EUROGRAPHICS 2010 / M. Chen and O. Deussen
(Guest Editors)

Volume 30 (2011), Number 2

Shape Analysis with Subspace Symmetries

Alexander Berner[1] Michael Wand[1,2] Niloy J. Mitra[3] Daniel Mewes[1] Hans-Peter Seidel[1]

[1]MPI Informatik [2]Saarland University [3]KAUST / IIT Delhi

Abstract

We address the problem of partial symmetry detection, i.e., the identification of building blocks a complex shape is composed of. Previous techniques identify parts that relate to each other by simple rigid mappings, similarity transforms, or, more recently, intrinsic isometries. Our approach generalizes the notion of partial symmetries to more general deformations. We introduce subspace symmetries whereby we characterize similarity by requiring the set of symmetric parts to form a low dimensional shape space. We present an algorithm to discover subspace symmetries based on detecting linearly correlated correspondences among graphs of invariant features. The detected subspace symmetries along with the modeled variations are useful for a variety of applications including shape completion, non-local and non-rigid denoising. We evaluate our technique on various data sets. We show that for models with pronounced surface features, subspace symmetries can be found fully automatically. For complicated cases, a small amount of user input is used to resolve ambiguities. Our technique computes dense correspondences that can subsequently be used in various applications, such as model repair and denoising.

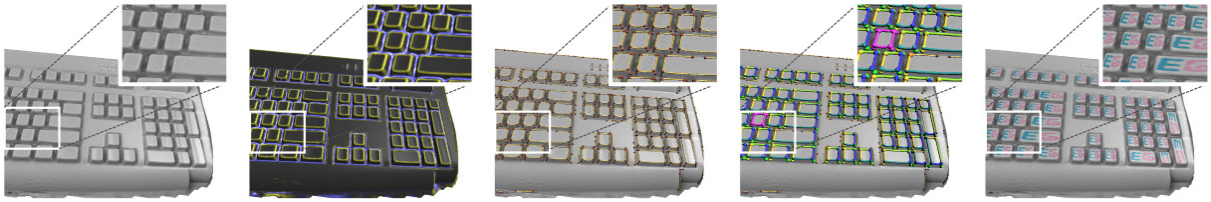
1. Introduction

Self-similarity and repetitions are ubiquitous in man-made and natural objects. Such structural regularities often relate to form, function, aesthetics, and design considerations. Discovering structural redundancies along with their dominant variation modes from 3D geometry not only allows us to better *understand* the underlying objects, but is also beneficial for several geometry processing tasks including compact representation, symmetrization, shape completion, intuitive shape manipulation.

Given an object $\mathcal{S}$, our goal is to look for a piece of geometry $\mathcal{U} \subseteq \mathcal{S}$ and a collection of associated mapping functions $\mathbf{f}^{(i)} : \mathcal{U} \to \mathbb{R}^3$ that respectively create instantiations $\mathbf{f}^{(i)}(\mathcal{U})$, thereby matching the original geometry $\mathcal{S}$ up to an allowable error tolerance. Most previous efforts restrict the mapping functions to families of reflections, rigid mappings, and similarity transforms [MGP06, PSG*06, MGP07, PMW*08]. Many real-world objects such as different windows of a building or ornamental structures in man-made sculptures, however, exhibit structural redundancies that cannot be captured by such constrained mappings (see Figure 7).

A central challenge in generalizing the notion of symmetry is to decide on the allowable space of admissible transformations $\mathbf{f}^{(i)}$: While too much flexibility using a large number of parameters lead to overfitting and spurious matches, an overly restrictive mapping fails to compactly capture redundancy present in the input. Our key idea is to not look at a prescribed set of admissible transformations independent of the input but first look for more general mappings supported by the input data. Then, we learn the space of useful variations by analyzing the ensemble of detected mapping functions, thus building a model for capturing *variations* exhibited by the input geometry.

We formalize the above observation by introducing the notion of *subspace symmetries*. We constrain the deformation functions $\mathbf{f}^{(i)}$ to lie within a low dimensional affine subspace of all possible mappings. Thus, although each instance is described by a small amount of data, i.e., the low dimensional coordinate vector within the subspace, we still allow large variations for the individual mapping functions. In spirit of classical principal component analysis (PCA), the existence of a low dimensional structure within a high dimensional shape space serves as a validation criterion to characterize corresponding geometry. Our problem setting, however, is different from traditional PCA: We have to search for the useful correspondences, which are unknown, in the raw input to help reveal such a subspace structure. This is a challenging search problem. We propose an algorithm to find subspace symmetries that acts in three steps (see Figure



(a) input model (3D scan) (b) max. principal curvaure (blue neg., yellow pos.) (c) feature graph (d) graph matches (corr. color coded, ref. pink) (e) dense correspondences (texture shows corr.)

