A Compact Spectral Descriptor for Shape Deformations

Skylar Sible[1] and **Rodrigo Iza-Teran**[2] and **Jochen Garcke**[3] and **Nikola Aulig**[4] and **Patricia Wollstadt**[5]

Abstract. Modern product design in the engineering domain is increasingly driven by computational analysis including finite-element based simulation, computational optimization, and modern data analysis techniques such as machine learning. To apply these methods, suitable data representations for components under development as well as for related design criteria have to be found. While a component's geometry is typically represented by a polygon surface mesh, it is often not clear how to parametrize critical design properties in order to enable efficient computational analysis. In the present work, we propose a novel methodology to obtain a parameterization of a component's plastic deformation behavior under stress, which is an important design criterion in many application domains, for example, when optimizing the crash behavior in the automotive context. Existing parameterizations limit computational analysis to relatively simple deformations and typically require extensive input by an expert, making the design process time intensive and costly. Hence, we propose a way to derive a compact descriptor of deformation behavior that is based on spectral mesh processing and enables a low-dimensional representation of also complex deformations. We demonstrate the descriptor's ability to represent relevant deformation behavior by applying it in a nearest-neighbor search to identify similar simulation results in a filtering task. The proposed descriptor provides a novel approach to the parametrization of geometric deformation behavior and enables the use of state-of-the-art data analysis techniques such as machine learning to engineering tasks concerned with plastic deformation behavior.

1 INTRODUCTION

Modern engineering product design relies heavily on computer-aided engineering (CAE) methods such as finite-element based simulation or computational optimization. With increasing computational capabilities and shorter design cycles in many industries, CAE methods are applied more and more extensively. As a result, a tremendous amount of data has become available, especially due to many components being repeatedly re-designed or optimized. This data provides potential to apply state-of-the-art machine learning techniques [2, 7, 25] including deep learning approaches [8, 10, 23] in order to increase the efficiency and quality of the design process, but also to handle the amount of generated data itself [2, 20]. One obstacle are the unstructured and finely detailed mesh representations typically used for design parts resulting in high-dimensional data vectors. Hence, to enable the application of computational tools, designers have to find suitable representations and parameterizations for both, components under development as well as related design criteria [9, 20].

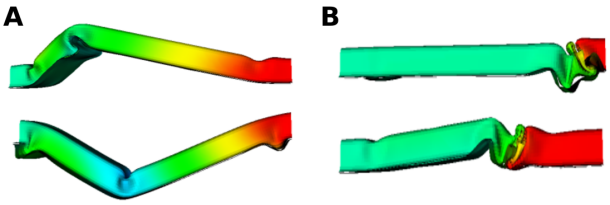

Figure 1. Modes of deformation typically encountered in automotive engineering [5]: (A) Upward and downward bending deformation mode, (B) axial crushing or folding deformation mode.

A critical design criterion in many application domains is the plastic deformation of a component under stress. For example, in automotive design, components are developed to exhibit a specific plastic deformation behavior during a crash in order to protect vehicle occupants [5]. A component's crash performance is improved if its plastic deformation avoids intrusion into the passenger cabin or allows for high energy absorption [5, 6, 14]. Hence, the ability to computationally analyze or optimize for specific deformation behaviors is a powerful design tool.

Current CAE-approaches typically study deformation behavior of a design part by representing the part as a polygon surface mesh and quantifying the displacement of a subset of selected nodes, for instance, during a crash simulation [6] (see also [16] for an example). Such a parameterization is computationally efficient because it requires to only monitor a small set of quantities of interest during the simulation—yet, it is limited in terms of the geometric complexity it is able to represent. For example, a bending of a part may be easily quantified by the displacement of a single node in a specified direction (Figure 1A), whereas more complex deformations, such as an axial crushing of the part, is not as easily quantified by the displacement of pre-selected mesh nodes (Figure 1B). Hence, the evaluation of deformation behavior, e.g., as a result of a crash simulation, requires an expert to manually check the resulting deformation and to alter the design if necessary. As a result, the development of a component requires many iterations of simulation and visual inspection (e.g., [21]), which makes the design process costly.

In this paper we present a novel representation for geometric deformations using a compact descriptor that allows for the represen-

[1] The Ohio State University, Columbus, Ohio USA, email: sible.3@buckeyemail.osu.edu
[2] Fraunhofer Center for ML and FhI. SCAI, Sankt Augustin, Germany, email: rodrigo.iza-teran@scai.fraunhofer.de
[3] Fraunhofer Center for ML and FhI. SCAI, Sankt Augustin, Germany, and Institut für Numerische Simulation, Universität Bonn, Germany email: jochen.garcke@scai.fraunhofer.de
[4] Honda Research Institute Europe GmbH, Offenbach/Main, Germany, email: nikola.aulig@honda-ri.de
[5] Honda Research Institute Europe GmbH, Offenbach/Main, Germany, email: patricia.wollstadt@honda-ri.de

tation of complex deformation modes using methods from spectral mesh processing (e.g., [19, 24]). Spectral mesh processing represents geometric shapes as a set of coefficients with respect to an eigenbasis obtained from spectral decomposition of a discrete mesh operator (see next section). We propose to represent deformations by identifying the subset of spectral coefficients, sufficient to describes a shape's plastic deformation with respect to its original form, and present a workflow to select these relevant spectral coefficients. In the following, we will first introduce spectral mesh processing and its application in the engineering domain, before we formally introduce our proposed method and demonstrate its application in a filtering task, relevant to engineering contexts. We conclude with a discussion of the results and directions for further research.

