EUROGRAPHICS 2020 / U. Assarsson and D. Panozzo
(Guest Editors)

Volume 39 (2020), Number 2

Locally supported tangential vector, *n*-vector, and tensor fields

Ahmad Nasikun[1,2] Christopher Brandt[3] Klaus Hildebrandt[1]

[1]Delft University of Technology, The Netherlands
[2]Department of Electrical and Information Engineering, Universitas Gadjah Mada, Indonesia
[3]École Polytechnique Fédérale de Lausanne, Switzerland

Abstract

We introduce a construction of subspaces of the spaces of tangential vector, n-vector, and tensor fields on surfaces. The resulting subspaces can be used as the basis of fast approximation algorithms for design and processing problems that involve tangential fields. Important features of our construction are that it is based on a general principle, from which constructions for different types of tangential fields can be derived, and that it is scalable, making it possible to efficiently compute and store large subspace bases for large meshes. Moreover, the construction is adaptive, which allows for controlling the distribution of the degrees of freedom of the subspaces over the surface. We evaluate our construction in several experiments addressing approximation quality, scalability, adaptivity, computation times and memory requirements. Our design choices are justified by comparing our construction to possible alternatives. Finally, we discuss examples of how subspace methods can be used to build interactive tools for tangential field design and processing tasks.

1. Introduction

Directional information along a surface is usually encoded as a tangential vector, *n*-vector or tensor field. Many applications in computer graphics, such as line art rendering, meshing, texturing, and BRDF design, rely on techniques for the design and processing of such tangential fields. A problem arises from the fact that the triangular meshes describing the surface are usually of high resolution and the complexity of the tangential fields is connected to that of the meshes. As a result, large-scale equations and optimization problems have to be solved for field design and processing, while the applications expect fast response times because workflows often involve user interaction. Established acceleration methods, such as radial basis functions, which proved to be effective for problems like interactive shape deformation, cannot be used for the processing of tangential fields since the fields are defined on curved surfaces and the tangential bundles of the surfaces are non-trivial.

We introduce constructions of tangential vector, *n*-vector and tensor fields on meshes and use them to construct subspaces for design and processing tasks. By restricting the equations and optimization problems to such a subspace, the degrees of freedom of a design or processing problem can be adjusted detached from the complexity of the triangle mesh representing the surface. This is an important step towards enabling interactive techniques for the design and processing of tangential fields on large meshes.

The idea underlying our approach is to construct tangential fields by computing the lowest eigenfields of a suitable Laplace operator restricted to small disk-shaped subsets of the surface. Boundary conditions on the eigenproblems are imposed to guaranty that each of the resulting fields vanishes at the boundaries of its subset. The construction of the individual fields is combined with a procedure that defines the subsets in the surface so that useful bases of subspaces are created. Important aspects of our construction are:

- *Generality* Our construction is derived from a general principle. By choosing an appropriate Laplace operator, specific constructions for the different types of tangential fields can be derived. We explicitly describe constructions of vector, *n*-vector, and tensor fields.
- *Scalability* We show that large subspaces with several thousand dimensions on meshes with more than a million triangles can be efficiently constructed and stored. A prerequisite for this is the localization of the tangential fields, which facilitates the efficient computation and storage of the individual fields. The cost for field construction depends only on the size of the disk-like subsets and storage requirements are low as the fields can be represented by sparse vectors.
- *Smoothness* The construction is chosen so that the resulting fields are smooth. The eigenproblems we solve can be written as optimization problems, and, as minimizers, the lowest eigenfields are the smoothest fields that vanish outside their assigned subsets. Here smoothness is measured by the Dirichlet energy corresponding to the Laplace operator that is used.
- *Approximation* We show that the resulting subspaces can approximate smooth fields well. To evaluate this aspect, we compute residuals when projecting fields into the subspace and when solving optimization problems in the subspaces, and compare the approximation results with results obtained using other possible constructions of tangential fields.

- *Adaptivity* We show that adaptivity can be effortlessly integrated to the construction. This allows for controlling the distribution of the degrees of freedom of a subspace over the surface. For example, fields that include details in designated areas of the surface can be better represented in the subspaces.

This is the first method for constructing tangential vector, n-vector and tensor fields with these properties. In particular, scalability and adaptivity distinguish the construction from alternative vector field constructions. These two points are crucial for efficiently generating subspaces with good approximation properties and modeling capacity on larger meshes. To justify our design choices for the proposed construction, we compare our construction with other existing and possible constructions in Section 7. In addition, we evaluate the approximation quality of the subspaces in different settings, Section 6, and show the benefits of the bases for the applications in Section 8.

2. Related work

Tangential fields The efficient design and processing of tangential vector, n-vector and tensor fields is important for a broad range of applications in computer graphics. Examples are texture generation [PFH00, Tur01, WL01, CYZL14, KCPS15], line art [HZ00] and painterly rendering [ZHT07], anisotropic shading [MRMH12, RGB*14], image stylization [YCLJ12], surface segmentation [SBCBG11, ZZCJ14], surface construction [IBB15, PLS*15], meshing [RLL*06, KNP07, BZK09, LLZ*11, TPP*11], and the simulation of fluid and liquids on surfaces [AWO*14, AVW*15]. Tangential field design and processing presents many challenges, and different approaches have been proposed to address these problems. In the following, we briefly discuss approaches that are closely related to our work. For further background information and references, we refer to the surveys [dGDT15, VCD*16].

Variational approaches for the design and processing of tangential fields minimize an objective that combines a fairness measure and functionals that penalize the deviation of the field from user input or geometric properties of the surface like curvature directions. The fairness measures quantify the variation of the field along the surface. For *vector field* design, quadratic objectives based on the divergence and curl of the fields can be used [FSDH07]. Discrete differentiable operators for *tensor fields*, based on Discrete Exterior Calculus [DHLM05], are introduced in [dGLB*14]. For *n-rotational symmetric vector fields* (n-fields), the fairness measure introduced in [HZ00] measures the deviation in angle between close-by n-vectors. To disambiguate the definition of angles between n-vectors, a periodic function is used in the fairness measure. The concept of representation vectors corresponding to n-vectors [RLL*06, PZ07] allows to model n-vector design using optimization of the representation vector fields. The representation vectors can be used to define a linear structure on n-vectors. This allows to model n-field design and processing problems using linear systems [KCPS13, LTGD16, BSEH18]. For the design of general, not necessarily rotational symmetric, n-fields, the polyvector representation [DVPSH14, DVPSH15, SFCBCV19] was introduced. While typically fairness measures are modeled as intrinsic objectives, an extrinsic objective was proposed in [JTPSH15, HJ16]. The objective combines intrinsic fairness and alignment to curvature, while at the same time avoiding the need to use parallel transport on the surface for evaluation of the objective. In another line of work, explicit matchings that encode the n pairs of corresponding vectors between neighboring n-vectors are used [KNP07, RVLL08, BZK09]. In an optimization, the matchings are treated as variables which leads to mixed-integer problems that have to be solved. In addition to variational design of vector fields, approaches that construct vector fields from user input specifying the location and degree of singularities have been proposed [ZMT06, PZ07, RVAL09, CDS10, LJX*10]. A subdivision scheme for discrete differential forms on meshes was introduced in [WYT*06]. The scheme combines different subdivision rules for the different k-forms such that the subdivision operations commute with the exterior derivative. This approach was extended to a subdivision exterior calculus [dGDMD16] that provides a way to apply the numerical tools from Discrete Exterior Calculus to subdivision surfaces. Among other applications, the approach can be used for the design of vector fields on subdivision surfaces. In recent work [CV20], a structure-preserving subdivision approach for tangential direction fields was developed and used for directional field design on subdivision surfaces.

Subspace methods Subspace methods can be used for the design of fast approximation algorithms for complex systems. In the preprocessing stage, the subspace and additional structures for evaluation the objective and its derivatives are constructed. In the online stage, the precomputed structures are used to accelerate computations. The low computational cost in the online stage, makes subspace methods attractive for interactive graphics applications. Reduced systems have been proposed for the simulation of fluids [TLP06, LMH*15, CSK18], elastic solids and shells [BJ05, AKJ08, YLX*15, BEH18], fluid-solid interaction [LJF16, BSEH19], example-based elastic material [ZZM15], motion planning [BdSP09, HSvTP12, PM18], clothing [HTC*14], and hair [CZZ14]. In the context of mesh processing, subspace methods have been introduced for surface modeling [HSL*06, HSvTP11, JBK*12, WJBK15], shape interpolation [vTSSH15, vRESH16], injective mappings [HCW19], motion processing [BvTH16], and spectral mesh processing [NBH18]. The goal of this paper is to explore subspace constructions for tangential vector, n-vector, and tensor fields on surfaces and the use of subspace methods for the design and processing of tangential fields.

Subspace methods for tangential fields For tangential vector fields, eigenfields of vector Laplace operators have been used for defining functional operators on spaces of vector fields [ABCCO13, AOCBC15], subspace fluid simulation on surfaces [LMH*15] and spectral vector field processing [BSEH17]. Eigenfields on an n-vector field Laplacian were used as the basis of an approach for interactive n-field design in [BSEH18]. Although eigenfields are useful for the construction of subspace in certain scenarios, there are also fundamental limitations. We propose an alternative construction of subspaces that addresses these limitations. In particular, we aim at reducing the memory requirements for storing the bases and the computational cost for constructing the bases. In Section 7, we compare the proposed subspaces to eigenspaces.

3. Laplace operators

In this section, we briefly describe the discrete Laplace operators for tangential vector, n-vector, and tensor fields that are needed for the proposed construction of localized fields. To our knowledge, the tensor field Laplacian we describe is novel.

Discrete Fields There are various possibilities for discretizing fields on meshes. Degrees of freedom of the fields can be associated with the meshes' vertices, edges, faces or combinations of these. We refer to the survey [dGDT15] for a detailed discussion of the benefits and drawbacks of different discretizations. Our basis construction can be used with any discretization as long as a Laplace operator on the space is available. For the evaluation of our construction, we consider vector, n-vector, and tensor fields that are constant and tangential in every face. We denote by k the dimension of the space of fields we consider. For tangential vector fields, for example, k equals twice the number of triangles of the mesh.

Laplacian for vector fields A discrete Hodge–Laplace operator Δ for piecewise constant vector fields is discussed in [BSEH17]. The operator combines discrete divergence and curl operators with the gradient and a 90-degree rotation in the tangent plane, which we denote by J,

$$\Delta = -\text{grad}\,\text{div} - \text{J}\,\text{grad}\,\text{curl}^*. \tag{1}$$

The divergence and curl operators map piecewise constant fields to piecewise linear polynomials and the gradient maps piecewise linear polynomials to piecewise constant vector fields. Matrix representation of all involved operators are described in [BSEH17]

For the piecewise constant fields on a mesh, a Hodge decompositions can be defined [PP00, War06]. This is an orthogonal decomposition of the space of piecewise constant fields in gradients and co-gradients (J grad) and harmonic fields. For the decompositions, two function spaces are needed: the space of continuous, piecewise linear polynomials (linear Lagrange finite elements) and the space of edge-midpoint continuous, piecewise linear polynomials (linear Crouzeix–Raviart elements). The Hodge–Laplacian can be constructed such that it respects the decomposition, which means that it maps gradient fields to gradient fields, co-gradient fields to co-gradient fields, and has exactly the harmonic fields in its kernel. To achieve this, one of the div and curl operators has to map to the space of continuous, piecewise linear polynomials and the other one to the edge-midpoint continuous, piecewise linear polynomials. In equation (1), we indicate that the curl operator maps to the space of edge-midpoint continuous functions by adding an asterisk.

