as symmetrical as possible".

Let M be (an isometric deformation of) a surface of revolution. In this case, M has exactly one KVF U, with ϕ^t as its induced deformation. It is easy to check that the flow of U on M generates closed curves. Furthermore, given a point $p \in M$ and γ the integral curve starting from p, we have that $\|U(\gamma(t))\|$ does not depend on t, and is proportional to the length of γ. This means that for two points $p_1, p_2 \in M$, with integral curves γ_1 and γ_2 respectively, such that $\gamma_1 \neq \gamma_2$, we have that: $L(\gamma_2)/\|U(\gamma_2)\| = L(\gamma_1)/\|U(\gamma_1)\| = T$ where $L(\gamma)$ is the length of the curve γ. Hence, we have that $\phi^T(p) = p$ for any point $p \in M$. Now, we can choose a number $q \in R$, such

that $T/q = m$ is an integer, and get the discrete symmetry group: $\left\{\phi^q, \phi^{2q}, ..., \phi^{(m-1)q}, \phi^{mq} = \phi^T = id\right\}$.

To generate a pattern we choose a point p, find its images under the discrete symmetry group $p^1, p^2, \ldots, p^{m-1}$, and map a small environment of these points to a common domain, e.g. the unit disk. Now any pattern applied to this common domain – such as a texture, or a height function – would be reflected in each of the points p^i. In addition, as is shown in Figure 14 (*d*), we can also use our vector fields to drive the pattern generation from [LFZ*10].

Figure 14 shows a few patterns generated this way on surfaces which are either surfaces of revolution, or almost isometric deformations of such. To generate the texture coordinates we choose vertices v_i and a radius r, and find the vertex sets $V_i = \{v \mid d(v, v_i) < r \|U(p_i)\|\}$. Then we map each set V_i to the unit disk, and define the texture coordinates of the rest of the vertices in the mesh, as the (interpolated) texture coordinates of their pre-image under the flow of the vector field. Since the size of the mapped area is proportional to the norm of the AKVF, we get a nice scaling effect of the texture. Note that this is considerably less complicated than mapping the whole surface to the plane.



(*a*) (*b*) (*c*) (*d*) (*e*)

Figure 14: *Intrinsic patterns on almost symmetric surfaces, generated by following the flow lines of the AKVF from a few selected vertices for fixed time intervals.(a-c) generated using texture coordinates, and (d) using the method from [*LFZ*10*]. (e) was generated by adding the first two AKVFs of the shape in Fig. 11.*

The biggest limitation of our method is that it is restricted to continuous symmetries by the very nature of the definition of KVFs. Thus, more complex models, which do in-fact exhibit discrete symmetries, will usually not posses KVFs. However, we believe the way to avoid this problem is by decomposing a shape into smaller pieces or by removing existing reliefs or decorations. Then, approximate continuous symmetries can be found and used for extracting the discrete symmetries which the original shape possessed. We believe this could be a fruitful avenue for further research.

5. Discussion

We have proposed a new method for tackling the challenging subject of continuous intrinsic symmetries of surfaces. We showed how to replace the rigid transformations used for defining extrinsic symmetries with deformations generated by Killing vector fields. Furthermore, we showed how to relax the restriction of exact symmetry to allow finding structure in almost intrinsically symmetric surfaces. Our formulation is quite simple, requiring only the solution of an eigenvalue problem defined in terms of well known Discrete Exterior Calculus operators. Finally, we proposed a simple application for generating symmetric patterns on symmetric surfaces using intrinsic symmetry groups.

In the future we wish to explore further issues relating to discrete KVFs. The most prominent one is how to use this machinery for extracting patterns from existing surfaces, as opposed to generating them. From a theoretical point of view, we would like to better understand the relationship between the minimal λ and the existence of an "almost" symmetry group. It is also possible that the spectrum of R would prove useful for shape classification in a similar manner to the Laplacian spectrum. AKVFs might also be relevant for deformation applications.

To sum up, AKVFs provide a new way of investigating intrinsic approximate symmetries of surfaces. Moreover, it seems they hold important information about a shape, and thus can potentially become a valuable tool in additional geometry processing applications.

Acknowledgements

We thank Peter Wonka and Yuan Li for their help with the vase model, and gratefully acknowledge the support of the following grants: The Fulbright fellowship, the Weizmann Institute's "Women in Science" award, NSF grants 0808515 and 0808515, NIH grant GM-072970, a grant from the King Abdullah University of Science and Technology, and a gift from Google, Inc.

References

[ACXG09] AKLEMAN E., CHEN J., XING Q., GROSS J. L.: Cyclic plain-weaving on polygonal mesh surfaces with extended graph rotation systems. In *Proc. SIGGRAPH '09* (2009).