2 SPECTRAL MESH PROCESSING IN THE ENGINEERING DOMAIN

Spectral mesh processing is an approach to geometry processing that, similar to Fourier analysis in the 1D-domain, is based on the eigendecomposition of a suitable, discrete operator defined on the mesh (e.g., a discrete Laplace-Beltrami operator [24]). The decomposition returns eigenvectors that form an orthonormal basis into which any discrete function defined on the mesh can be projected, yielding a set of spectral coefficients. A common approach is to interpret the mesh vertices' Euclidean coordinates as mesh functions and project them into the eigenbasis to obtain a spectral representation of the mesh geometry. This transformation is invertible such that the geometry's spatial representation can be reconstructed from its spectral representation.

Representing mesh geometries in the spectral domain has interesting properties and allows to address various geometric processing tasks more easily. For example, obtained eigenvectors represent different geometric contributions to a geometry, where eigenvectors associated with smaller eigenvalues describe low-frequency functions on the mesh, while larger eigenvalues describe high-frequency functions [19] (where the ordering depends on the operator used). This property is used in spectral mesh compression [13], which efficiently stores geometries in terms of their first M low-frequency eigenvectors and corresponding spectral coefficients. This approach assumes that most geometries can be approximated sufficiently by their first M low-frequency, spectral components. From this reduced set of spectral components, the geometry's spatial representation can be reconstructed with acceptable loss. Similar spectral approximations have been used to realize shape matching and retrieval based on a subset of eigenvectors [17, 18]. Selecting a subset of spectral components for reconstruction using an orthogonal basis also relates to nonlinear approximation, see e.g. DeVore [4], and has also been used in image processing [15].

In the engineering domain, it has been shown that spectral mesh representations using the Laplace-Beltrami operator are an adequate choice for representing several deformations in a compact way [7, 11, 12]. Here, an approach has been proposed that uses the decomposition of one Laplace-Beltrami operator for a series of deformations, which are assumed to be isometric to a base shape. Spectral coefficients obtained by projecting several mesh deformations in x-, y- and z-directions into a common eigenbasis are used as a dimensionality reduction and visualization technique for large sets of deformed shapes resulting from crash simulations. It has also been proposed that certain eigenvectors can be interpreted as specific geometric operations on the shape, e.g., a translation in the Euclidean space [12]. Furthermore, the presented approach allows to handle arbitrary discrete functions defined on the mesh, so that additional functional properties of a part, such as plastic strain or stress, may be represented in the spectral domain.

Finally, note that in geometry processing the spectral representation of a shape is often used to find pose invariant representations, i.e., to distinguish a shape independent of pose changes by using the eigenfunctions as shape features. On the other hand, in the studied engineering applications part of the objective is to distinguish them in a pose dependent way, which is achieved by projecting the deformations as functions into the (joint) eigenbasis.

3 PROPOSED METHOD

Building on previous work in spectral mesh processing, the present paper proposes to solve the problem of efficiently representing geometric deformations by constructing a compact, spectral descriptor through the targeted and adaptive selection of spectral components. In particular, we propose to select the subset of only those components that are relevant for representing application-specific deformations in the spatial domain, and not just use the eigenvalue order as a fixed global criteria for the subset selection. Relevant components can be identified by the relative magnitude of their spectral coefficients, which indicates the importance of an eigenvector for representing the geometry in the spatial domain. In other words, opposed to previous applications such as mesh compression, we propose to select a subset of spectral components not based on their corresponding geometric frequency, but based on their contribution to the spatial representation irrespective of frequency, as indicated by the magnitude of their spectral coefficients.

In particular, we propose a workflow to find a compact spectral descriptor for deformations found in a set of geometries in three steps: a) compute a common spectral basis that can be used for all simulations, b) identify deformation modes in the set of simulations (e.g., through clustering in the spectral domain), and c) obtain spectral descriptors by identifying relevant spectral components for each deformation mode.

The obtained spectral descriptor is significantly smaller than the full mesh representation in the spatial domain, while preserving a high amount of relevant geometric information, yielding an efficient representation of geometric deformations for further computational analysis. Furthermore, we demonstrate empirically that the resulting descriptor provides an abstract representation of the deformation behavior by applying it in a nearest-neighbor search to identify similar simulation results in a filtering task.

3.1 Spectral mesh representation

In order to describe the generation of the descriptor, we first provide a formal definition of spectral mesh processing. Let $K = (G, P)$ be a triangle mesh embedded in $\mathbb{R}^3$ with a graph $G = (V, E)$ describing the connectivity of the mesh, where V are mesh vertices with $|V| = N$ and $E \subseteq V \times V$ the set of edges. The matrix $P \in \mathbb{R}^{N\times 3}$ describes the coordinates of mesh vertices in Euclidean space such that each vertex has coordinates $\mathbf{p}_i = (x_i, y_i, z_i)$. We view K as an approximation of the Riemannian manifold $\mathcal{M}$, isometrically embedded into $\mathbb{R}^3$. Furthermore, let $f : \mathcal{M} \to \mathbb{R}$ be a continuous function on $\mathcal{M}$. Evaluating this function at vertices V yields the discrete mesh function $f_K : K \to \mathbb{R}$.

We may now define a discrete, linear operator on $\mathcal{M}$ (see [24] for a discussion of possible operators). We here consider the Laplace-Beltrami operator, which is defined as the divergence of the gradient

in the intrinsic geometry of the shape and is a generalization of the Laplace operator to Riemannian manifolds. The operator is invariant under isometric transformations, i.e., transformations that preserve geodesic distances on the shape. In the following, we thus assume that we operate on sets of shapes that are isometries of each other in order to assume constant eigenbases between deformed shapes. In practice, numerically ϵ-isometries will be present, which will result in only approximately the same eigenbasis. Spectral mesh decomposition still can be used accordingly since under suitable conditions the Laplacians only differ by a scaling factor in such a case [12].