Figure 1: *Finding subspace symmetries. We use features to find initial guesses for the correspondences that span the subspace. For an input model (a), we identify crease lines using extremal principal curvatures (b), and extract graphs of such crease lines (c). Subgraphs are then matched and refined using a learned subspace model to establish initial sparse correspondences (d). Finally, dense correspondences are estimated using a regularized subspace fitting technique (e).*

1): Assuming that the mapping functions approximate preserve invariance of chosen features, the algorithm first finds patterns of features that are linearly correlated. Starting from the initial set of sparse matches, we use a variational regularizer to extend the mappings to dense correspondences and build the final subspace. We then use this knowledge to refine further search results. Although the approach is limited by the design and choice of the feature detection scheme, to the best of our knowledge, this is the first algorithm for automatic detection of subspace symmetries. To overcome the restrictions, we also support a semi-supervised variant of the algorithm that allows the user to bootstrap the subspace learning by providing a few hints regarding suitable features for processing challenging input models, especially noisy and incomplete ones. We tested our algorithm on a set of synthetic benchmark scenes as well as real world 3D scanner data in order to study its performance. For scenes with clear feature structure (see Figure 1), we detect the dominant salient symmetries fully automatically. In challenging and ambiguous scenes, however, we require the user to mark a few points to guide the feature detector. Applications to non-local non-rigid denoising and shape completion are also presented.

Our main contributions are – introduction of subspace symmetries that define symmetry in geometric data without restrictions to simple, parameterized mapping functions along with practical algorithms to efficiently compute such symmetries and the applications thereof.

2. Related Work

Shape analysis. Symmetry detection, a central topic in shape analysis, is widely used for pattern detection and regularity analysis in images and in 3D geometry. An object is said to be symmetric if it is (partially) invariant under the action of allowable symmetry transforms. A common approach is to identify a set of candidate transforms derived using potential correspondences, and map the correspondences to a space of transforms. This remodels the global problem of symmetry detection to local identification of clusters in the transformation space [LE06, MGP06, PSG*06]. The approach has been extended to detect (commutative) Euclidean regularity [PMW*08] and isometric regularity [MBB10] in 3D geometry. Such techniques are designed to handle transformation families that can be compactly represented using a few parameters, e.g., translation, rotation, uniform scaling. However, generalizing the concept to handle other transformations involving many more parameters is challenging due to the ambiguity in the mappings, and difficulty in identifying good set of potential correspondences. Further, it is difficult to extend such techniques to learn data dependent allowable variation modes.

Enumeration based methods including geometric hashing [GCO06], robust auto-alignment [SKS06], spherical harmonics analysis [MSHS06], primitive fitting [SWWK08] are also inapplicable given the high dimensionality of the non-rigid transformation space. Ovsjanikov et al. [OSG08] perform global intrinsic symmetry detection using eigen-analysis of the surface Laplace-Beltrami operator, while Xu et al. [XZT*09] extend Euclidean symmetry transform [PSG*06] using a computed scalar field to detect partial intrinsic reflection symmetry. Lasowski et al. [LTSW09] perform partial intrinsic symmetry detection based on a Markov random field model. Recently, Kim et al. [KLCF10] propose a robust algorithm based on Mobius voting for global intrinsic symmetry detection. It is unclear how to extend these methods, specifically designed to handle intrinsic symmetries, to handle more general transformations.

Graph-based symmetry detection. Symmetry detection can be formulated as an instance of partial graph matching. Graphs, constructed using feature lines or curves as nodes and their respective aligning transformations as edges, are analyzed to identify repeated subgraphs revealing partial symmetries. The main challenge of such feature-based methods is to compute reliable and stable features from noisy and incomplete data. Berner et al. [BBW*08] use regions of minimum slippability, while Bokeloh et al. [BBW*09b] employ crease lines as features, which yield particularly useful cues for shape matching. These algorithms rely on rigid alignment of local line patterns using iterative-closest-lines.