Laplacians for n-fields Laplace operators for n-fields were proposed for a vertex-based representation in [KCPS13] and for face-based representation in [DVPSH14, BSEH18]. For our experiments, we use the face-based Laplacian. It computes differences of the n-vectors of each triangle to the n-vectors of the neighboring triangles. For this, the n-vectors of the neighbor triangles are parallelly transported to the corresponding triangle. In order to be able to form differences between n-vectors, a linear structure for n-vectors is required. This can be obtained using the concept of the representation vector of an n-field [RLL*06, PZ07]. The n-vectors are first converted to the corresponding representation vectors, then the difference is computed and the result is converted back to an n-vectors. For the choice of weights for the differences and a matrix representation of the Laplace operator, we refer to [BSEH18].

Laplacians for tensor fields We introduce a discrete Laplace operator for piecewise constant tensor fields on surface meshes. In this paragraph, we provide an overview of the construction and discuss details in the appendix. The operator is a weighted finite difference operator on the triangles of the mesh. To compute a difference between the tensor A_i of triangle T_i and the tensor A_j of a neighbor triangle T_j, we transport A_j parallelly to triangle T_i. The transport of a tensor to its neighboring triangle varies for the different tensor types depending on how the tensor transforms from one basis to another. We illustrate this at the example of $(1,1)$-tensors in the appendix. To describe the construction of the Laplace operator, we denote the transport of the tensor A_j to the triangle T_i by τ_{ji}. The Laplacian of a tensor field A is again a tensor field. In the triangle T_i the tensor field ΔA is given by

$$(\Delta A)_i = \frac{1}{m_i} \sum_{j \in N_i} w_{ij}(A_i - \tau_{ji}(A_j)), \tag{2}$$

where N_i is the list of the three neighbors of triangle T_i and

$$w_{ij} = \frac{3\,(\text{length}(e_{ij}))^2}{\text{area}(T_i \cup T_j)} \quad \text{and} \quad m_i = \text{area}(T_i)$$

are weights depending on the geometry of the triangles T_i and T_j. In the appendix, we show how Voigt's notation can be used to derive a linear representation of the tensors and the transport operator τ_{ij}. This can be used to construct the stiffness and mass matrices for this Laplacian.

4. Spaces of locally supported fields

In this Section, we describe our construction of subspaces of the spaces of tangential vector, n-vector and tensor fields. Our construction is based on a general principle that can be applied to the different types of tangential fields. We first introduce the construction of individual fields, then we describe how the field construction can be used to assemble bases of subspaces.

Field construction The input to the field construction are a Laplace operator, given by a stiffness matrix S and a diagonal mass matrix M, and a subset D of the set of triangles of the mesh that serves as the support of the field. Depending on whether the Laplacian operates on vector, n-vector or tensor fields, corresponding fields are constructed. Our approach is to compute the m lowest eigenfields of the Laplace operator subject to the constraint that the field is zero for all triangles that are not in the subset D. These fields can be characterized as the minimizers of the optimization problem

$$\begin{aligned} &\min_{\Phi \in \mathbb{R}^{k \times m}} \text{tr}\left(\Phi^T S \Phi\right) \\ &\text{subject to } \Phi^T M \Phi = Id \quad \text{and} \\ &\Phi_{ij} = 0 \text{ if } i \text{ belongs to a triangle not in } D. \end{aligned} \tag{3}$$

Each column of the minimizer Φ describes a localized field. The first constraint ensures that the fields are M-orthonormal and the

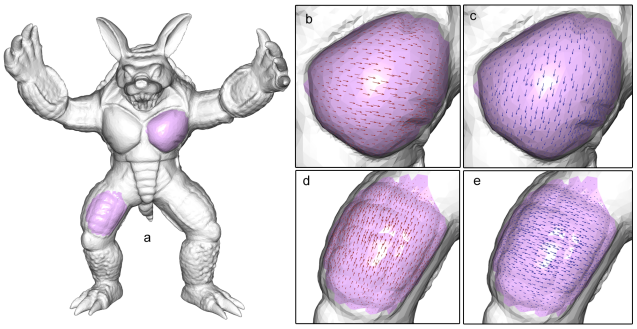

Figure 1: *Examples of localized tangential vector fields computed with our approach are shown. On the left, the locations of the support areas of the fields, and, on the right, four fields are shown.*

second constraint ensures that the fields vanish outside of the specified region.

Since the second constraint is linear, we can specify a basis for the space of vector fields that satisfy the constraints. We construct a matrix $V \in \mathbb{R}^{k \times k_D}$ whose columns form a basis of the space of admissible fields, where k_D is the dimension of the space of admissible fields. This matrix has one non-zero entry per column and the entries are located at the degrees of freedom of the vector associated with the selected triangles. Each entry takes the value $1/\sqrt{M_{ii}}$, where M_{ii} is the diagonal entry of the mass matrix M and i is the row index of the entry. The matrix V allows us to parametrize the space of admissible fields, *i.e.* for any admissible field $X \in \mathbb{R}^k$ there is a corresponding $x \in \mathbb{R}^{k_D}$ such that

$$X = Vx. \tag{4}$$

We consider the restricted stiffness and mass matrices

$$\bar{S} = V^T S V \qquad \text{and} \qquad \bar{M} = V^T M V. \tag{5}$$

By our construction of V, the restricted mass matrix $\bar{M}$ is the $k_D \times k_D$ identity matrix.

Using (4) and (5), we can rephrase the optimization problem (3)

$$\min_{\phi \in \mathbb{R}^{k_D \times m}} \operatorname{tr}\left(\phi^T \bar{S} \phi\right) \tag{6}$$
$$\text{subject to } \phi^T \phi = Id.$$

Benefits of this formulation are that we reduced the problem's dimension to $k_D m$, simplified the second constraint, and eliminated the third constraint.

The problem (6) is a sparse eigenvalue problem and the solutions are pairs (λ_i, ϕ_i) satisfying the equation

$$\bar{S}\phi_i = \lambda_i \phi_i. \tag{7}$$

The solutions $\phi_i \in \mathbb{R}^{k_D}$ are mapped to the corresponding vectors $\Phi_i \in \mathbb{R}^k$ by multiplying them with the matrix V. Examples of local vector and tensor fields are shown in Figures 1 and 2.

Subspace construction To construct subspace bases using the field construction, we need to define the support regions for the individual basis fields. Our approach is to cover the surface with disk-shaped regions. In this paragraph, we discuss a uniform distribution of the regions and extend the approach to adaptive distributions of the regions in the last paragraph of this section.

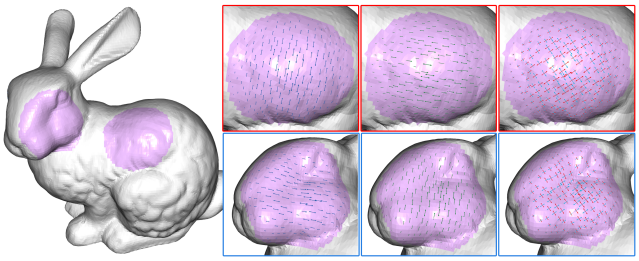

Figure 2: *Examples of localized tensor fields computed with our construction are shown. For each tensor, the shown sticks point in the two eigendirections and the lengths of the sticks are proportional to the absolute values of eigenvalues of the tensor.*

Each region is an approximate geodesic disk. All disks have the same radius r and the value of r is chosen such that the disks have sufficient overlap. We will discuss the choice of r we used for our experiments in Section 6. To place the disks on the surface, we sample triangles of the mesh that serve as the centers of the disks. To distribute the sampling uniformly, we use a furthest point sampling scheme. The distance is measured by a weighted Dijkstra algorithm that operates on the mesh's dual graph. The nodes of this graph are the mesh's triangles and there are edges between nodes if the faces are neighbors. The weights for the edges are the geodesic distances of the barycenters of the triangles. Examples of resulting samplings are shown in Figure 3. After placing the samples, we define the regions associated with the samples using a region growing algorithm. The algorithm is using the Dijkstra distance and grows the regions until the distance r is reached.

To ensure the geodesic disks cover the whole surface, we can use the distance ρ of the last sample placed by the furthest point sampling. If r is larger than ρ, then the disks cover the whole surface. In practice, we choose r much larger than ρ as we want the disks to have sufficient overlap.

Once the regions are defined, we compute m eigenfields for each region by solving the sparse eigenvalue problem (7). The choice

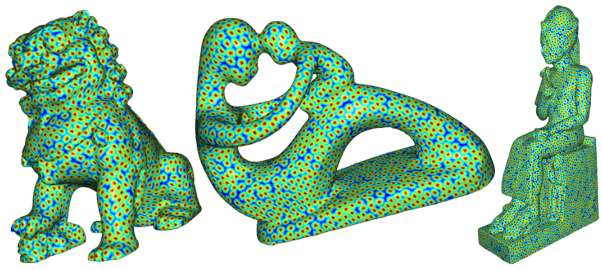

Figure 3: *Examples of farthest point samplings (1k-5k samples) constructed on various mesh models (50k-2m faces).*

of m depends on the type of field. For example, for vector fields and n-fields, we set $m = 2$, and, for (1,1)-tensors, we set $m = 3$. This choice is based on the multiplicity the lowest eigenvalue of the corresponding Laplace operator and our experimental results as discussed in Section 6. The size k_D of the sparse matrix occurring in the eigenvalue problem depends on the number of triangles belonging to the region. The resulting k_D-dimensional eigenvectors ϕ_i describe the fields in the regions. We lift the ϕ_is to vector fields Φ_i defined on the whole surface using (4) and stack the lifted fields Φ_i as the columns of a $k \times d$ matrix U. Here d denotes the total number of fields that are constructed, which is m times the number of regions. Since each Φ_i vanishes outside of its support region, the matrix U is sparse.

Scalability Our subspace construction is designed to be scalable, meaning that we want to be able to efficiently construct and store large subspaces on large meshes. Our motivation to aim for a scalable construction is that we want to be able to obtain subspaces that, on the one hand, include enough degrees of freedom to support general purpose design and processing tasks, and, on the other hand, allow to control the size of the optimization problem independently of the resolution of the meshes that are used.