[BBW*09] BOKELOH M., BERNER A., WAND M., SEIDEL H.-P., SCHILLING A.: Symmetry detection using line features. *Computer Graphics Forum 28*, 2 (2009), 697–706.

[Bee08] BEETLE C.: Approximate Killing fields as an eigenvalue problem. *http://adsabs.harvard.edu/abs/2008arXiv0808.1745B* (2008).

[Boo02] BOOTHBY W. M.: *An introduction to differentiable manifolds and Riemannian geometry*. Academic Press, 2002.

[doC92] DOCARMO M. P.: *Riemannian geometry*. Birkhäuser Boston, 1992.

[FSDH07] FISHER M., SCHRÖDER P., DESBRUN M., HOPPE H.: Design of tangent vector fields. In *Proc. SIGGRAPH '07* (2007).

[GMDW09] GRAVE F., MÜLLER T., DACHSBACHER C., WUNNER G.: The Gödel Engine - An interactive approach to visualization in general relativity. *Computer Graphics Forum 28*, 3 (2009), 807–814.

[Hal04] HALL G. S.: *Symmetries and curvature structure in general relativity*. World Scientific Publishing, 2004.

[Hir03] HIRANI A. N.: *Discrete Exterior Calculus.* PhD thesis, Caltech, 2003.

[Kap09] KAPLAN, C. S.: Semiregular patterns on surfaces. In *Proc. Int. Symp. on NPR Animation and Rendering* (2009), 35–39.

[KMP07] KILIAN M., MITRA N. J., POTTMANN H.: Geometric modeling in shape space. In *Proc. SIGGRAPH '07* (2007).

[LFZ*10] LI Y., FAN B., ZHANG E., KOBAYASHI Y., WONKA P.: Geometry synthesis on surfaces using field-guided shape grammars. *IEEE Trans. on Vis. and Computer Graphics*, to appear (2010).

[LTSW09] LASOWSKI R., TEVS A., SEIDEL H.-P., WAND M.: A probabilistic framework for partial intrinsic symmetries in geometric data. In *Proc. ICCV '09* (2009).

[Mat68] MATZNER R. A.: Almost symmetric spaces and gravitational radiation. *J. Math. Phys. 9,* 10 (1968), 1657-1668.

[MGP06] MITRA N. J., GUIBAS L., PAULY M.: Partial and approximate symmetry detection for 3D geometry. In *Proc. SIGGRAPH '06* (2006).

[Mye36] MYERS S. B.: Isometries of 2-dimensional Riemannian manifolds into themselves. In *Proc. National Academy of Sciences of the USA 22*, 5 (1936), 297-300.

[OSG08] OVSJANIKOV M., SUN J., GUIBAS L. J.: Global intrinsic symmetries of shapes. *Comp. Graphics Forum 27*, 5 (2008), 1341-1348.

[Pet97] PETERSEN P.: *Riemannian geometry*. Springer, 1997.

[PLC*08] PARK M., LEE S., CHEN P.-C., KASHYAP S., BUTT A. A., LIU Y.: Performance evaluation of state-of-the-art discrete symmetry detection algorithms. In *Proc. CVPR '08* (2008).

[PMW*08] PAULY M., MITRA N. J., WALLNER J., POTTMANN H., GUIBAS L.: Discovering structural regularity in 3D geometry. In *Proc. SIGGRAPH '08* (2008).

[RBBK10] RAVIV D., BRONSTEIN A. M., BRONSTEIN M. M., KIMMEL R.: FULl and partial symmetries of non-rigid shapes. *Intl. J. of Computer Vision*, to appear (2010).

[WAL84] WALD R. M.: *General Relativity*. University Of Chicago Press, 1984.

[XZT*09] XU K., ZHANG H., *TAGLIASACCHI A., LIU L., LI* G. MENG M., XIONG Y.: Partial intrinsic reflectional symmetry of 3D shapes. In *Proc. SIGAsia '09* (2009).

[Yan52] YANO K.: On harmonic and Killing vector fields. *The Annals of Mathematics 55*, 1 (1952), 38-45.

[Zal99] ZALALETDINOV R.: Approximate symmetries in general relativity. *arXiv:gr-qc/9912021* (1999).

[ZHW*06] ZHOU K., HUANG X., WANG X., TONG Y., DESBRUN M., GUO B., SHUM H.-Y.: Mesh quilting for geometric texture synthesis. In *Proc. SIGGRAPH '06* (2006).

Appendix A:

We first derive an identity connecting the pointwise value of $||KU||$ to that of $||d\omega||$ and $\delta\omega$, where ω is the one-form associated to U. Since our expressions are coordinate-independent, we perform the calculation in *geodesic normal coordinates*, so that the metric is as simple as possible. This way, the components of the matrix representing g equal those of the Euclidean metric at an arbitrary $p \in M$ and have vanishing first derivatives at p.