Global Shape Registration. In shape retrieval, topological matching techniques have been used to recognize semantically similar shapes, e.g., [HSKK01]. Recently, Zhang et al. [ZSCO*08] propose a global shape matching framework based on comparing graphs of extremity features and evaluating the induced deformation of an assignment in order to match shapes such as humans or animals. The use of an elastic deformation model, however, limits the variability of models that can be handled. This problem has been partially addressed in [ATCO*10], where objects are reduced to skeletons and graph matching, and multi-dimensional scaling is employed to find similar skeletons. A similar idea has also been explored for global registration of animation sequences of 3D scans of deformable objects [ZST*10]. Existing methods for densely matching significantly dissimilar objects mostly assume a smooth mapping function such as thin plate splines [ACP03], or work with restricted statistical models [HSS*09], and require manual initialization of each match.

Dimension reduction. Projections on low dimensional affine (or non-linear [SSM98]) subspaces have been used in a large number of computer vision and graphics applications. Eigenfaces [KS90, TP91] use low dimensional spaces to model photographs of human faces. The method was extended to handle geometric data by Blanz and Vetter [BV99] in their highly influential work to construct a PCA space of faces from a collection of 3D range scans of images registered using optical flow. Allen et al. [ACP03] extend the method to match different human body shapes using local optimization guided by manually annotated markers. Establishing dense correspondence allows the use of statistical learning techniques to describe spaces of plausible shapes, poses, and dynamics [ASK*05, SZGP05, HSS*09], and also for specific families of objects such as the shape of car bodies [KFS*07]. We explore affine 3D shape spaces with correspondences for symmetry detection – a direction that has been unexplored by previous methods and no global unsupervised or semi-supervised partial matching method has been provided in the cited work.

3. Subspace Symmetries

In this section, we introduce the notion of *subspace symmetries*. Our goal is to produce an *output* model comprising of a set of shapes in correspondence such that they form an affine shape space. The key challenge is to simultaneously estimate shape spaces and their associated correspondences.

Affine shape spaces. Let a set of shapes $\mathcal{S} := \{\mathcal{S}_1, \ldots, \mathcal{S}_k\}$ be in correspondence with each shape $\mathcal{S}_i := (V^{(i)}, E)$ being represented as a set of vertices $V^{(i)} = \{\mathbf{v}_1^{(i)}, \ldots, \mathbf{v}_n^{(i)}\}$ sharing the same connectivity structure encoded by a set of edges E. Thus, each shape $\mathcal{S}_i$ can be considered a point in a $3n$-dimensional shape space, i.e., $\mathcal{S}_i \in \mathbb{R}^{3n}$. Assume that the respective points of $V^{(i)}$ across shapes are in correspondence.

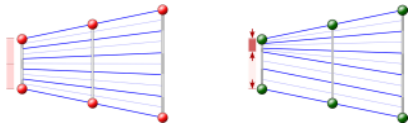


Figure 2: *Affine subspace correspondences are not always unique. The three flat shapes form a 1-dimensional shape space, but the two examples show two different solutions to the correspondence problem.*

The set $\mathcal{S}$ is said to be spanned by independent *basis* shapes $\mathcal{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_d\}$ and a mean shape $\mathbf{b}_0$ with $\mathbf{b}_i \in \mathbb{R}^{3n}$ sharing the same edge connectivity E, if and only if, each shape $\mathcal{S}_i$ can be *uniquely* expressed as:

$$\mathcal{S}_i = \mathbf{T}_i \left(\mathbf{b}_0 + \sum_{k=1}^{d} \lambda_k^{(i)} \mathbf{b}_k \right), \tag{1}$$

with $\lambda_k^{(i)}$ being scalar coefficients and addition referring to vector addition of respective elements of the vertex sets. Since such a linear space does not represent rotations well, we additionally store a rigid transformation $\mathbf{T}_i$ for each instance $\mathcal{S}_i$. Thus, shape S_i can be compactly encoded as $\{\lambda_1^{(i)}, \ldots, \lambda_d^{(i)}\}$ and its rigid placement $\mathbf{T}_i$ in the scene. Given an example set $\mathcal{S}$, we can compute a model according to Equation 1 using a polar decomposition to extract the rigid motion and PCA to retrieve the subspace. The mean vector of $\{V^{(i)}\}$ yields $\mathbf{b}_0$, while covariance analysis of the mean centered vertex sets $\{V^{(i)} - \mathbf{b}_0\}$ produces an orthogonal basis set $\{\mathbf{b}_1, ..., \mathbf{b}_d\}$, which spans the affine subspace. Further, the respective covariance values $\sigma_1, ..., \sigma_d$ along the principal directions encode the likelihood of variations by a positive definite quadratic form. This information can be used to estimate the likelihood of a given shape to lie in this space of variations.