One important feature that makes the method scalable is the localized support of the basis fields. For storing the basis, this means that the vector representing the basis fields are sparse vectors. The size of the support of the fields needs to be large enough such that there is sufficient overlap of each basis field with some other fields. On the other hand, a too large support is less efficient in terms of storage requirements. This means that if we construct two spaces with different dimension on the same surface, then the individual basis fields of the larger space will have a smaller support. Since the basis fields are stored as sparse vectors, this implies that the higher dimensional space requires more basis fields to be stored while each basis fields requires less storage. In our experiments, we found that the storage requirements for storing spaces of different dimension on the same mesh are approximately the same. This enables us to work with large subspaces with 500 to 5000 or even more dimensions. In addition, the method allows us to construct spaces on larger meshes, as the computation of the individual fields only requires solving an eigenvalue problem for a small region of the surface (the support of the field). These properties are advantages of our construction over eigenbases of Laplace operators, which have higher storage requirements, as a large dense matrix must be stored, and require higher computational costs for solving the large scale eigenvalue problems.

Adaptive subspaces Adaptivity can be integrated to the basis construction in a way that is simple to implement. The sampling method and the size of the regions depend on the weighted Dijkstra distance on the dual graph. For the uniform construction, the edge weights are chosen according to the geodesic distances of the barycenters of the triangles. To make the method adaptive, we change the edge weights in the dual graph. Changing the edge weights in some part of the surface affects the sampling and the sizes of the support regions of the basis fields. For example, when the weights are increased in some part of the surface, the sampling in the part of the surface becomes denser and the support regions of the basis fields decrease. The resulting subspace has more degrees of freedom in the part of the surface where the weight is increased, and, therefore, can better represent fields that have high frequency features and details in these areas. If the weights are reduced, fewer and larger regions are constructed in the corresponding part of the surface. For the subspaces, this means that fields with little detail in the corresponding part of the surface are represented more efficiently. The adaptive construction is simple to implement as after rescaling of the edge weights of the dual graph, the same algorithm as in the uniform case is executed. Scaling factors for the edges can be obtained for example from user input or an analysis of example fields.

5. Subspace methods

The main goal of our construction is to enable subspace methods for vector, n-vector and tensor field design and processing. In this section, we will discuss a model problem that we will later use as part of the evaluation of our subspace construction.

We consider the following least-squares problem

$$\min_{X \in \mathbb{R}^n} \left(\mu_S X^T S X + \mu_B X^T B X + \mu_C \|CX - c\|^2 \right), \tag{8}$$

where C is a matrix that defines weak constraints, c specifies the values of the constraints, the μs are a positive weights, M is the mass matrix, S is the stiffness matrix of the Laplacian and $B = SM^{-1}S$. The first two summands are the harmonic and biharmonic energies of X, which act as regularizers. This type of problem arises in field design tasks, for example, when fields are modeled with a stroke-based user interface. The resulting fields should align with the strokes but not follow them exactly. Other examples of applications that can be formulated as in (8) are smoothing of an input fields and extrapolating fields that is given only on parts of the surface to fields defined on the whole surface. For the smoothing application, C is the identity matrix and c the input field. For the stroke-based design and the extension of the field, C is a selector matrix that selects the vectors (n-vectors, tensors) of the parts of the surface where the input field is defined and c specifies the vectors of the field in these regions. The minimizer of (13) satisfies

$$(\mu_S S + \mu_B B + \mu_C C^T C) X = \mu_C C^T c. \tag{9}$$

The reduced minimization problem, which restrict the optimization to the subspace, is

$$\min_{x \in \mathbb{R}^d} \left(\mu_S x^T U^T S U x + \mu_B x^T U^T B U x + \mu_C \|CUx - c\|^2 \right). \tag{10}$$

The solution can be computed by solving the system

$$U^T (\mu_S S + \mu_B B + \mu_C C^T C) U x = \mu_C U^T C^T c. \tag{11}$$

The advantage of the reduced problem is that (11) is a sparse low-dimensional system. In particular, the system is independent of the resolution of the mesh and only depends on the dimension of the subspace. The scalability of our basis construction allows for working with spaces of several thousand dimensions, which provide a rich space for design and processing problems. At the same time, solving the reduced problems only takes few milliseconds which enables interaction and interactive steering of parameters. For example, a user can change the parameters μ_S, μ_B, and μ_C and receive

immediate feedback. In contrast, without reduction, any parameter adjustment requires solving a large-scale sparse linear system, which is prohibitive for interactive applications.

The reduction of the model problem (8) can be extended to include more features. For example, hard constraints can be efficiently included using a Schur complement approach. We refer to [BSEH17, BSEH18] for details. The subspaces can also be used to reduce the complexity of general non-linear problems. Reducing the problem's dimension lowers the computational cost of minimization steps and accelerates the convergence of solvers. However, a difference to the model problem is that the evaluations of a non-linear objective and its gradient and Hessian still depend on the complexity of the mesh. Methods for fast approximation of a non-linear objective and its derivatives have been proposed, for example, in the context of real-time simulation [AKJ08, vTSSH13, YLX*15, BEH18]. In this paper, we use the model problem (8) for evaluating the quality of the proposed subspace basis and leave the adaption of fast approximation schemes for non-linear objectives as future work.

6. Experiments

In this section, we discuss our experimental evaluation of our subspace construction focusing on scalability, approximation and adaptivity. We tested our approach on different meshes including some with low mesh quality exhibiting a large number of triangles with acute angles. Figure 4 shows examples of meshes we used.

Implementation We implemented our subspace constructions using the *Eigen* [GJ*10] and *LibIGL* [JP*16] libraries. The basis fields of the subspaces are constructed in parallel. To solve the local eigenproblems, we use the *SpectrA* library [Qiu15] with Cholmod's supernodal sparse Cholesky decomposition [CDHR08] being applied to solve the linear systems. CUDA's GPU-based *cuSparse* is employed to lift the fields from the reduced space to the full space, which requires matrix-vector multiplication with the sparse matrix U that stores the subspace basis. To solve the linear systems in the design and processing applications, we use Pardiso's symmetric indefinite factorization [KFS18].

Scalability In our experiments, we evaluated the memory requirements for storing the subspace basis and the computation times required for the construction of the basis for large subspaces and also for larger meshes. When reporting results, we state the subspace dimension we worked with. Since we compute a constant number of basis fields per region, 2 for vector fields and n-fields and 3 for (1,1)-tensor fields, the number of regions is half of the subspace dimension for vector and n-fields and a third for the tensor fields.

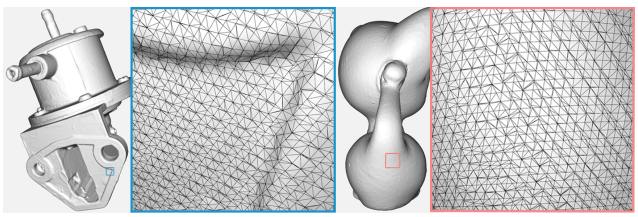

Figure 4: *We tested our approach on irregular meshes having many acute angles.*

In the first experiment, we constructed subspaces of different dimension d ranging from 500 to $20k$ on a mesh with $1m$ triangles and report the number of non-zero entries of the subspace bases as well as computational times required for basis construction. The data is summarized in Table 1. For each subspace dimension, we need to choose a proper value r for the radii of the geodesics disks. It is important to choose r large enough such that the individual vector fields can interact with their neighbors and information can spread. On the other hand, a too large value of r makes the basis less efficient as more storage is required. Explicitly, we set

$$r = \sqrt{\frac{\sigma A}{d\pi}}, \tag{12}$$

where A is the area of the surface, d the dimension of the subspace and σ a parameter, which we set to 40 for this experiment. The motivation for using this formula is that we want to find the radius r such that the combined area of all disks is σ times the area of the surface. To arrive at a simple formula, we replaced the average area of the geodesic disks by $r^2\pi$, which is the the area of the Euclidean disk of radius r. In this experiment, setting $\sigma = 40$ results in matrices U whose average number of non-zero entries per row is about 40. This means that in average every triangle is in the support of 40 basis fields.

The total time required for basis construction is listed in the fifth column of the table. The higher the dimension of the subspace, the more eigenvalue problems need to be solved. On the other hand, each of the eigenvalue problems is smaller as the support of the fields decreases. The total time for constructing the bases for the different subspaces on a regular desktop computer is between 3.5 and 7 minutes. The dimensions are between 500 and $20k$ and the underlying mesh has $1m$ triangles.

In the second experiment, we constructed $1k$, $2k$, and $10k$-dimensional subspaces on meshes with a number of triangles in the range of $274k$ to $2.21m$ and measured the time needed for basis construction. Experimental results are summarized in Table 2. As in the first experiment, we observe that the computation time

Subs. dim.	Basis construction (in seconds) Sampling	Construct patches	Solve eigenp.	Total	Sparsity #nnz U	#nnz $\bar{S}$
500	1.6	82.9	168.8	253.4	80.4m	78.3k
1k	1.9	84.0	125.9	211.8	80.4m	163.0k
2k	2.4	80.5	126.2	209.1	80.4m	333.0k
5k	3.5	77.2	152.1	232.8	80.4m	854.5k
10k	5.3	80.1	182.6	267.9	80.5m	1.7m
20k	8.4	103.1	283.7	395.3	80.5m	3.5m

Table 1: *Scalability analysis of our basis construction. Computation times and numbers of non-zero entries (#nnz) of the matrices storing the basis (U) and the restricted stiffness matrices ($\bar{S}$) for subspaces of different dimension on the Bimba model with* $1m$ *triangles are shown.*

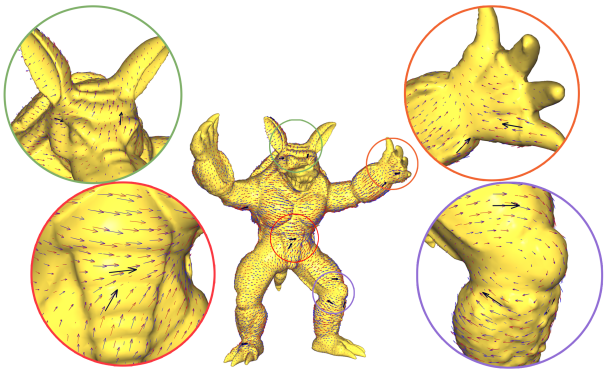

Figure 5: *Visual comparison of a reference vector fields (blue) and its projection to a 2k-dimensional subspace (red). The relative L^2 approximation error is 1.93×10^{-2}.*

required for the construction of subspaces of different dimension on the same mesh changes only slightly. When comparing the construction time for the different meshes, we observe an increase in computation time that is approximately linear in the number of triangles.