The eigendecomposition of the operator returns eigenvectors $E = [\psi_1, \psi_2, \ldots, \psi_N]$, ordered by the magnitude of the corresponding eigenvalues, $\lambda_1 < \lambda_2 < \ldots < \lambda_N$, where each eigenvector corresponds to a frequency component of the mesh function in increasing order [19]. Note that opposed to classical Fourier transform where basis functions are fixed, the orthonormal basis obtained from the spectral decomposition of a mesh operator depends on the mesh geometry and operator used. Often, only the set of the first M eigenvectors ordered by the magnitude of eigenvalues, E_M, is used for further processing such that high-frequency components are discarded.

The normalized eigenvectors E of a symmetric operator form an orthonormal basis. We therefore may map any discrete mesh function f_K, given by a vector $\mathbf{f}$, into this basis to obtain a representation, or encoding, in the spectral domain,

$$\hat{\mathbf{f}} = E^\top \mathbf{f}, \tag{1}$$

where the columns of E are eigenvectors $\{\psi_i\}_{i=1}^N$ and $\hat{\mathbf{f}}$ contains corresponding spectral coefficients $\{\alpha_i\}_{i=1}^N$. The spectral coefficients are thus obtained by calculating $\{\alpha_i\}_{i=1}^N = \langle \mathbf{f}, \psi_i \rangle$. The inverse transform reconstructs, or decodes, the mesh function in the spatial domain,

$$\mathbf{f} = E\,\hat{\mathbf{f}}. \tag{2}$$

By considering Euclidean coordinates, $P = [\mathbf{f}_x, \mathbf{f}_y, \mathbf{f}_z]$, as mesh functions we project the mesh geometry into the spectral domain,

$$\hat{P} = E^\top P, \tag{3}$$

such that each row of $\hat{P}$, $\hat{\mathbf{p}}_i = [\alpha_i^x, \alpha_i^y, \alpha_i^z], i = 1, \ldots, N$, contains spectral coefficients that can be used to express x-, y-, and z-coordinates as

$$\mathbf{f}_x = \sum_{i=1}^{N} \alpha_i^x \psi_i,\ \mathbf{f}_y = \sum_{i=1}^{N} \alpha_i^y \psi_i,\ \mathbf{f}_z = \sum_{i=1}^{N} \alpha_i^z \psi_i. \tag{4}$$

An approximation of the mesh geometry is obtained by using the coefficients corresponding to the first $M \ll N$ eigenfunctions, or suitably selected ones.

In a CAE-application, the spectral representation may now be used to project a set of geometries, each represented as three-valued mesh functions, into the same spectral basis, allowing for a joint handling of the data in a common space. This approach assumes that geometries are available in a regular mesh format and that the mesh is the same, or is suitably interpolated, for all geometries in the set. A single Laplace-Beltrami operator is then computed for the set of deformations [7, 12]. Observe that the operator is computed using geodesic distances along the surface of a shape, where one assumes that the deformations do not modify this distance. As a result, the approach yields a common representation for deformations using the spectral coefficients obtained by projecting three functions, each for mesh deformations in x-, y- and z-directions, to the eigenvectors of only one shape.

We would like to explain some of the properties of the data representation by comparing to principal component analysis (PCA). In that data-driven approach, a data matrix (e.g., comprising of the deformations as vectors) gets compressed into a small number components based on the variance of the data, where the largest variance will be contained in the first principal component. As an example let us consider a series of deformations that puts two areas of a part nearby. Here, the PCA will provide principal components that concentrate on those sections of the part and will reproduce behavior similar to this one. Now, let us assume one did not include data for a deformation that affects only one area of the part. The principal components will not be able to suitably represent such an unseen and strongly different deformation behavior since it was not trained for that.

On the other hand, take the basis obtained from the Laplace-Beltraim operator, where only the shape geometry is taken into account. In this basis, the (new) deformation of the shape can be reproduced in the same fashion as the earlier ones. In other words, whereas in the PCA higher variance, which can be interpreted as higher frequencies of the input data, is discarded, in the Laplace-Beltrami basis higher (geometric) frequencies of the underlying shape are discarded. Note here also, it was shown using suitable assumptions that for functions bounded in the H^1-Sobolev norm the L_2-approximation using the orthonormal basis obtained from the Laplace operator is optimal in a certain best basis sense [1]; this result can be extended to the Laplace-Beltrami operator and functions in the Sobolev space $H^{2,2}$ on the underlying manifold [22].

3.2 Finding an efficient spectral descriptor for plastic deformations

In the spectral domain, individual eigenvectors, ψ_i, can be interpreted as geometric contributions to the spatial representation of the shape (geometric frequencies), where in general eigenvectors with low eigenvalues represent more low-frequency contributions and those with high eigenvalues high-frequency contributions (depending on the operator used). Furthermore, eigenvectors may be associated with specific geometric transformations of the shape, such that changes in the corresponding spectral coefficients can even have a mathematical (e.g., rotation of a shape in the underlying space) or physical interpretation (e.g., deformation in parts of the shape).

When projecting Euclidean coordinates, $P = [\mathbf{f}_x, \mathbf{f}_y, \mathbf{f}_z]$, into the eigenbasis, the magnitude of resulting spectral coefficients associated with each eigenvector, $\{\alpha_i^x, \alpha_i^y, \alpha_i^z\}_{i=1}^N$, represents the relevance of that eigenvector's geometric contribution to the deformed shape's spatial representation [12]. Based on this property, we propose to find a compact spectral descriptor, $\mathbf{S}$, for a deformation, by selecting only those coefficients, $\alpha_j^{(\cdot)}$, that have a high relative magnitude compared to a suitable baseline. This is a further filtering in comparison to [12] or mesh compression, where a fixed number of the spectral components is used, based on the order of the corresponding eigenvalues. We assume that coefficients with high values indicate that corresponding eigenvectors, ψ_j, represent geometric information relevant for the description of the deformation in the spatial domain. Figure 2 shows an exemplary workflow for finding $\mathbf{S}$, which is described in detail in the following.