Correspondences. In the following, we use $\mathbf{f}_{i \to j}$ to denote the function that maps points $\mathbf{v}_k^{(i)}$ to their corresponding points $\mathbf{v}_k^{(j)}$ for $k \in \{1, \ldots, n\}$. It is easy to see that these functions, after factoring out the rigid components, form an affine subspace of dimension d for any fixed i as well; correspondence functions could be seen as an alternative parametrization of the space. This also holds for correspondences from the mean shape $\mathbf{b}_0$ to models in $\mathcal{S}_i$.

Fitting the model to data. Once we know an affine model of the subspace symmetries, we can robustly perform model completion from imperfect data. Thus given a noisy and incomplete shape $\mathbf{s}$, we project the shape to the base space $\mathcal{B}$ by optimizing for coefficients $\{\beta_1, \ldots, \beta_d\}$ that best represent $\mathbf{s}$ in the least squares sense. More specifically, our goal is to solve the optimization:

$$\min_{T, \{\beta_k\}} d\left(\mathbf{s}, \mathbf{T} \left(\mathbf{b}_0 + \sum_k \beta_k \mathbf{b}_k \right) \right), \tag{2}$$

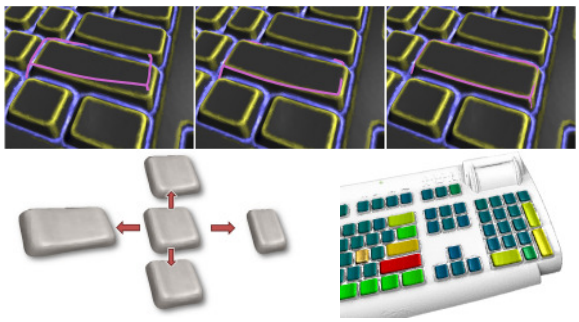


Figure 3: *Subspace assumption helps in refining matches. (Bottom-left) The two principal eigenmodes of the "key" element, (top) iterative fitting, alternating between rigid alignment and subspace coordinates, (bottom-right) rainbow color coded dominant eigenmode for signed subspace coordinates for all detected matches, blue being minimum.*

where, T denotes a rigid transform, and $d(\mathbf{a},\mathbf{b})$ is a suitable (squared) distance measure between two shapes $\mathbf{a}$ and $\mathbf{b}$. In this paper, we use the sum of squared point-to-plane distances (with truncation for far away points to improve outlier robustness) as our distance measure. We solve this optimization problem by iteratively computing the best rigid match to closest model points, using iterated closest point (ICP), and projecting the corresponding points into the PCA space. Later, we describe how to use feature based matching to initialize the optimization.

Invariance and regularization. In general, the restriction that the set of symmetry instances have to form an affine shape space of low dimension is not sufficient to uniquely establish correspondences across these shapes. For example, for corresponding planar regions in a shape we can find multiple different correspondence functions that form an affine subspace (see Figure 2). We therefore employ an additional (weak) regularizer: We minimize the spatial second derivatives of the correspondence functions $\mathbf{f}_{i\to j}$. This means, among all ambiguous solutions, we prefer the one with the least spatial bending of the correspondences functions as measured using a "thin plate spline" regularizer.

4. Extracting Subspace Symmetries

In this section, we discuss how to identify subspace models from input geometry that are used to find symmetric parts. The main challenge is to find correspondences that actually span a low dimensional affine shape space. Without any a-priori knowledge, this is challenging due to the combinatorial nature of the problem. A simple brute-force search would require computation time exponential in the number of correspondences involved. We therefore propose an algorithm that is based on the additional assumption of the availability of *invariant* features: We assume that corresponding pieces of geometry show features that are invariant under the space of mappings $\mathbf{f}_{i\to j}$. On the other hand, matching features do not necessarily imply right correspondence.

We detect an arrangement of such features (Section 4.1) and use them to initialize a subspace search algorithm that establishes dense subspace correspondences (Section 4.2). Although this fully automatic symmetry detection approach works well on clean models, it may fail on challenging cases when the input is severely corrupted with noise and has large missing parts. Such failures arise in the initialization stage as our assumption on invariant features breaks down. In such cases, we propose a semi-supervised extraction strategy (Section 4.3) to allow the user to annotate a few training correspondences to seed the search for subspace variations.

Note that our feature matching strategy is not the only possibility, but other variations are conceivable, leading to similar or improved results. The focus of this paper, however, is not on feature detection. Instead our goal is to detect subspace symmetries (mostly) automatically, but in a practically feasible way.