Approximation In addition to the scalability of the basis construction, the approximation quality of the resulting subspace is important. In the first series of experiments, we evaluate the approximation quality by projecting a set of tangential vector fields to the subspace and computing the relative L^2-norm of the difference between the input field and projected field. For different meshes, we created sets of 50 test fields by placing 10-50 interpolation constraints at random locations on the surfaces and using the vector field construction method from [BSEH17] to generate fields interpolating the constraints. For an input field X on the surface, the projection x to the subspace with basis U is defined as the minimizer of the quadratic objective

$$\|X - Ux\|_M^2 . \tag{13}$$

The relative L^2-error is $\|X - Ux\|_M / \|X\|_M$. Table 3 shows the resulting average errors for 50 test fields on three different meshes. For all three models, the relative L^2 error using a 2k-dimensional subspace is 2 percent. Figure 5 shows an overlay of a test field and its projection on the Armadillo mesh. In addition to evaluating the projection error, we also compare the solutions of the optimization problem (8) and solution of the corresponding reduced problem (10). The relative error of the optimization problem is between 2 and 4 percent as listed in Table 3 (rows labeled 'Minim.'). Based on the comparisons to alternatives and variations of our construction, which are discussed in Section 7, we consider this a very good approximation quality.

In the second experiment, we measured how the relative L^2 approximation error changes with increasing size of the subspace. The test fields were generated in the same way as in the previous experiment. We constructed 500-20k dimensional subspaces on the Fertility mesh (483k triangles). To set the radii of the geodesic disk, we use equation (12) and set $\sigma = 80$ for all spaces. Results are shown in Figure 8. The results illustrate that the approximation error can be reduced when subspaces of higher dimension are used.

Mesh	#Faces	Dim.	Basis Construction (in seconds)			
			Sampl.	Patches	Eig.solv.	Total
Kitten	274k	1k	0.54	18.32	38.70	57.56
		2k	0.67	18.88	34.83	54.37
		10k	1.43	22.45	45.53	69.41
Fertility	483k	1k	1.10	35.63	64.42	101.16
		2k	1.33	32.52	64.97	98.82
		10k	2.70	41.67	77.95	122.33
Bimba	1m	1k	1.94	83.98	125.90	211.83
		2k	2.41	80.48	126.20	209.09
		10k	5.25	80.06	182.56	267.87
Ramses	1.65m	1k	4.50	178.81	261.89	445.20
		2k	5.20	168.17	222.36	395.73
		10k	10.70	159.61	347.29	517.61
Isidore horse	2.21m	1k	5.02	213.02	366.57	584.60
		2k	6.10	203.36	286.89	496.35
		10k	13.25	199.49	457.64	670.38

Table 2: *Timings for the constructions of subspaces of different dimension on various meshes are shown. Computation times for farthest point sampling (Sampl.), construction of the local patches and corresponding matrices (Patches), solving the eigenproblems (Eig. solv.), and the total time for all three steps (Total) are shown.*

Mesh	Method	Ours	Bihar-monic	Variant of ours	Patched Eigenf.	Grad-ients
Arma-dillo	L^2-proj.	0.02	0.05	0.06	0.14	0.16
	Minim.	0.04	0.21	0.11	0.25	0.48
Chinese Lion	L^2-proj.	0.02	0.07	0.06	0.10	0.19
	Minim.	0.03	0.18	0.09	0.18	0.46
Fertility	L^2-proj.	0.02	0.03	0.04	0.71	0.59
	Minim.	0.02	0.09	0.10	0.71	0.65

Table 3: *Approximation errors of L^2-projection of vector fields to a 2k-dimensional subspaces and solutions of the optimization problem (8) for subspaces resulting from the proposed subspace construction and four possible alternative constructions.*

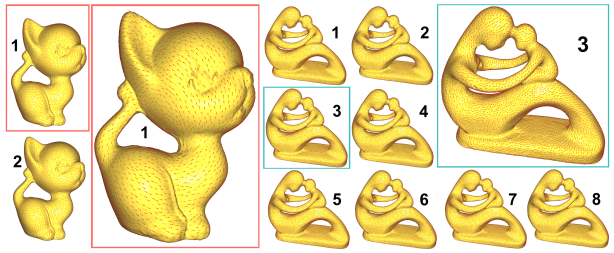

Figure 6: *Approximations of harmonic fields in subspaces on models with non-trivial genus (Kitten=1 and Fertility=4).*

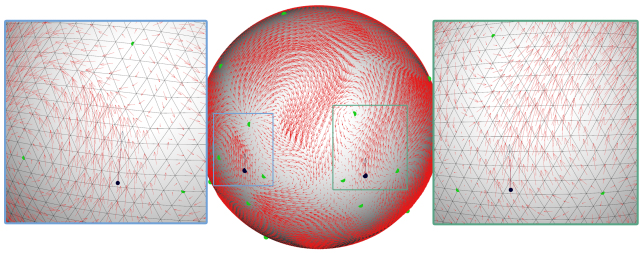

Figure 7: *Example showing the placement of singularities in the subspace.*

The approximation error will vanish once the subspace dimension equals the dimension of the full space. As increasing the dimension of the subspace causes higher computational cost for constructing the subspace and solving the reduced problems, in practice one needs to find a compromise between expressiveness of the space on the one hand and computational costs on the other hand.

In the third experiment, we tested whether the subspaces of vector fields we construct include approximations of the harmonic fields. On smooth surfaces the harmonic fields are the eigenfields of the Hodge–Laplace operator with vanishing eigenvalue. A surface of genus g has $2g$ linearly independent harmonic fields. The discrete Hodge–Laplacian (1) is designed such that it preserves this structure and has a $2g$-dimensional kernel of discrete harmonic fields. We wanted to test whether our subspaces contain approximations of these fields. In our experiments, we obtained $2g$ approximate harmonic fields with very small, though not vanishing, eigenvalues. Examples of these fields are shown in Figure 6. Additionally, Figure 19 shows plots of the approximate eigenvalues for some meshes, two genus zero meshes and the genus one Kitten model. In the lower left corner, the first eigenvalues are shown. For the Kitten model, the first two values are are the eigenvalues of the approximate harmonic fields.

In the fourth experiment, we evaluated the capabilities of the subspaces to support the placement of singularities. In Figure 7, we show an example of a vector field in a 200-dimensional subspace on which singularities at certain locations on the surface are enforced using the approach discussed in [BSEH18].

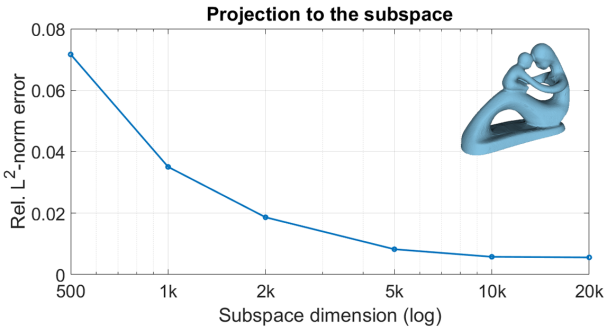

Figure 8: *The relative L^2 approximation error for subspaces of different dimension is shown.*

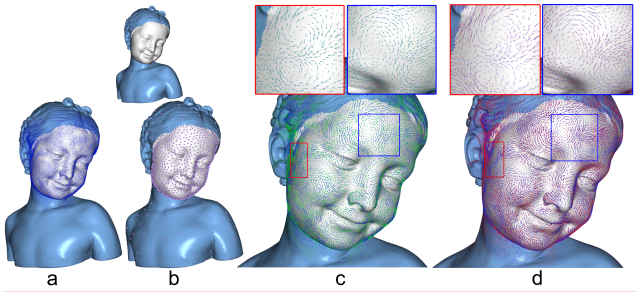

Figure 9: *Results of an experiment with our adaptive subspace construction. The figure shows (on top left) the Bimba mesh with* $1m$ *triangles and a selected region on the surface, (a) a vector field on the surface, (b) the sampling used for the adaptive subspace construction, (c) an overlay of the vector field in blue and its L^2-projection to an adaptive subspace in green, and (d) an overlay of the vector field in blue and its L^2-projection to a uniform subspace in red.*

Adaptivity To explore the benefits of adaptive subspaces, we conducted an experiment in which a vector field, which has many features concentrated in some area of a surface and almost vanishes away from that region, is projected to a uniform and an adaptive subspace. The adaptive subspace is constructed by rescaling the weights of the dual graph in a region of the surface, which we defined by hand. Results of the experiment are shown in Figure 9. The figure illustrates the benefits of adaptive subspaces for the approximation of the fields. As a quantitative evaluation, we computed the relative L^2 approximation errors. The adaptive subspace yields a relative L^2 error of 9.2×10^{-2}, which compares to 35.2×10^{-2} for the uniform subspace.

7. Comparisons

During the development of the proposed subspace construction, we implemented and tested various possible alternatives. In this section, we provide some comparisons of the proposed and possible alternative constructions. In addition, we compare our subspace construction with Laplace eigenfields.

Alternative constructions We report approximation results for four alternative constructions in Table 3. The first alternative is to use not only the lowest two eigenfields for every geodesic disk, but more. The column "Variant of ours" shows results for the case that the lowest 10 eigenfields are used for each geodesic disk. In order to get a fair comparison, fewer disks are used in total such that the dimensions of the subspaces are the same. The second alternative is to solve two biharmonic problems on each geodesic disk instead of the eigenproblems. For the biharmonic problems, we set interpolation constraints: we specify a unit vector in the center triangle of the geodesic disk and enforce that the field vanishes for all triangles not in the geodesic disk. The latter constraint implements zero Dirichlet and Neumann boundary conditions for the biharmonic problem.

Two fields are generated by specifying orthogonal vectors in the center triangle. For details on how to solve biharmonic problems with interpolation constraints for tangential vector fields, we refer to [BSEH17]. The results for this construction are listed in the column labeled "Biharmonic". The third construction makes use of radial basic functions on surfaces as described in [NBH18]. The radial basis functions are defined on the surface and are localized. To obtain vector fields from the radial basis functions, we compute the gradients and their co-gradients. This construction is labeled "Gradients" in the table. The fourth alternative also uses the radial basis functions. The idea is to compute two eigenfields of the whole surface and to use the radial basis functions to scale the vectors of the fields. This results in localized fields, which we use as subspace bases. The approximation results for these bases are listed in the column labeled "Patched Eigenf." in the table. The experiments we performed are the same tests as discussed in the paragraph *Approximation* of Section 6. 2k-dimensional subspaces were used for all tests. In our experiments, the proposed basis construction outperformed the alternative construction by a large margin as also documented in Table 3.

In the second experiment, we computed approximations of the lowest eigenvalues of the Laplace operator in subspaces constructed with the different schemes. For quantitative evaluations, we compare the results to the eigenvalues computed in the full space. Results are shown in Figure 11. Our construction very closely matches the true eigenvalues and outperforms all other constructions. Examples of basis fields resulting from the different constructions are shown in Figure 10. Though at first sight the fields obtained solving a biharmonic problem look similar to the field resulting from the proposed construction. A closer look reveals differences in the scaling of the vectors, which turns out to be important for the performance of the resulting subspaces.