To find $\mathbf{S}$, we first have to provide a mesh representing the geometry of a desired deformation behavior. For example, in the automotive context we may wish to describe an axial folding of a beam during a frontal crash, because this deformation mode leads to high energy absorption. Such a geometry may be selected from a set of k exist-

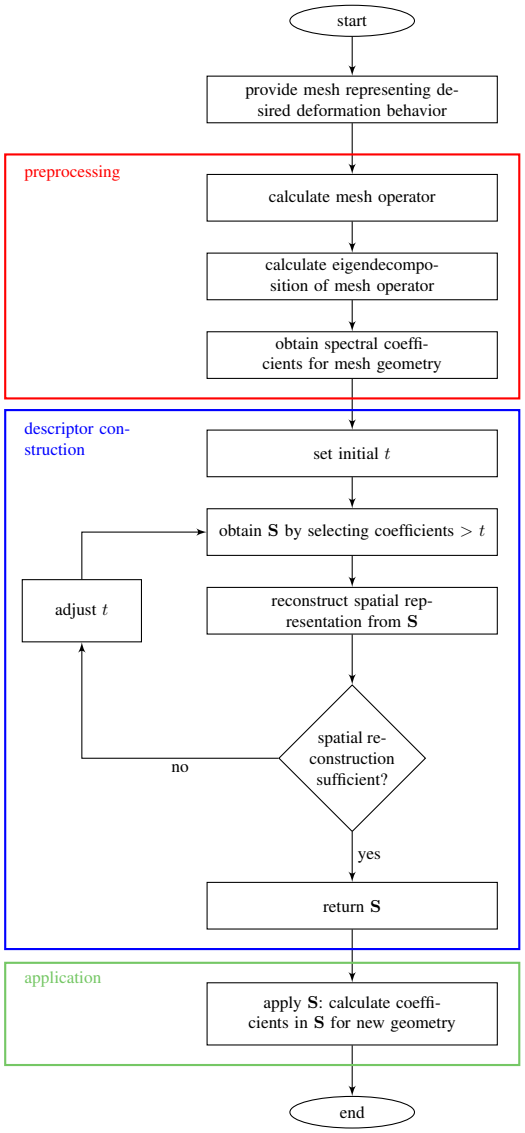

Figure 2. Exemplary workflow for finding and applying the proposed spectral shape descriptor: preprocessing (red box); construction of $\mathbf{S}$ by selection of spectral coefficients (blue box); application of selected descriptor (green box). See main text for detailed description.

ing, different simulations. The desired deformation may be identified either through manual selection by an expert motivated by functional requirements, or through data-driven approaches in either the spatial or spectral domain. For example, clustering of spectral coefficients may be used to reveal main modes of deformations present in the set of simulations (Figure 3), e.g., bending or folding behavior (Figure 1). We may either cluster the geometries based on their deformation in the final time step of a simulation run, the transient data from the full simulation run over time, or some other physical quantities such as plastic strain or stress on the surface of the geometry. In particular, a suitable visual representation of the coarse behavior of the deformation can often be obtained by using the x-, y-, z-spectral coefficients of the first eigenvector in eq. (4) [7, 12].

Note that the application of the descriptor requires that the deformation, or other quantities of interest, is sufficiently represented by the initial surface mesh. In particular, the mesh resolution has to be fine enough to represent the deformation. For application domains concerned with more high-frequency deformations, this results in fine meshes with a high number of nodes. Here, the applicability of the method may be limited by the practical run time of the proposed method, which depends on the number of nodes in the mesh (see also the asymptotic run time of the proposed approach, discussed in section *Computational complexity of the proposed workflow*).

Once a mesh representing the target deformation is identified, we calculate its spectral representation according to eq. (3) (Figure 2, red box) to obtain coefficients $\hat{\mathbf{p}}_i$. Now, several approaches to defining $\mathbf{S}$ by selecting relevant coefficients are conceivable. In many applications, e.g., computer graphics, typically the first M eigenfunctions are used. In contrast to this approach, we propose to identify relevant eigenfunctions based on the magnitude of their *coefficients*. For a single shape exhibiting the targeted deformation, we may thus select all coefficients exceeding a threshold t,

$$\mathbf{S} = \left\{ \alpha_j^x, \alpha_j^y, \alpha_j^z | \alpha_j^x > t \vee \alpha_j^y > t \vee \alpha_j^z > t \right\}, \tag{5}$$

where t may be set based on a statistical criterion, e.g., relative to the mean of coefficients, to identify those coefficients that indicate a high relevance for representing the shape.

Alternatively, the shape describing the desired deformation may be contrasted against the undeformed baseline shape, such that relevant coefficients can be identified by the largest differences in coefficients between baseline and deformed shape (Figure 4B,C). Here again the setting of a threshold t for selecting the highest differences is required.

Both procedures identify components that contribute to the spatial representation of the targeted deformation while ignoring less relevant geometric components. Hence, we obtain a sparse description of the geometric information relevant to characterize a specific deformation in the spatial domain.