4.1. Graph Matching

We now present our algorithm to find and match invariant features (see Figure 1). First, we extract a graph of features that is resilient to moderate deformations. The graph extraction is motivated by the observation that relationships across feature points and their connecting feature lines are better preserved under deformations, as compared to absolute geometry. We use crease lines of high curvature that have been demonstrated to capture shape characteristics [OBS04]. Bokeloh et al. [BBW*09b] have effectively used such feature graphs for Euclidean symmetry detection. We present an overview of the graph creation algorithm and refer to our technical report for further details [BBW*09a].

Feature graphs. Given the input model (Figure 1a), we estimate the principal curvature values and directions at every point of the model using moving-least-squares fitting [DB02] (Figure 1b). We then threshold the resulting scalar field using a user specified threshold, keeping only points with large absolute maximum curvature. Next, we apply a morphological erosion operation to shrink the remaining area to thin lines and crossings. We place feature points at each line end and at each intersection. We then construct the feature graph with the feature points as nodes and the intermediate line segments acting as edges (Figure 1c).

Pairwise graph matching. We now use the computed graph to search for candidate matches. Based on our assumption that partial similarity between adjacent feature lines remains invariant under moderate deformations, detecting repetition patterns amounts to solving a partial graph matching problem. To this end, we employ a greedy algorithm. We extract a small number of example correspondences using a simple and conservative approach to initiate the creation of

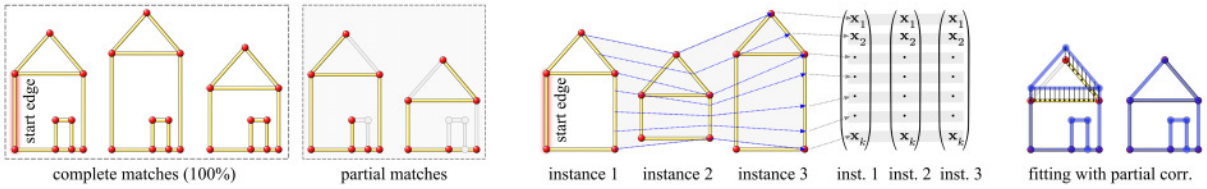


Figure 4: *Building and using a discrete subspace model. (a) Complete and incomplete graph matches; (b) the algorithm builds a model of the complete matches by parametrizing edges by edge length and mapping corresponding points to vectors that are used in a PCA analysis; (c) partial matches are refined and refitted using the learned PCA space.*

the subspace model. Later, we extend the model by detecting other repeated instances using a constrained search in the discovered subspace (see Section 5).

For matching, each edge in the graph is tagged by a number of attributes. We store the intrinsic length, as well as the average principal curvature values of points along the crease lines that the edges follow. We retain the signs to discriminate between ridges and valleys. Each edge in the graph is additional annotated by the outdegree of its two adjacent vertices, and the angles between the edge and further edges adjacent to the vertices.

We use a simple rule that edges can only be matched if their length values are within a certain ratio of each other (3 in our experiments), the value of the average crease curvature matches up to a constant factor (3 in our experiments) and the outdegree of adjacent vertices matches up to ± 1. In addition, pairs of edges are matched if their intermediate angles agree up to a threshold (± 10 degrees). All these user defined threshold define the matching model.

We explore potential matches in the graph by sampling and region growing. Starting from a random seed edge, we collect all its potentially matching edges. Then, we test each pairwise match against its candidate matching edges. For each pairwise match, we greedily grow the match by adding adjacent edges to the current match if they satisfy the matching criterion.

In our algorithm, we discard matches that do not have a matching graph topology, i.e., edges that form a loop on the source side must also form the same loop at the target side. This is important as the graph topology carries a strong cue for identifying correct matches. However, as we allow for over-segmentation of edges, i.e., additional vertices, we skip over vertices based on their deviation from the expected parameters. Specifically, among all the possible vertices, we select the vertex with the smallest deviation from the expected metric parameters.