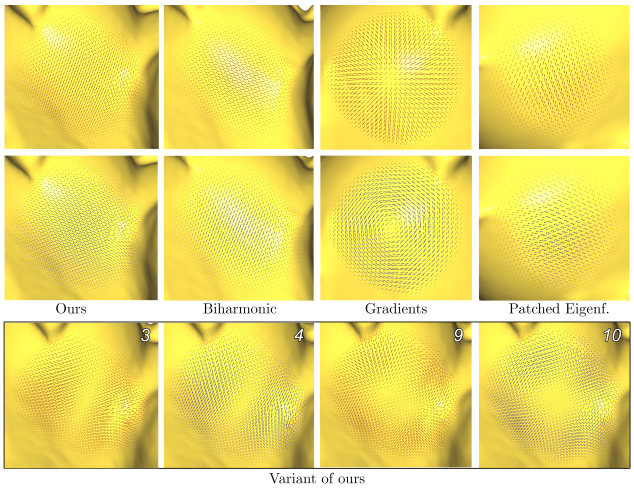

Figure 10: *Examples of basis fields resulting from the alternative constructions we compare to in Table 3. For the construction that involves 10 eigenfunctions per geodesic disk ("Variant of ours"), the 3rd, 4th, 9th and 10th eigenfields on a geodesic disk are shown.*

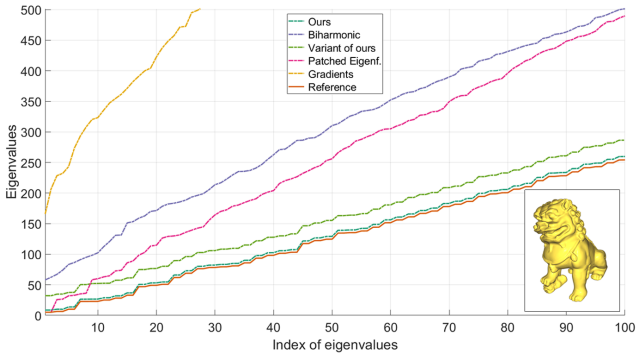

Figure 11: *Approximation of eigenvalues using various alternatives of locally supported basis functions.*

Mesh coarsening In addition to the four alternatives discussed above, we can also use mesh coarsening for constructing subspaces. We compared the performance of the proposed construction to a mesh coarsening scheme that we developed. As for the approximation experiments in Section 6, we evaluated the relative approximation error for L^2 projection and the residual for the minimization problem (8). Results are shown in Table 4. In all our experiments, the results of the proposed method are significantly better than those of the coarsening approach. Figure 12 illustrates the coarsening-based subspace construction and includes a visual comparison of results obtained with our construction and the coarsening-based construction. The coarsening-based approach starts with coarsening of the mesh. To map a vector field on the coarse mesh to the fine mesh, we find for every triangle on the fine mesh the closest triangle of the coarse mesh and project the vector of the coarse mesh to the plane containing the fine triangle. The

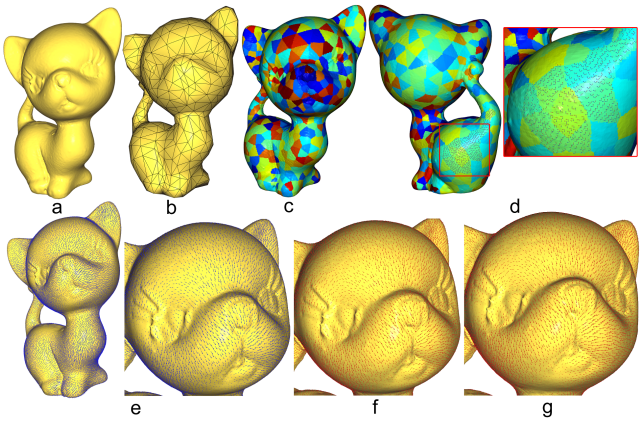

Figure 12: *Comparison to a subspace construction based on mesh coarsening. The coarsening approach is illustrated in images (a)-(d). For comparison, a smooth tangential field (e) is projected onto the coarsening-based subspace (f), and onto the subspace proposed in this paper (g).*

Model	Basis	L^2 Proj.	Mimimization
Kitten	Ours	1.77×10^{-2}	2.75×10^{-2}
	Coarsening	14.8×10^{-2}	48.5×10^{-2}
CDragon	Ours	6.67×10^{-2}	3.12×10^{-2}
	Coarsening	25.6×10^{-2}	70.0×10^{-2}
Fertility	Ours	1.76×10^{-2}	2.27×10^{-2}
	Coarsening	20.0×10^{-2}	51.2×10^{-2}

Table 4: *The comparison of the proposed subspace construction and a possible alternative construction based on mesh coarsening.*

images of the canonical basis fields on the coarse mesh provide us with a subspace on the fine mesh.

Laplace eigenbasis To our knowledge, the only alternative construction of subspace of spaces of tangential vector and n-vector fields on surfaces are the eigenfields of vector field Laplacians and n-field Laplacians. In our experiments, we compared the eigenbases against our construction. The approximation quality of the eigenbases also serves as a baseline as using eigenmodes is a common subspace construction method in other contexts. While low-dimensional subspaces constructed from eigenfields can approximate smooth fields well, the eigenfields are globally supported. This means storing an eigenbasis is expensive as a large dense matrix must be stored. As a result, the eigenbasis cannot compete with the proposed construction in terms of scalability. In addition to storage requirements, the computation of the eigenfield is more costly than the local eigenproblems we need to solve for our construction. Table 5 shows results of our experiments that compare L^2 approximation error and residuals of the minimization problem (8) as well as the number of non-zero entries of the matrix storing the basis. For our construction the radius of the geodesic balls is set using formula (12) with $\sigma = 40$. The results show that an eigenbasis with just 40 eigenfields has similar storage requirement as our basis spanning a 2000-dimensional subspace. Concerning the approximation quality, however, a 2000d subspace resulting from our construction is much better than a 40d space spanned by the lowest eigenfields. To achieve a comparable approximation quality, more than 600 eigenfields are necessary. The storage requirements for such an eigenbasis is more than an order of magnitude higher than for 2000 basis fields computed with our construction.

Model	#Faces	Basis	Dim.	#NNZ	L^2 proj. error	Minimization
Kitten	274k	Eigenf.	40	$2.19 \cdot 10^7$	0.42	0.40
			250	$1.37 \cdot 10^8$	0.06	0.06
			667	$3.66 \cdot 10^8$	0.02	0.02
		Ours	2000	$2.19 \cdot 10^7$	0.02	0.03
Fertility	483k	Eigenf.	40	$3.87 \cdot 10^7$	0.37	0.36
			250	$2.42 \cdot 10^8$	0.21	0.21
			667	$6.45 \cdot 10^8$	Mem. bound	
		Ours	2000	$3.87 \cdot 10^7$	0.02	0.02
Bimba	1m	Eigenf.	40	$8.04 \cdot 10^7$	0.33	0.33
			250	$5.03 \cdot 10^9$	Mem. bound	
			667	$1.34 \cdot 10^9$	Mem. bound	
		Ours	2000	$8.04 \cdot 10^7$	0.02	0.03

Table 5: *Comparison to eigenfields. Given the same storage, our locally supported bases outperform the eigenbases. Our method can achieve similar performance while requiring less storage.*

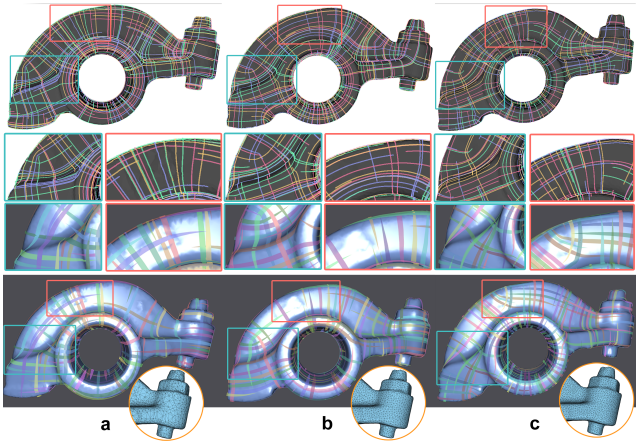

Figure 13: *Comparison of smooth, curvature-aligned 4-fields computed on different meshes that approximate the same surface using the proposed techniques (top row) and the approach proposed in [JTPSH15].*

Comparison to Instant Field Aligned Meshes In this paragraph, we discuss how our approach compares to the approach for real-time n-fields design that was introduced in [JTPSH15]. They use an extrinsic energy, which combines fairness of the field and alignment to the surface curvature directions, to guide their n-field construction. In a precomputation, a multilevel hierarchy is constructed. Then, for field design, minimization steps with respect to the extrinsic energy are performed on the different levels of the hierarchy from coarse to fine. When used as an interactive design tool, the number of minimization steps per level are fixed in order to get fast responses. Due to the strict time constraint, the number of steps is typically not sufficient for the solver to converge to a minimum. We show a comparison of results in Figure 13. In the shown example, we compute smooth curvature aligned 4-fields on different meshes that approximate the same rockerarm surface. One can see that our approach produces more consistent fields for the different meshes compared to the approach from [JTPSH15]. We refer the reader to [BSEH18] for additional evaluation of the Instant Field Aligned Meshes approach.

8. Applications

In this section, we discuss some applications of the proposed subspace construction.

Vector fields design The tangential vector field design approach proposed in [FSDH07] computes fields that are on the one hand smooth and on the other hand align with input data such as strokes

drawn by a user and an input field. This is modeled as a minimization problem similar to problem (8). Computing the minimizer requires solving a sparse linear system, whose size depends on the number of triangles of the surface. For fast solving, a sparse Cholesky factorization of the system is computed once and used for solving the systems. Cholesky updates are applied to update the system's matrix when the user changes the constraints. Though backsubstitution and the Cholesky updates are fast, they still scale with size of the mesh. As a result this approach enables interactive design only for a limited mesh resolution. Moreover, the interaction is limited to those operations that can be treated with Cholesky updates of the system's matrix. In contrast to this, the reduced system (10) is independent of the mesh's resolution. The proposed subspace construction allows the construction of subspaces that are large and therefore provide a rich modeling spaces to designers while keeping the computational cost for solving the reduced systems low. Figure 15 shows results of a modeling session using our approach, on the Antique Head model of $1.3m$ faces with a $2k$-dimensional subspace. Our subspace methods provides interactive feedback. Timings are: $45ms$ for computing a factorization of the reduced system, which includes the harmonic and biharmonic energies, $1.4ms$ for solving a system using the factorization and $23ms$ for mapping the reduced coordinates to a vector field on the surface. For comparison, we list corresponding timings for the unreduced system: $43s$ for computing the sparse factorization and $2.2s$ for solving a system using the factorization.

Fur design An application of the vector field design is fur editing. We used our subspaces in the fur editing approach proposed in [BSEH17]. Results for a modeling session that involves a Wolf mesh with $1m$ triangles and a 2000-dimensional subspace are shown in Figure 14. The tools allows to specify hard constraints and a Schur complement approach is used for solving the resulting reduced linear systems. The system requires less than $40ms$ for computing a solution when the interpolation constraints are modified.