For both approaches, t may be either set based on a statistical criterion, but may also be found through an iterative process that alternates between lowering or increasing t and evaluating the reconstruction quality in the spatial domain (Figure 2, blue box). Reconstruction quality may either be judged through visual inspection by an expert or through calculation of an error metric between the original and reconstructed shape. Adjusting t controls for the size of the descriptor, $|\mathbf{S}| = M$, and thus allows for a trade-off between compactness and the level of geometric detail captured by the descriptor. Note that recovering the original mesh representation requires *all* spectral coefficients (according to eq. 2), hence—even though $\mathbf{S}$ is sufficient to represent the geometric deformation of a part—it is not possible to recover the original mesh representation from $\mathbf{S}$ alone. Nevertheless, an approximate reconstruction is still possible.

In our experiment, we demonstrate that, for a part typically encountered in engineering contexts, setting t based on a statistical criterion resulted in a small descriptor size, $M \ll N$, where N is the number of nodes in a triangular surface mesh. This descriptor was able to robustly filter simulation results based on their deformation mode despite the considerable reduction in description length, indicating that a high amount of application-relevant geometric detail was retained.

3.3 Application scenarios

Once the descriptor $\mathbf{S}$ is constructed, it can be applied in further computational tasks (Figure 2, green box). For example, the descriptor may be used as a feature in a machine learning task, encoding the plastic deformation behavior of a part. A possible application is

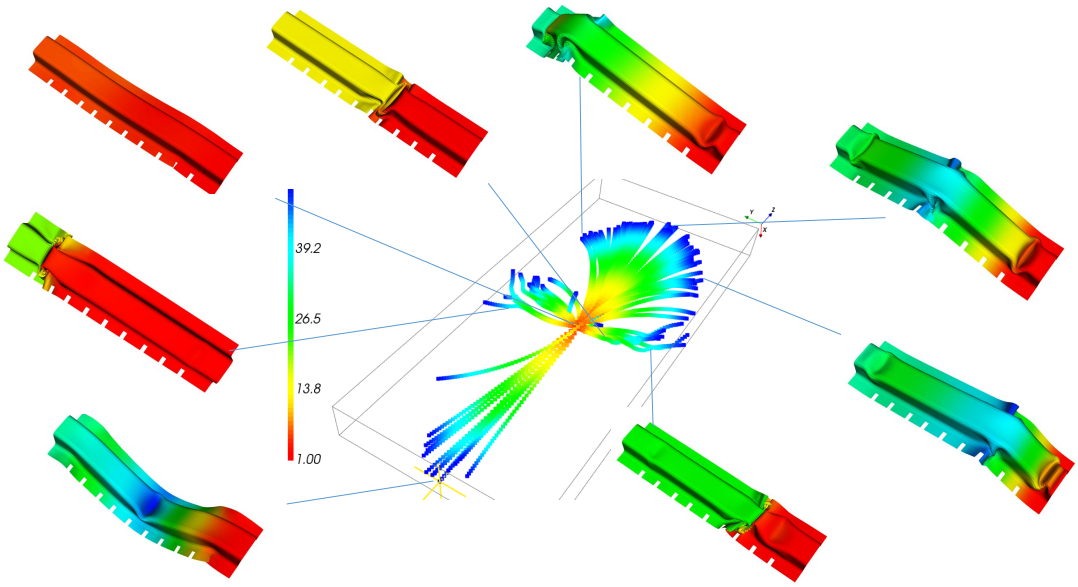

Figure 3. Visualization of the spectral coefficients for the first eigenvector of the x-, y-, z-coordinates for all simulation runs and all time steps (color coding indicates simulation time steps, where red denotes the first time step). Note how the spectral representation allows for an identification of the different deformation modes based on the distance of simulation runs in the spectral domain. Coefficients in the bottom left correspond to a downward bend, top right corresponds to upward bend, and left and right correspond to axial crushing along the length of the part. Color coding of parts indicates the difference to the undeformed baseline geometry, where red indicates smaller and blue larger distances.

meta-modelling, which aims at replacing costly simulations or optimization runs by a cheaper evaluation of statistical models. The descriptor can be used in case the design process investigates the relationship between a geometry and an objective function of interest. Learning such a meta-model requires the efficient parametrizations of the properties of interest, such as deformation behavior. Further tasks may include optimization of the part with respect to some property, e.g., material thickness, while the shape descriptor is used as a constraint or part of the objective function to ensure the desired deformation behavior. Here, the descriptor can be used to describe the targeted deformation behavior and to describe the deformation behavior of parts in intermediate steps of the optimization. As soon as the desired and actual deformation behavior diverge, measurable by an increasing distance between the descriptors, the optimization can be stopped automatically. Alternatively, the distance may be used as part of the objective function, e.g., when using evolutionary optimization techniques.

A further task, common in the engineering design process, is the filtering and verification of simulation results based on geometric properties. Often, large numbers of simulation runs are performed in order to investigate the impact of variations in design parameters on the component under development. This results in large amounts of simulation data that typically require time-intensive visual inspection by an expert in order to verify the success of individual simulations [25, 2]. Our descriptor can be used both, to automatically verify the outcome of a simulation with respect to the desired deformation behavior, and to filter simulation results based on the deformation behavior (see section *Experiments*). In particular, the proposed workflow may again be used to find the spectral descriptor for the targeted deformation behavior and the actual deformation behavior of the part being optimized. To verify a result, a threshold on the acceptable distance between both descriptors can be set, allowing for the verification of large sets of simulation results without the need for manual inspection by an expert.

Note that since the Laplace-Beltrami operator is only invariant under isometric transformations, the presented approach is limited to scenarios where all shapes involved in finding and using the descriptor represent the same baseline geometry in different states of (isometric) deformation. This is however the case for many engineering applications, where properties of the structure are determined by variables that can be varied independently of the mesh geometry. Two examples are thickness of shell finite elements or material properties of finite elements such as failure criteria. Another application scenario is the variation of loading or boundary conditions whose variation can lead to different deformation behaviors while the base geometry of the component of interest stays the same. In these scenarios, our method has relevance for robustness or optimization studies, where certain deformation modes are desired as a main objective. Additionally, small changes to the geometry can be handled by interpolating the mesh to a joint reference mesh.