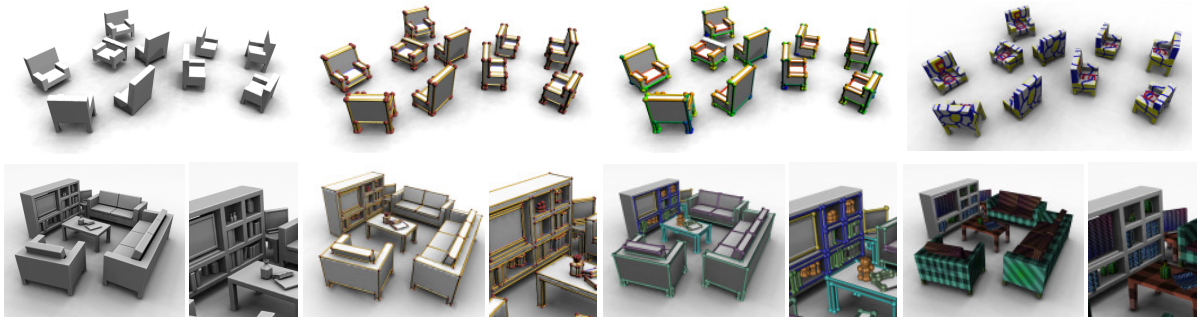
We repeat this several times starting from independent random seeds and retain only the best matches, i.e., the ones containing the largest number of matched edges and instances. In all our experiments, we perform 200 rounds of random sampling and keeping matches involving at least 5 graph elements.

Multiple instances. After the initialization step, we have a list of partial subgraph matches. Each match starts from a source graph containing the start edge, and maps to different target graphs, only partially overlapping in the source domain. For further processing, we only handle instances with sufficient overlap, thus describing the same instance. Therefore, we delete all matches without substantial overlap (typically, 60% of the candidate matches) with the largest found match. The remaining partial graph matches are used in further processing (Figure 4a).

We also determine all the complete matches and use them to initialize the subspace model. The partial matches are used as candidates to be matched to the subspace model. For increased robustness, we iterate the whole graph matching pipeline multiple times with random seeds in an outer loop (100 iterations in our experiments). We keep the solution that maximizes the product of the number of instances that are in complete correspondence and the number of edges involved.

Building a discrete subspace model. We now compute a subspace approximation of the candidate salient line patterns, which are likely to correspond to actual subspace symmetric geometry. First, we establish correspondence across the line segments using a simple arclength parametrization (Figure 4b), normalized to overall unit length. Points at the same distance from the start vertex are then set to be corresponding. We sample each edge uniformly and form a long vector of corresponding points. We compute a subspace representation of the form of Equation 1 using PCA, with rigid transforms factored out, as described in Section 3. We keep those eigenmodes with eigenvalues of at least 10% of the (unsigned) magnitude of the largest.

Discrete refinement. Using the learned subspace, we now refine the remaining matches by considering the previously unconsidered partial matches. The local geometry of the candidates are validated in the subspace model of the feature lines (Figure 4c) by minimizing Equation 2, as discussed in Section 3. For increased robustness, we only use the previously extracted points of high curvature as target shape for the alignment since the feature lines cannot map to flat regions. After matching, we measure the Mahalanobis distance to the PCA model, and discard matches with a distance larger than three times the standard deviation. Note



(a) input model (mesh) (b) feature graph (c) discrete matches (d) dense correspondences

Figure 5: *Detected subspace symmetries on two synthetic examples. Dense correspondence visualized using mapped textures.*

that our model space is significantly low rank in proportion to the high dimensional embedding shape space. Hence, a slight amount of random noise can lead to, with high probability, unstable Mahalanobis distance. Therefore, we assume a small uniform covariance of $\sigma^2\mathbf{I}$ in all directions, and add this to the covariance obtained from the PCA analysis. σ corresponds to the noise level in the data (including small effects not modeled by our subspace model).

Note that the refinement step is critical to a viable solution to the subspace symmetry detection problem. Importantly, the subspace symmetries can be initialized with a small amount of data. Subsequently, we use the model to identify and verify candidate matches arising out of partial and ambiguous data. Figure 1d shows the final result of the discrete matching after discrete refinement, while Figure 3 shows how PCA coordinates can be learned and how a partial example match is found.

4.2. Dense Subspace Correspondences

Corresponding feature line skeletons that form a low-dimensional subspace are good indicators that the dense geometry points enclosed by the cells of these feature graph also form a low dimensional subspace. To test such candidates, we bring the dense points into correspondence and then again perform PCA while factoring out rigid alignment.

In order to extend the discrete matches to dense correspondences, we use a regularized deformable ICP similar to the deformation framework introduced by Allen et al. [ACP03]. We cut out the dense geometry $\mathcal{S}_0$ enclosed by the source graph, and match back onto the original input geometry $\mathcal{S}_1$ at a different location, where a second instance has been detected by the discrete matching. We denote the matching function by $\mathbf{f} : \mathcal{S}_0 \rightarrow \mathbb{R}^3$. We regularize f using a thin-plate-spline energy (as motivated in Section 3), seeding the optimization using the computed discrete matches as known boundary condition correspondences.