***n*-field design** The n-field design approach proposed in [BSEH18] computes fields that minimize a biharmonic energy subject to interpolation constraints imposed by a user. Additionally, a penalty for deviation from an input field can be included to the optimization. The approach requires solving a linear system and a reduction of the problem using eigenfields of the biharmonic energy is discussed in the paper. Since our subspace construction is more efficient for larger meshes and larger subspaces than the eigenbasis, see Section 7, our construction can be used to improve the scalability of the n-field design approach. Figure 16 shows results for n-fields design using our subspaces. In our experiments, the design tool achieved 30 fps when working with a $2k$-dimensional subspace on a model with $1m$ triangles.

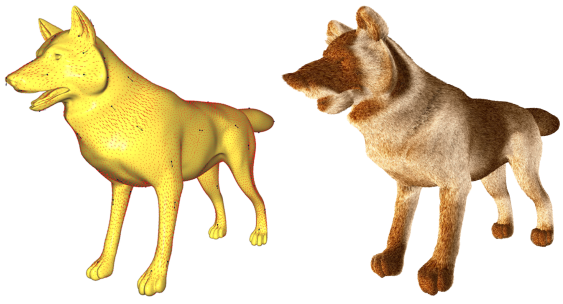

Figure 14: *Interactive vector field editing on the Wolf mesh (left, 1m triangles, 2k dimensional subspace) and the resulting fur rendering (right).*

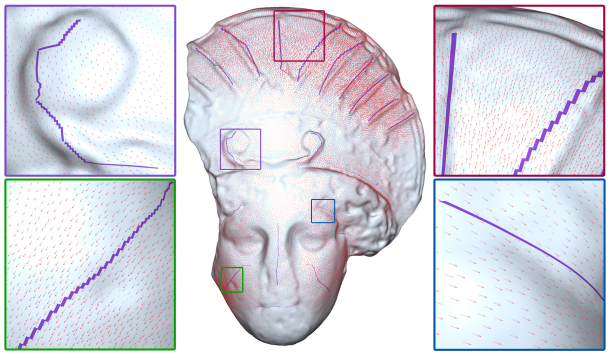

Figure 15: *Interactive stroke-based vector field design on the Antique Head model (1.3m triangles, 2k subspace).*

Hatching n-field design can be used to control the stroke directions of line art renderings. We used our subspaces in combination with the approach for controlling hatchings of surfaces from [BEH18]. The approach uses 2-fields to control hatching directions. The fields are first aligned to the maximum principal curvature directions of the surface. Then, the users can use interpolation constraints to modify the 2-field. Results are shown in Figure 17.

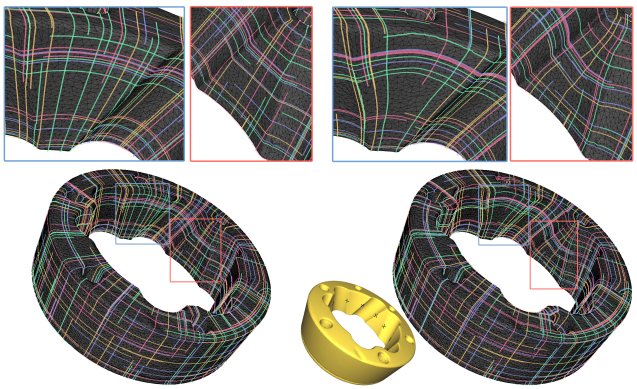

Figure 16: *Example of 4-field design with interpolation constraints. Fields can be aligned to the curvature directions (left). Additional interpolation constraints allow users to modify the 4-field such that it better aligns to surface features (right). Our tool provides interactive responses when interpolation constraints are added, removed or modified.*

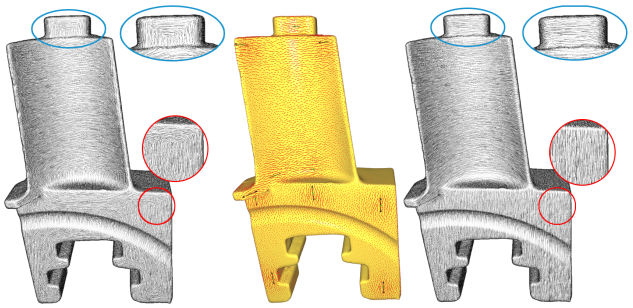

Figure 17: *Hatching of the Blade model (390k faces) generated from 2-fields aligned to the maximal principal curvature (left). Modified fields, subject to interpolation constraints (right). The constraints are shown in the middle.*

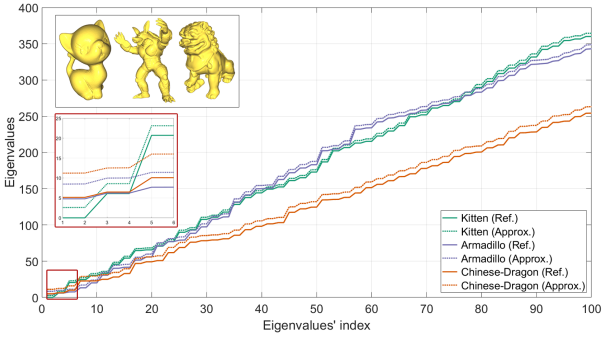

Figure 19: *Approximation of eigenvalues of the Hodge–Laplace operator. For different meshes, approximated and reference eigenvalues are shown.*

Tensor field smoothing We implemented a tensor field smoothing tool that minimizes a weighted sum of a function penalizing the deviation from the input field and the harmonic and biharmomic energies for tensor fields as fairness energies. The resulting optimization problem is analogous to problem (8). The tool uses our subspace construction to build a subspace and solves the reduced problem. This allows users to interactively adjust the weights for the three terms. To realize this, we precompute the reduced matrices for the three terms during the preprocessing stage, and, in the online stage, we build the weighted sum of the matrices and solve the resulting sparse linear system, whose size equals the dimension of the subspace.

We used the tool for smoothing the shape operator, whose eigenvectors are the principal curvature directions. Results are shown in Figure 18. We found this a useful tool for curvature computations, which usually need to be smoothed and require users to specify how strongly the field should be smoothed. With the tool, users receive immediate feedback when adjusting the smoothing parameters. For example, in the case of the Oil Pump model (1.1m faces) shown in Figure 18, our reduced system (using a 2k dimensional subspace) requires 45ms to compute a factorization, 1.5ms to solve using the factorization, and 55ms to lift the reduced solution to a tensor field defined on the whole surface. For comparison, the timings for solving the unreduced system are: 49s to compute a factorization and 2.7s to solve using the factorization. We want to emphasize that many operations, such as changing the weights for the harmonic or biharmonic energy, require computing a new factorization as the system's matrix changes.

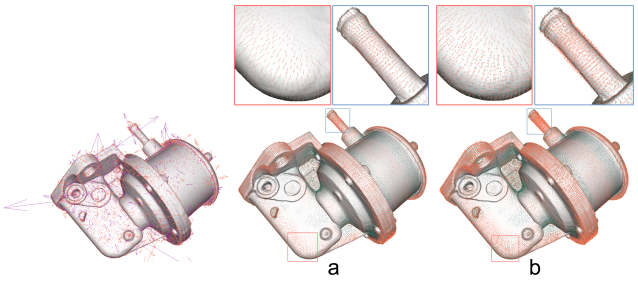

Figure 18: *Results of our tensor smoothing tool. Curvatures computed on a mesh with 1m triangles (left). Smoothing results for different parameter settings (middle and right). When parameters are changed, new solutions are computed at interactive rates.*

While we show an example, in which the user can only modify the weights, a more sophisticated interactive tool that allows users to specify different weights for different areas of the surface can be realized in a similar way. Another possible extension of this smoothing method would be a fast non-linear smoothing scheme that iteratively solves linear systems in the subspace.

Laplace spectrum Our subspaces can be used to efficiently compute approximations of the eigenvalues and eigenfields of Laplace operators. For computing approximations of m eigenfields, we construct a 2m-dimensional subspace and compute the restricted eigenvalue problem in the subspace. This technique has been recently introduced for the case of eigenfunction of the Laplace–Beltrami operator in [NBH18]. We refer to this paper for a description of the reduced eigenvalue problem, which is analogous to the reduced eigenvalue problem for eigenfields. Table 6 compares timings for solving the reduced and unreduced eigenproblems. Figure 19 compares the reference and approximated eigenvalues.

Model	Faces	Basis	Red. Syst.	Solve Eigenp.	Ref.	Speed up
Armadillo	86k	9	3.5	3.4	419	26.1
Ch.Dragon	255k	42	10.1	3.3	1376	25.0
Fertility	483k	99	17.7	3.4	Mem.	Inf.
Ramses	1.6m	401	84.4	3.3		Inf.
Neptune	4.0m	1200	212.6	4.8	bound	Inf.

Table 6: *Timings (in seconds) for the approximation of 500 eigenfields are shown (3-5th) columns). For comparison timings for computing the unreduced reference eigenvalues (6th column) using MATLAB's sparse eigensolver are shown.*

9. Conclusion

We introduce a construction of subspaces of tangential vector, n-vector, and tensor fields that is scalable and results in subspaces that can approximate smooth fields well. The construction can easily be extended to a construction of adaptive subspaces. We experimentally evaluate the approach and justify our construction by comparing it to possible alternative constructions. Finally, we discuss applications of our approach.

Challenges and limitations Our goal is to develop the techniques that enable interactive field design and processing tools that work on large meshes. The proposed subspace construction takes a step in this direction by enabling us to decouple the resolution of the meshes from the degrees of freedom used for the design and processing problems. In this work, we limit our focus to optimization problems with quadratic objectives. For more general problems, additional techniques need to be developed that allow us to approximate the objective and its gradients at a cost that does not depend on the mesh resolution. Developing such techniques for field design and processing problems poses interesting challenges that are beyond the scope of this paper. Another potential use of the proposed fields would be to build subspaces for a multilevel solver for problems involving tangential fields. Due to the scalability of our approach, it could be used to build subspaces of different resolution in which systems are solved and solutions are propagated. Finally, our approach could be extended to include more sophisticated boundary conditions for the eigenvalue problems we use for field construction. For example, boundary conditions that fit to the Hodge decomposition of vector fields have been proposed in [PP16].

Acknowledgements

We thank Leonardo Scandolo for his help with the renderings used for the applications and the anonymous reviewers for their constructive feedback. This project is partly supported by the Indonesia Endowment Fund for Education (LPDP) through a doctoral scholarship for Ahmad Nasikun. For our experiments, we used models from the Stanford Computer Graphics Laboratory (Stanford 3D Scanning Repository), the TOSCA [BBK08] data sets, INRIA, Turbosquid and the Aim@Shape repository.