3.4 Computational complexity of the proposed workflow

In terms of computational efficiency, the most costly operations in constructing the descriptor are the preprocessing phase (Figure 2, red box) comprising the calculation of the mesh operator, its eigendecomposition, and the projection of Euclidean coordinates into the eigenbases, eq. (3). The asymptotic time complexity of the preprocessing is dominated by the matrix multiplications performed as part of the operator definition and its eigendecomposition, which is cubic in the number of mesh vertices, $\mathcal{O}(N^3)$. This operation is done only once and can be performed offline with respect to the application of the descriptor in a subsequent task (Figure 2, green box). The

time complexity is cubic if a naive algorithm is used, but also algorithms with subcubic runtimes are available (e.g., [3]). Furthermore, since the discrete Laplace-Beltrami matrix is inherently data sparse, fast computations of the first, say, 1000 eigenvectors using numerical approaches exploiting this data sparseness seem possible, which is part of future work. In any case, this one-time pre-processing step is small in comparison to the runtime of a single numerical simulation performed to generate one datum.

The application of the descriptor to new geometries in the online phase (Figure 2, green box) requires the projection of the new geometry's Euclidean coordinates, P, into the bases indicated by the descriptor, $E_{\mathbf{S}}$. The projection consists of a matrix multiplication, $E_{\mathbf{S}}^T P$, eq. (3), which has asymptotic time complexity $\mathcal{O}(3MN)$, with $M \ll N$.

4 EXPERIMENTS

4.1 Data generation

To demonstrate the effectiveness of our method, we used the proposed descriptor to represent the deformation behavior of a hat section beam in a crash simulation and filter simulation results based on deformation mode.

A hat section beam is a structural element common in engineering domains such as civil or automotive engineering (Figure 4A). It consists of a top-section with a hat-shaped cross-section that is joined together with a base plate.

We simulated an axial crush of the beam using LS-DYNA mpp R7.1.1 (Figure 4B,C), double precision, while inducing various deformation modes. On the resulting data, we used the proposed descriptor to filter results based on a desired deformation behavior.

Variation in the deformation behavior was introduced by adding "notches" in the flange of the hat section at defined locations along the length of the part (Figure 4A) and varying the material thickness of the part between 1 mm and 10 mm. A notch was simulated by setting the material thickness to 0.0001 mm in order to simulate a removal of the material at that point. The setup resulted in a simulation bundle consisting of $k = 100$ simulations with a large variety of deformation modes, each with several saved time steps.

4.2 Calculation of mesh spectral coefficients and selection of shape descriptor

For further analysis, we considered the set of deformed shapes from the final time step of all simulation runs. The spectral representation according to eq. (3) was computed, where we used the first 500 eigenvectors with smallest eigenvalues. We selected geometries representative of one of three deformation modes, namely *upward bend*, *downward bend*, and *axial crush* (Figure 1A, B), identified through clustering of the first 500 coefficients in the spectral domain. Alternatively, Figure 3 shows the spectral coefficients of x-, y-, z-coordinates corresponding to the first eigenvector for all simulation runs and time steps, from this visualization the different modes could also be selected.

For each deformation mode we obtained the spectral descriptor according to eq. (5) by setting t to one standard deviation above the mean over all coefficients for a representative shape. This approach resulted in descriptor sizes of $M = 14$ for axial crush, $M = 17$ for upward bend, $M = 16$ for downward bend, compared to an original mesh size, N, of around 9000 nodes. The quality of the descriptor was validated through visual inspection of the spatial reconstruction (Figure 5A). Note that to allow for proper reconstruction, for each

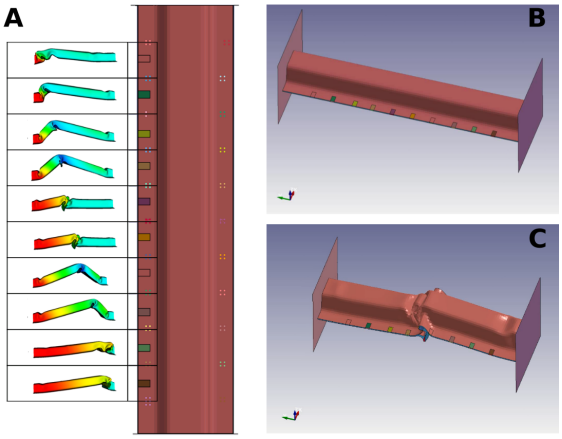

Figure 4. (A) Hat section beam used for data generation. Colored markers indicate notches introduced at various locations along the length of the shape, which lead to various different deformation behaviors shown to the left (notches denote areas of material thickness of 0.0001 mm). (B) Simulation setup with undeformed baseline shape. (C) Exemplary final time step of simulation run showing deformed shape.

selected coefficient, $\alpha_j^{(\cdot)} \in \mathbf{S}$, also the spectral coefficients corresponding to the remaining two 3D coordinates were added if not already contained in the descriptor. Furthermore, eigenvectors ψ_1 and ψ_2 were added to the reconstruction.