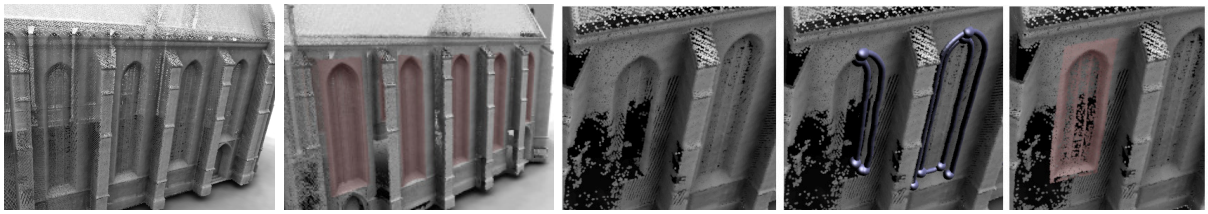
Let the known discrete point-to-point matches, obtained by sampling along the edges of the feature graph, be denoted as: $(\mathbf{x}_i, \mathbf{f}_i), i = 1, \ldots, k$, where $\mathbf{f}_i$ denotes the known value for $\mathbf{f}(\mathbf{x}_i)$. Overall, we minimize the following least-squares energy function:

$$E(\mathbf{f}) := \lambda_d \int_{\mathcal{S}_0} d\left(\mathbf{f}(\mathbf{x}), \mathcal{S}_1\right) d\mathbf{x} + \lambda_r \left\|\mathbf{H}_\mathbf{f}\right\|_F^2 d\mathbf{x} + \lambda_b \sum_{i=1}^{K} \left\|\mathbf{f}_i - \mathbf{f}(\mathbf{x}_i)\right\|^2$$

where, $\mathbf{H}_\mathbf{f}$ a 3×9 matrix containing all second derivatives of $\mathbf{f}$. In other words, $\mathbf{H}_\mathbf{f}$ is a concatenation of the Hessian matrices of all three component functions of $\mathbf{f}$ that describe the mapping in the (output) x-, y-, and z-directions, with the second derivatives taken with respect to the input x-, y-, and z-directions. We take the Frobenius norm of this matrix, thereby uniformly penalizing spatial curvature of the mapping using the well-known thin-plate spline energy. The weights λ_d, λ_r and λ_b balance the effects of the (squared) distance to the target surface, smoothness of the solution, and the deviation from the boundary conditions, respectively. We start with $\lambda_d = 0$ and progressively increase the weight for the data term in course of the optimization to obtain a tight fit. The contribution due to the Hessian component is controlled by a user parameter λ_r. For known correspondences, the minimization is convex and solved using a linear system. Therefore, the first iteration results in a globally optimal alignment according to the feature matches. Thus we have a reliable pre-alignment for the later steps. We now introduce the data term, and the optimization is performed iteratively. In each subsequent iteration, the distance function d is updated using the (squared) point-to-plane distance to the closest model point. We discretize $\mathbf{f}$ on a simple spatial finite-differences grid and solve the resulting sparse linear system using conjugate gradients.

We perform this matching for each of the found (discrete) symmetries, thus resulting in dense correspondences encoded by a number of matching functions $\mathbf{f}_i := \mathbf{f}_{1 \rightarrow i}$, where we use index one to denote our start instance, i.e., the initial location of the start edge. Finally, we compute a space $\mathcal{S} = \{\mathcal{S}_1, \mathbf{f}_2(\mathcal{S}_1), \ldots, \mathbf{f}_k(\mathcal{S}_1)\}$ to compute the final subspace model using rigid alignment followed by a PCA analysis.

(a) input model (mesh) (b) feature graph (c) discrete matches (d) dense correspondences



(e) Reconstruction: The deformed instances are overlaid and projected to the subspace for denoising and detail transfer.

(f) Hole filling: a partial match is identified and filled with matching geometry from the subspace

Figure 6: *Top row: Automatic detection of subspace symmetry on a scanned point cloud data of a church ("Marktkirche" in Hannover). Bottom row: Automatic non-local non-rigid denoising and hole filling on the same model.*

4.3. Semi-Supervised Algorithm

When the input objects have significant variations or are highly corrupted by noise, our initial discrete feature graph correspondences can be unstable, and fail to suitably seed the symmetry subspace search. In such cases, we provide an optional mechanism to train the feature matching model in a semi-supervised fashion. Although better feature detectors may produce slightly improved performance, we believe that in certain cases user inputs are unavoidable to provide unambiguous leads for a reliable creation of initial subspace candidates, which can then be automatically refined and other instances learned.