References

[ABCCO13] AZENCOT O., BEN-CHEN M., CHAZAL F., OVSJANIKOV M.: An operator approach to tangent vector field processing. Comp. Graph. Forum 32, 5 (2013), 73–82. 2

[AKJ08] AN S. S., KIM T., JAMES D. L.: Optimizing cubature for efficient integration of subspace deformations. ACM Trans. Graph. 27, 5 (2008), 165:1–165:10. 2, 6

[AOCBC15] AZENCOT O., OVSJANIKOV M., CHAZAL F., BEN-CHEN M.: Discrete derivatives of vector fields on surfaces – an operator approach. ACM Trans. Graph. 34, 3 (2015), 29:1–29:13. 2

[AVW*15] AZENCOT O., VANTZOS O., WARDETZKY M., RUMPF M., BEN-CHEN M.: Functional thin films on surfaces. In Symposium on Computer Animation (2015), pp. 137–146. 2

[AWO*14] AZENCOT O., WEISSMANN S., OVSJANIKOV M., WARDETZKY M., BEN-CHEN M.: Functional fluids on surfaces. In Comp. Graph. Forum (2014), vol. 33, Wiley Online Library. 2

[BBK08] BRONSTEIN A. M., BRONSTEIN M. M., KIMMEL R.: Numerical geometry of non-rigid shapes. Springer, 2008. 13

[BdSP09] BARBIČ J., DA SILVA M., POPOVIĆ J.: Deformable object animation using reduced optimal control. ACM Trans. Graph. 28, 3 (2009), 53:1–53:9. 2

[BEH18] BRANDT C., EISEMANN E., HILDEBRANDT K.: Hyper-reduced projective dynamics. ACM Trans. Graph. 37, 4 (2018), 80:1–80:13. 2, 6, 11

[BJ05] BARBIČ J., JAMES D. L.: Real-time subspace integration for St. Venant-Kirchhoff deformable models. ACM Trans. Graph. 24, 3 (2005), 982–990. 2

[BSEH17] BRANDT C., SCANDOLO L., EISEMANN E., HILDEBRANDT K.: Spectral processing of tangential vector fields. Comp. Graph. Forum 36, 6 (2017), 338–353. 2, 3, 6, 7, 9, 11

[BSEH18] BRANDT C., SCANDOLO L., EISEMANN E., HILDEBRANDT K.: Modeling n-symmetry vector fields using higher-order energies. ACM Trans. Graph. 37, 2 (2018), 18:1–18:18. 2, 3, 6, 8, 10, 11

[BSEH19] BRANDT C., SCANDOLO L., EISEMANN E., HILDEBRANDT K.: The reduced immersed method for real-time fluid-elastic solid interaction and contact simulation. ACM Trans. Graph. 38, 6 (2019). 2

[BvTH16] BRANDT C., VON TYCOWICZ C., HILDEBRANDT K.: Geometric flows of curves in shape space for processing motion of deformable objects. Comp. Graph. Forum 35, 2 (2016). 2

[BZK09] BOMMES D., ZIMMER H., KOBBELT L.: Mixed-integer quadrangulation. ACM Trans. Graph. 28, 3 (2009), 77:1–77:10. 2

[CDHR08] CHEN Y., DAVIS T. A., HAGER W. W., RAJAMANICKAM S.: Algorithm 887: CHOLMOD, supernodal sparse Cholesky factorization and update/downdate, Oct. 2008. 6

[CDS10] CRANE K., DESBRUN M., SCHRÖDER P.: Trivial connections on discrete surfaces. Comp. Graph. Forum 29, 5 (2010), 1525–1533. 2

[CSK18] CUI Q., SEN P., KIM T.: Scalable Laplacian eigenfluids. ACM Trans. Graph. 37, 4 (2018), 87:1–87:12. 2

[CV20] CUSTERS B., VAXMAN A.: Subdivision directional fields. ACM Trans. Graph. 39, 2 (2020). 2

[CYZL14] CHI M., YAO C., ZHANG E., LEE T.: Optical illusion shape texturing using repeated asymmetric patterns. The Visual Computer 30, 6-8 (2014). 2

[CZZ14] CHAI M., ZHENG C., ZHOU K.: A reduced model for interactive hairs. ACM Trans. Graph. 33, 4 (2014), 124:1–124:11. 2

[dGDMD16] DE GOES F., DESBRUN M., MEYER M., DEROSE T.: Subdivision exterior calculus for geometry processing. ACM Trans. Graph. 35, 4 (2016), 133:1–133:11. 2

[dGDT15] DE GOES F., DESBRUN M., TONG Y.: Vector field processing on triangle meshes. In SIGGRAPH Asia 2015 Courses (2015), pp. 17:1–17:48. 2, 3

[dGLB*14] DE GOES F., LIU B., BUDNINSKIY M., TONG Y., DESBRUN M.: Discrete 2-tensor fields on triangulations. Comput. Graph. Forum 33, 5 (2014), 13–24. 2

[DHLM05] DESBRUN M., HIRANI A., LEOK M., MARSDEN J.: Discrete exterior calculus. preprint, arXiv:math.DG/0508341, 2005. 2

[DVPSH14] DIAMANTI O., VAXMAN A., PANOZZO D., SORKINE-HORNUNG O.: Designing N-PolyVector fields with complex polynomials. Comp. Graph. Forum 33, 5 (2014), 1–11. 2, 3

[DVPSH15] DIAMANTI O., VAXMAN A., PANOZZO D., SORKINE-HORNUNG O.: Integrable PolyVector fields. ACM Trans. Graph. 34, 4 (2015), 38:1–38:12. 2

[FSDH07] FISHER M., SCHRÖDER P., DESBRUN M., HOPPE H.: Design of tangent vector fields. In ACM Trans. Graph. (2007), vol. 26, ACM, p. 56. 2, 10

[GJ*10] GUENNEBAUD G., JACOB B., ET AL.: Eigen v3. http://eigen.tuxfamily.org, 2010. 6

[HCW19] HEFETZ E. F., CHIEN E., WEBER O.: A subspace method for fast locally injective harmonic mapping. Comput. Graph. Forum 38, 2 (2019), 105–119. 2

[HJ16] HUANG Z., JU T.: Extrinsically smooth direction fields. Computers & Graphics 58 (2016), 109–117. 2

[HSL*06] HUANG J., SHI X., LIU X., ZHOU K., WEI L., TENG S., BAO H., GUO B., SHUM H.: Subspace gradient domain mesh deformation. ACM Trans. Graph. 25, 3 (2006), 1126–1134. 2

[HSvTP11] HILDEBRANDT K., SCHULZ C., VON TYCOWICZ C., POLTHIER K.: Interactive surface modeling using modal analysis. ACM Trans. Graph. 30, 5 (2011), 119:1–119:11. 2

[HSvTP12] HILDEBRANDT K., SCHULZ C., VON TYCOWICZ C., POLTHIER K.: Interactive spacetime control of deformable objects. ACM Trans Graph 31, 4 (July 2012), 71:1–71:8. 2

[HTC*14] HAHN F., THOMASZEWSKI B., COROS S., SUMNER R. W., COLE F., MEYER M., DEROSE T., GROSS M. H.: Subspace clothing simulation using adaptive bases. ACM Trans. Graph. 33, 4 (2014), 105:1–105:9. 2

[HZ00] HERTZMANN A., ZORIN D.: Illustrating smooth surfaces. In Proc. SIGGRAPH (2000). 2

[IBB15] IARUSSI E., BOMMES D., BOUSSEAU A.: Bendfields: Regularized curvature fields from rough concept sketches. ACM Trans. Graph. 34, 3 (2015). 2

[JBK*12] JACOBSON A., BARAN I., KAVAN L., POPOVIĆ J., SORKINE O.: Fast automatic skinning transformations. ACM Trans. Graph. 31, 4 (2012), 77:1–77:10. 2

[JP*16] JACOBSON A., PANOZZO D., ET AL.: libigl: A simple C++ geometry processing library, 2016. http://libigl.github.io/libigl/. 6

[JTPSH15] JAKOB W., TARINI M., PANOZZO D., SORKINE-HORNUNG O.: Instant field-aligned meshes. ACM Trans. Graph. 34, 4 (2015), 189:1–189:15. 2, 10

[KCPS13] KNÖPPEL F., CRANE K., PINKALL U., SCHRÖDER P.: Globally optimal direction fields. ACM Trans. Graph. 32, 4 (2013), 59:1–59:10. 2, 3

[KCPS15] KNÖPPEL F., CRANE K., PINKALL U., SCHRÖDER P.: Stripe patterns on surfaces. ACM Trans. Graph. 34 (2015). 2

[KFS18] KOUROUNIS D., FUCHS A., SCHENK O.: Towards the next generation of multiperiod optimal power flow solvers. IEEE Transactions on Power Systems PP, 99 (2018), 1–10. 6

[KNP07] KÄLBERER F., NIESER M., POLTHIER K.: Quadcover - surface parameterization using branched coverings. Comp. Graph. Forum 26, 3 (2007). 2

[LJF16] LU W., JIN N., FEDKIW R.: Two-way coupling of fluids to reduced deformable bodies. In ACM SIGGRAPH/Eurographics Symposium on Computer Animation (2016), pp. 67–76. 2

[LJX*10] LAI Y.-K., JIN M., XIE X., HE Y., PALACIOS J., ZHANG E., HU S.-M., GU X.: Metric-driven rosy field design and remeshing. IEEE Trans. Vis. Comput. Graph. 16, 1 (2010). 2

[LLZ*11] LI E., LÉVY B., ZHANG X., CHE W., DONG W., PAUL J.-C.: Meshless quadrangulation by global parameterization. Computers & Graphics (2011). 2

[LMH*15] LIU B., MASON G., HODGSON J., TONG Y., DESBRUN M.: Model-reduced variational fluid simulation. ACM Trans. Graph. 34, 6 (2015), 244:1–244:12. 2

[LTGD16] LIU B., TONG Y., GOES F. D., DESBRUN M.: Discrete connection and covariant derivative for vector field analysis and design. ACM Trans. Graph. 35, 3 (2016), 23:1–23:17. 2

[MRMH12] MEHTA S. U., RAMAMOORTHI R., MEYER M., HERY C.: Analytic tangent irradiance environment maps for anisotropic surfaces. Comp. Graph. Forum 31, 4 (2012). 2

[NBH18] NASIKUN A., BRANDT C., HILDEBRANDT K.: Fast approximation of Laplace–Beltrami eigenproblems. Comp. Graph. Forum 37, 5 (2018). 2, 9, 12

[PFH00] PRAUN E., FINKELSTEIN A., HOPPE H.: Lapped textures. In Proc. SIGGRAPH (2000), pp. 465–470. 2

[PLS*15] PAN H., LIU Y., SHEFFER A., VINING N., LI C.-J., WANG W.: Flow aligned surfacing of curve networks. ACM Trans. Graph. 34, 4 (2015). 2

[PM18] PAN Z., MANOCHA D.: Active animations of reduced deformable models with environment interactions. ACM Trans. Graph. 37, 3 (2018), 36:1–36:17. 2

[PP00] POLTHIER K., PREUSS E.: Variational approach to vector field decomposition. In Symposium on Data Visualization (2000), Springer, pp. 147–155. 3

[PP16] POELKE K., POLTHIER K.: Boundary-aware Hodge decompositions for piecewise constant vector fields. Computer-Aided Design 78 (2016), 126–136. 13