We compared the spatial reconstruction of the meshes from our descriptor of length M to the reconstruction from the first M low-frequency components (eigenvectors with smallest eigenvalues), i.e. $N = M$ in eq. (4) (Figure 5A, bottom row). The latter is a common approach in dimensionality reduction or compression applications [13, 7]. The comparison shows that our descriptor obtained a better reconstruction quality than the reconstruction from low-frequency components alone. In particular, the high-frequency components included in the descriptor captured also finer detail on the mesh, for example, the edge of the hat section in the center of the bent parts. An approximate decoding from the sparse encoding of size M is therefore possible, where the quality is good enough to reconstruct the main behavior, but not the details, e.g., the location of the axial crush is not preserved.

4.3 Application of spectral descriptor for filtering and shape retrieval

We used the spectral descriptors to filter the set of simulation results for geometries exhibiting a specific crash behavior. As described in section *Application scenario*, filtering large-scale simulation runs based on geometric properties is a common application scenario in the engineering design process.

To filter simulation runs using the proposed descriptor, $\mathbf{S}$, we first obtained spectral coefficients for all shapes through projection into the spectral domain; we then calculated the similarity between these coefficients and coefficients in $\mathbf{S}$ using the cosine similarity. Figure 5B shows the similarity between the descriptor of the axial crush deformation mode and all simulation results. Our approach correctly identified all simulation results exhibiting an axial crushing as most similar to the spectral descriptor, with the exception of one geometry showing an upward bend. Figure 5C shows the nine simulation results most similar to the descriptor, which all show an axial crushing of the beam. Note that the approach was able to identify simulation

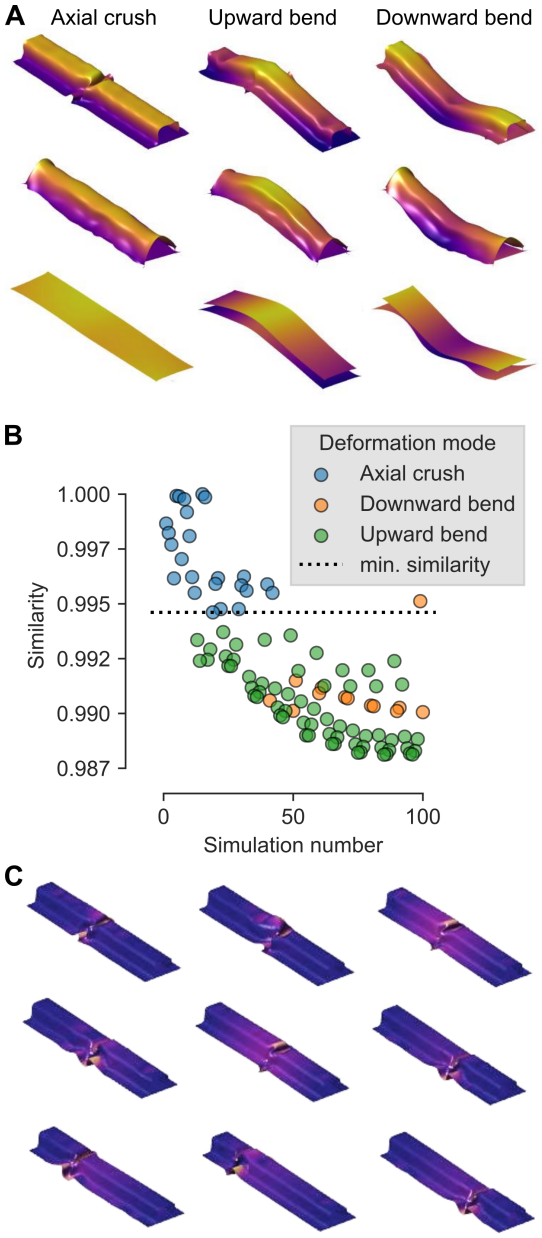

Figure 5. (A) Mesh reconstruction for three deformation modes (color scale indicates displacement in z-direction): top row shows mesh reconstructions from the first 500 eigenvectors ordered by magnitude of eigenvalues; middle row shows reconstruction from the proposed descriptor with sizes $M = 14$ for axial crush, $M = 17$ for upward bend, $M = 16$ for downward bend, bottom row shows reconstruction from the first $M = 14$, $M = 17$, and $M = 16$ eigenvectors ordered by magnitude of their eigenvalues. (B) Cosine similarity between axial crush descriptor and all simulation runs. (C) First nine most similar simulation results for axial crush spectral descriptor (color scale indicates displacement in z-direction).

results exhibiting an axial crushing irrespective of the exact location of the axial folding along the part.

In summary, the descriptor provided an abstract description of the deformation behavior that did not require the specification of an exact deformation, e.g., in terms of the displacement of individual nodes. The descriptor successfully represented application-relevant geometric information through the targeted selection of coefficients while being of much smaller size than the full mesh representation in the spatial domain.

5 CONCLUSION AND FUTURE WORK

We proposed a novel approach for the efficient representation of geometric deformations using a spectral descriptor comprising components selected in a targeted and adaptive fashion. The selection procedure ensures that only geometric components relevant for specifying the targeted deformation are included, which makes the descriptor much smaller in size than a full geometric representation of the deformation, e.g., by a surface mesh, and also smaller and more focused than using the first few hundred spectral components as in [12]. Despite its compactness, the descriptor was able to capture a high amount of application-relevant geometric information and provided the necessary descriptive power to distinguish between deformation modes in a filtering task. The trade-off between size and represented geometric detail makes the descriptor a promising tool for the parametrization of also complex geometric deformations in various computational tasks.

Future work may explore the applicability of the descriptor, for example, in tasks such as structural optimization with plastic deformation as design criterion. Here, the proposed descriptor offers a powerful design tool and may improve on current approaches that require extensive manual intervention and evaluation by an expert. Furthermore, the descriptor may be used in post-processing of simulation data by identifying or filtering results based on geometric deformation. Especially in large-scale data sets, such an approach allows to automatically verify the success of a simulation instead of requiring the visual inspection of large sets of simulation results by an expert. Lastly, the descriptor may be used as a feature in machine learning tasks on large-scale engineering data sets, representing a central design property in many application domains.