Let ω have components ω_1 and ω_2 with respect to our chosen coordinates. Then $d\omega$ and KU are represented by the matrices whose (i,j) entries are $\nabla_j\omega_i - \nabla_i\omega_j$ and $\nabla_j\omega_i + \nabla_i\omega_j$, respectively, while $\delta\omega = -\nabla_1\omega_1 - \nabla_2\omega_2$. We first express $||KU||^2$ in terms of $||d\omega||^2$ and $(\delta\omega)^2$ as well as possible. Algebraic manipulations yield $\frac{1}{2}||KU||^2 =$

$$(\nabla_2\omega_1 - \nabla_1\omega_2)^2 + 2(\nabla_1\omega_1 + \nabla_2\omega_2)^2 - 4Q = \|d\omega\|^2 + 2(\delta\omega)^2 - 4Q$$

where $Q := \nabla_1\omega_1\nabla_2\omega_2 - \nabla_1\omega_2\nabla_2\omega_1$. We now show that Q is the co-differential of a one-form plus a remainder term. To begin, note that: $2Q =$

$$\nabla_1 F_1 + \nabla_2 F_2 + \omega_1(\nabla_2\nabla_1\omega_2 - \nabla_1\nabla_2\omega_2) - \omega_2(\nabla_2\nabla_1\omega_1 - \nabla_1\nabla_2\omega_1)$$

where:

$$F_1 = (\omega_1\nabla_2\omega_2 - \omega_2\nabla_2\omega_1) \text{ and } F_2 = (\omega_2\nabla_1\omega_1 - \omega_1\nabla_1\omega_2).$$

We re-formulate the two terms above as the divergence of a one-form, by introducing the one-form F that we define in a coordinate-independent manner as $F(Y) := -g(\nabla_{Y^\perp}\omega, U^\perp)$. Note that the components of F in the coordinates we are using are exactly F_1 and F_2. It now follows from the relation between curvature and second covariant derivatives, namely $\nabla_2\nabla_1\omega - \nabla_1\nabla_2\omega = -k\omega^\perp$, where k is the Gauss curvature of M, that $||KU||^2 = 2(||d\omega||^2 + 2(\delta\omega)^2 - 2k||\omega|| - 2\delta F)$. By integrating both sides, the co-differential term vanishes by Stokes' Theorem and we obtain the formula claimed in Theorem 1.

Appendix B:

We assume Σ is the sphere of radius r which has constant Gauss curvature equal to $k := r^{-2}$. Substitute $\omega := d\phi + *d\psi$, where $\phi,\psi: \Sigma \to \mathbb{R}$ are functions, into the AKVF equation.

We get $0 = *d(2\Delta\psi + (4k + \lambda)\psi) - d(4\Delta\phi + (4k + \lambda)\phi)$. By the orthogonality of the Hodge decomposition, the equation above implies $2\Delta\psi + (4k + \lambda)\psi = c_1$ and $4\Delta\phi + (4k + \lambda)\phi = c_2$, where c_1 and c_2 are constants. But since we can add any constant to ϕ or ψ without changing ω, then we can assume that $c_1 = c_2 = 0$. Hence ϕ and ψ are eigenfunctions of the scalar Laplace-Beltrami operator on the sphere of radius r. These are the *spherical harmonics*, denoted $Y_n^{l_n}$ for $l_n = 1,\ldots,\mu_n := 2n+1$ and $n \in \mathbb{N}$, and corresponding to the eigenvalue $\beta_n := n(n+1)k$. Three cases are possible:

$$\exists n,m\ 2k + \lambda/2 = \beta_n,\ k + \lambda/4 = \beta_m \Rightarrow \lambda = 20k, \psi = Y_n^{l_n}, \phi = Y_m^{l_m}$$

$$\exists n\ \ 2k + \lambda/2 = \beta_n,\ \forall m\ \ k + \lambda/4 \neq \beta_m \Rightarrow \begin{matrix} \lambda \in \{2\beta_n - 4k \mid n \neq 3\} \\ \psi = Y_n^{l_n}, \phi = 0 \end{matrix}$$

$$\forall n\ 2k + \lambda/2 \neq \beta_n,\ \exists m\ \ k + \lambda/4 = \beta_m \Rightarrow \begin{matrix} \lambda \in \{4\beta_m - 4k \mid m \neq 2\} \\ \psi = 0, \quad \phi = Y_m^{l_m} \end{matrix}$$

Note, that the eigenvalue $\lambda = -4k$ can be discounted since it corresponds to the eigenfunctions $\psi = const$ and $\phi = const$, and the AKVF corresponding to such a choice of functions is zero.