[PZ07] PALACIOS J., ZHANG E.: Rotational symmetry field design on surfaces. In ACM Trans. Graph. (2007), vol. 26, ACM, p. 55. 2, 3

[Qiu15] QIU Y.: SpectrA: C++ library for large scale eigenvalue problems, 2015. https://github.com/yixuan/spectra/. 6

[RGB*14] RAYMOND B., GUENNEBAUD G., BARLA P., PACANOWSKI R., GRANIER X.: Optimizing brdf orientations for the manipulation of anisotropic highlights. Comput. Graph. Forum 33, 2 (2014), 313–321. 2

[RLL*06] RAY N., LI W. C., LÉVY B., SHEFFER A., ALLIEZ P.: Periodic global parameterization. ACM Trans. Graph. 25, 4 (2006), 1460–1485. 2, 3

[RVAL09] RAY N., VALLET B., ALONSO L., LÉVY B.: Geometry-aware direction field processing. ACM Trans. Graph. 29, 1 (2009). 2

[RVLL08] RAY N., VALLET B., LI W. C., LÉVY B.: N-symmetry direction field design. ACM Trans. Graph. 27, 2 (2008). 2

[SBCBG11] SOLOMON J., BEN-CHEN M., BUTSCHER A., GUIBAS L.: Discovery of intrinsic primitives on triangle meshes. Comp. Graph. Forum 30, 2 (2011), 365–374. 2

[SFCBCV19] SAGEMAN-FURNAS A., CHERN A., BEN-CHEN M., VAXMAN A.: Chebyshev nets from commuting polyvector fields. ACM Trans. Graph. 38, 6 (2019). 2

[TLP06] TREUILLE A., LEWIS A., POPOVIC Z.: Model reduction for real-time fluids. ACM Trans. Graph. 25, 3 (2006), 826–834. 2

[TPP*11] TARINI M., PUPPO E., PANOZZO D., PIETRONI N., CIGNONI P.: Simple quad domains for field aligned mesh parametrization. Proc. SIGGRAPH Asia 2011 30, 6 (2011). 2

[Tur01] TURK G.: Texture synthesis on surfaces. In Proc. SIGGRPAH (2001), pp. 347–354. 2

[VCD*16] VAXMAN A., CAMPEN M., DIAMANTI O., PANOZZO D., BOMMES D., HILDEBRANDT K., BEN-CHEN M.: Directional field synthesis, design, and processing. Comp. Graph. Forum 35, 2 (2016), 545–572. 2

[vRESH16] VON RADZIEWSKY P., EISEMANN E., SEIDEL H.-P., HILDEBRANDT K.: Optimized subspaces for deformation-based shape editing and interpolation. Computers & Graphics 58 (2016), 128–138. 2

[vTSSH13] VON TYCOWICZ C., SCHULZ C., SEIDEL H.-P., HILDEBRANDT K.: An efficient construction of reduced deformable objects. ACM Trans. Graph. 32, 6 (2013), 213:1–213:10. 6

[vTSSH15] VON TYCOWICZ C., SCHULZ C., SEIDEL H.-P., HILDEBRANDT K.: Real-time nonlinear shape interpolation. ACM Trans. Graph. 34, 3 (2015), 34:1–34:10. 2

[War06] WARDETZKY M.: Discrete Differential Operators on Polyhedral Surfaces–Convergence and Approximation. PhD thesis, Freie Universität Berlin, 2006. 3

[WJBK15] WANG Y., JACOBSON A., BARBIČ J., KAVAN L.: Linear subspace design for real-time shape deformation. ACM Trans. Graph. 34, 4 (2015), 57:1–57:11. 2

[WL01] WEI L.-Y., LEVOY M.: Texture synthesis over arbitrary manifold surfaces. In SIGGRAPH (2001). 2

[WYT*06] WANG K., YANG W., TONG Y., DESBRUN M., SCHRÖDER P.: Edge subdivision schemes and the construction of smooth vector fields. ACM Trans. Graph. 25, 3 (2006), 1041–1048. 2

[YCLJ12] YAO C.-Y., CHI M.-T., LEE T.-Y., JU T.: Region-based line field design using harmonic functions. IEEE Transactions on Visualization and Computer Graphics 18, 6 (2012). 2

[YLX*15] YANG Y., LI D., XU W., TIAN Y., ZHENG C.: Expediting precomputation for reduced deformable simulation. ACM Trans. Graph. 34, 6 (2015), 243:1–243:13. 2, 6

[ZHT07] ZHANG E., HAYS J., TURK G.: Interactive tensor field design and visualization on surfaces. IEEE Transactions on Visualization and Computer Graphics 13, 1 (2007), 94–107. 2

[ZMT06] ZHANG E., MISCHAIKOW K., TURK G.: Vector field design on surfaces. ACM Trans. Graph. 25, 4 (2006), 1294–1326. 2

[ZZCJ14] ZHUANG Y., ZOU M., CARR N., JU T.: Anisotropic geodesics for live-wire mesh segmentation. Comput. Graph. Forum 33, 7 (2014). 2

[ZZM15] ZHANG W., ZHENG J., MAGNENAT-THALMANN N.: Real-time subspace integration for example-based elastic material. Comput. Graph. Forum 34, 2 (2015), 395–404. 2

A. Construction of the tensor field Laplacian

In this section, we provide details on the construction of the tensor field Laplace operator that is introduced in Section 3. We describe the construction at the example of second-order symmetric (1,1)-tensors. For other types of tensors, the Laplacians are constructed analogously. The tensor fields we consider are constant in every triangle of the surface mesh.

Transport of tensor We fix the coordinate system in every triangle by taking the first oriented normalized edge vector as the x-axis and the 90-degree rotated edge vector as the y-axis. Let A be the matrix representing a tensor in a triangle T_i. We want to transport the tensor to triangle T_j. In the case that the x-axes of the local coordinate systems in both triangles T_i and T_j are aligned with common edge e_{ij}, the transport is simply the identity. In general, this is not the case. Then, to transport A from T_i to T_j, we first transform the tensor in T_i to the e_{ij}-aligned coordinate system in T_i. Let R_i be the rotation matrix that maps the coordinates of vectors in the coordinate system in T_i to the coordinates of the vectors in the e_{ij}-aligned coordinate system. Then the transformation of A is $R_i A R_i^T$. The transport of $R_i A R_i^T$ to the e_{ij}-aligned coordinate system in T_j is the identity. Finally, we transform to the non e_{ij}-aligned coordinate system in T_j. Denoting the rotation matrix that transforms the coordinates in T_j to the e_{ij}-aligned coordinate system by R_j, the transported tensor in the coordinate system of T_j is $R_j^T R_i A R_i^T R_j$.

Mandel–Voigt notation When working with a linear operators on tensor fields, it is convenient to represent the tensors as vectors instead of matrices. Any matrix representing a symmetric $(1,1)$-tensor is a linear combination of the three matrices

$$\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \quad \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}, \quad \text{and} \quad \begin{pmatrix} 0 & 1/\sqrt{2} \\ 1/\sqrt{2} & 0 \end{pmatrix}, \tag{14}$$

which are orthonormal with respect to the Frobenius norm. The vector a that stacks the coefficients of a symmetric matrix A with respect to the three matrices (14),

$$A = \begin{pmatrix} a_1 & a_2 \\ a_2 & a_3 \end{pmatrix} \mapsto a = \begin{pmatrix} a_1 \\ a_3 \\ \sqrt{2}a_2 \end{pmatrix}, \tag{15}$$

is called the Mandel–Voigt representation of the matrix A.

Transport in Mandel–Voigt notation To write the tensor field Laplacian in Mandel–Voigt representation, we need to describe the transport of tensors in this representation. Let $G = R_i^T R_j$, then the transport of A is given by $G^T A G$. We want to find the matrix $P \in \mathbb{R}^{3\times 3}$ such that for any tensor A with Mandel–Voigt representation a, Pa is the Mandel–Voigt representation of $G^T A G$.

Let $G = \begin{pmatrix} g_1 & g_2 \\ g_3 & g_4 \end{pmatrix}$, $A = \begin{pmatrix} a_1 & a_2 \\ a_2 & a_3 \end{pmatrix}$, $B = G^\intercal A G$, and b the Mandel–Voigt representation of B. Then,

$$\begin{aligned} B &= G^\intercal A G \\ &= \begin{pmatrix} g_1 & g_3 \\ g_2 & g_4 \end{pmatrix} \begin{pmatrix} a_1 & a_2 \\ a_2 & a_3 \end{pmatrix} \begin{pmatrix} g_1 & g_2 \\ g_3 & g_4 \end{pmatrix} \qquad (16) \\ &= \begin{pmatrix} {g_1}^2 a_1 + 2g_1 g_3 a_2 + {g_3}^2 a_3 & g_1 g_2 a_1 + (g_2 g_3 + g_1 g_4) a_2 + g_3 g_4 a_3 \\ g_1 g_2 a_1 + (g_2 g_3 + g_1 g_4) a_2 + g_3 g_4 a_3 & {g_2}^2 a_1 + 2 g_2 g_4 a_2 + {g_4}^2 a_3 \end{pmatrix}. \end{aligned}$$

In Mandel–Voigt notation

$$\begin{aligned} b &= \begin{pmatrix} {g_1}^2 a_1 + 2 g_1 g_3 a_2 + {g_3}^2 a_3 \\ {g_2}^2 a_1 + 2 g_2 g_4 a_2 + {g_4}^2 a_3 \\ \sqrt{2}(g_1 g_2 a_1 + (g_2 g_3 + g_1 g_4) a_2 + g_3 g_4 a_3) \end{pmatrix} \qquad (17) \\ &= \begin{pmatrix} {g_1}^2 & {g_3}^2 & \sqrt{2} g_1 g_3 \\ {g_2}^2 & {g_4}^2 & \sqrt{2} g_2 g_4 \\ \sqrt{2} g_1 g_2 & \sqrt{2} g_3 g_4 & g_1 g_4 + g_2 g_3 \end{pmatrix} \begin{pmatrix} a_1 \\ a_3 \\ \sqrt{2} a_2 \end{pmatrix} \qquad (18) \\ &= Pa. \qquad (19) \end{aligned}$$

Laplacian for tensor fields A benefit of using the Mandel–Voigt representation is that the transport of tensors is realized by matrix multiplication. This gives the tensor field Laplacian (2) a structure that is similar to the usual structure of discrete Laplace operators. Let $P_{ij} \in \mathbb{R}^{3\times 3}$ denote the matrix realizing the transport of tensors from triangle T_i to T_j. Then the tensor field Laplacian in Mandel–Voigt notation is

$$(\Delta a)_i = \frac{1}{m_i} \sum_{j \in N_i} w_{ij} (a_i - P_{ji} a_j). \tag{20}$$

A matrix representation of the Laplace operator can be obtained by collecting the w_{ij} and P_{ij} in a stiffness matrix S and the m_i in a diagonal mass matrix M.