ACKNOWLEDGEMENTS

The authors thank Emily Nutwell of the OSU SIMCenter for support in generating the hat section model.

REFERENCES

[1] Haïm Brezis and David Gómez-Castro, 'Rigidity of optimal bases for signal spaces', *Comptes Rendus Mathematique*, **355**(7), 780–785, (2017).
[2] Steven Burrows, Benno Stein, Jörg Frochte, David Wiesner, and Katja Müller, 'Simulation data mining for supporting bridge design', *9th Australasian Data Mining Conference (AusDM 2011)*, **121**, 163–170, (2011).
[3] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, *Introduction to Algorithms*, The MIT Press, Cambridge, MA, 3 edn., 2009.
[4] Ronald A. DeVore, 'Nonlinear approximation', *Acta Numerica*, **7**, 51–150, (1998).
[5] Paul Du Bois, Clifford C. Chou, Bahig B. Fileta, Tawfik B. Khalil, Albert I King, Hikmat F. Mahmood, Harold J. Mertz, Jac Wismans, Priya Prasad, and Jamel E. Belwafa, 'Vehicle crashworthiness and occupant protection', *American Iron and Steel Institute*, (2004).
[6] Jianguang Fang, Guangyong Sun, Na Qiu, Nam H. Kim, and Qing Li, 'On design optimization for structural crashworthiness and its state of the art', *Structural and Multidisciplinary Optimization*, **3**(55), 1091–1119, (2017).
[7] Jochen Garcke and Rodrigo Iza-Teran, 'Machine learning approaches for data from car crashes and numerical car crash simulations', in *Proceedings of the NAFEMS World Congress 2017*, (2017).

[8] Theodoros Georgiou, Sebastian Schmitt, Markus Olhofer, Yu Liu, Thomas Bäck, and Michael Lew, 'Learning fluid flows', in *2018 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, (2018).

[9] Lars Graening and Bernhard Sendhoff, 'Shape mining: A holistic data mining approach for engineering design', *Advanced Engineering Informatics*, **28**(2), 166–185, (2014).

[10] Tinghao Guo, Danny J. Lohan, Ruijin Cang, Max Yi Ren, and James T. Allison, 'An indirect design representation for topology optimization using variational autoencoder and style transfer', *2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 804, (2018).

[11] Rodrigo Iza-Teran, *Geometrical Methods for the Analysis of Simulation Bundels*, Ph.D. dissertation, University of Bonn, Bonn, Germany, 2016.

[12] Rodrigo Iza-Teran and Jochen Garcke, 'A geometrical method for low-dimensional representations of simulations', *SIAM/ASA Journal on Uncertainty Quantification*, **7**(2), 472–496, (2019).

[13] Zachi Karni and Craig Gotsman, 'Spectral compression of mesh geometry', *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, 279–286, (2000).

[14] Shu Tian Liu, Ze Qi Tong, Zhi Liang Tang, and Zong Hua Zhang, 'Design optimization of the S-frame to improve crashworthiness', *Acta Mechanica Sinica*, **30**(4), 589–599, (2014).

[15] Gabriel Peyré, 'Manifold models for signals and images', *Computer Vision and Image Understanding*, **113**(2), 249–260, (2009).

[16] Marcus Redhe and Larsgunnar Nilsson, 'Optimization of the new Saab 9-3 exposed to impact load using a space mapping technique', *Structural and Multidisciplinary Optimization*, **27**(5), 411–420, (2004).

[17] Martin Reuter, Franz-Erich Wolter, and Niklas Peinecke, 'Laplace-spectra as fingerprints for shape matching', *Proceedings of the 2005 ACM Symposium on Solid and Physical Modeling*, **1**(212), 101–106, (2005).

[18] Martin Reuter, Franz-Erich Wolter, and Niklas Peinecke, 'Laplace-beltrami spectra as shape-dna of surfaces and solids', *Computer-Aided Design*, **38**, 342–366, (2006).

[19] Olga Sorkine, 'Laplacian Mesh Processing', in *Eurographics (STARs)*, pp. 53–70, (2005).

[20] Tobias C. Spruegel, Richard Rothfelder, Sebastian Bickel, Andreas Grauf, Christopher Sauer, Benjamin Schleich, and Sandro Wartzack, 'Methodology for plausibility checking of structural mechanics simulations using Deep Learning on existing simulation data', *Proceedings of NordDesign: Design in the Era of Digitalization, NordDesign 2018*, (2018).

[21] Tao Tang, Weigang Zhang, Hanfeng Yin, and Han Wang, 'Crushing analysis of thin-walled beams with various section geometries under lateral impact', *Thin-Walled Structures*, **102**, 43–57, (2016).

[22] Tobias Tesch, *Invariant operator approach for the analysis of simulation bundles*, Master thesis, Institut für Numerische Simulation, Universität Bonn, Bonn, Germany, 2018.

[23] Nobuyuki Umetani, 'Exploring generative 3D shapes using autoencoder networks', in *SIGGRAPH Asia Technical Briefs 2017*, (2017).

[24] Hao Zhang, Oliver van Kaick, and Ramsay Dyer, 'Spectral mesh processing', *Computer Graphics Forum*, **29**(6), 1865–1894, (2010).

[25] Zhijie Zhao, Xianlong Jin, Yuan Cao, and Jianwei Wang, 'Data mining application on crash simulation data of occupant restraint system', *Expert Systems with Applications*, **37**(8), 5788–5794, (2010).