We ask the user to click on a small number of feature points that should be in correspondence. The algorithm computes a descriptor of the local geometry for each point using histograms of intrinsic distances to nearby samples within a neighborhood ball. We build a PCA model of the descriptors and add a regularizer $\sigma_{descr}\mathbf{I}$ to the covariance matrix to account for noise. The user interactively specifies σ_{descr} checking when false positives start appearing. Using the PCA model, we automatically detect all points on the model with descriptors that fall within the covariance of the model, using a maximum Mahalanobis distance of three sigma. Next, we connect the feature points using geodesic paths on the underlying surface. We simultaneously grow regions from all the feature points to compute intrinsic Voronoi regions, and join only those point-pairs that have a connecting edge in the dual triangulation.

5. Applications

Instance replacement. The detected correspondence fields $\mathbf{f}_{i\to j}$ between all instances enable non-local edits, which are otherwise difficult to perform. Base geometric patches can now be easily textured, altered, and replaced, and the edits automatically propagated to all the symmetric instances. We use this tool to show the correspondence mapping across the instances.

Shape completion. Once a subspace model is discovered, we can robustly detect partial matches by verifying potential candidates against the subspace. This allows us to automatically repair incomplete geometry, which is common due to occlusion and scanning artifacts. We use the partial feature graph to initialize the matching and compute the least squares best fitting mapping function $\mathbf{f}$ by projecting the constraints into the subspace. Finally, we deform the base geometry using the established spatial mapping function $\mathbf{f}$ for shape completion.

Denoising. We use the established correspondences for scan denoising. This mode is particularly interesting since we can establish general, non-rigid mappings that previous techniques fail to detect. For denoising, we first compute the mapping functions from the mean shape $\mathbf{b}_0$ to all other instances as $\mathbf{f}_{mean\to i}$. Using the inverse mappings $\mathbf{f}^{-1}_{mean\to i}$ we then copy all the data points to the mean shape domain. Finally, we use a standard moving least squares reconstruction in the mean domain for denoising, and transfer back the results using the original maps $\mathbf{f}_{mean\to i}$.

6. Results and Discussion

We tested our algorithm on a variety of models, both synthetic and scanned. All the scanned model point clouds are reconstructed using Poisson surface reconstruction to generate approximately isotropic meshes as inputs.

Results of our algorithm are depicted in Figure 1, 5, 6, and

7. We visualize the discrete matches using the same color as the corresponding graph elements. In order to display the dense correspondence, we paint a texture on one of the instances and transfer it to the other instances using the computed deformation fields $\mathbf{f}_i$. Several examples contain self-symmetries, such as a chair that can be mirrored along its side and mapped back to itself. Our algorithm detects all (for synthetic) or most (for scanned examples) of these additional symmetries. However, for clarity of presentation we show only the global self-symmetry of the instances: We only visualize the match whose rigid component has the smallest deviation, in the Frobenius sense, from the identity map.

Synthetic test scenes. We first report experimental results on clean synthetic datasets, which are not corrupted with noise, and are complete. For such models, feature matches provides good seeds for the subspace search, and we reliably obtain perfect results, i.e., our algorithm finds all the non-rigid self-similarities that we expect to extract from the data (see Figure 5). As our algorithm works at the level of feature graphs, the matching results do not depend on the tessellation of the meshes, as long as the underlying geometry and topology remain unchanged.

Real-world 3D scans. The problem quickly becomes challenging for scanned inputs, which usually contain various artifacts including ambiguity due to noise, missing data, and allowed non-rigid deformations. Our first dataset is a scanned PC keyboard (see Figure 1). This data set reveals a clear feature structure that is extracted by our automatic algorithm. As shown in Figure 3, we obtain a subspace model with two dominant directions of variations, which are sufficient to match all keys in the keyboard with high precision. Thanks to the initial Poission surface reconstruction, our algorithm is quite robust under increasing noise levels. Figure 9 shows the same scene with artificial Gaussian noise added to the original scan. Recognition becomes problematic only after the noise level starts blurring out the feature lines. Figure 6 shows the results for a LiDAR scan of a 14th century gothic church – the *Marktkirche* from the well-known scan repository provided by the IKG of Hannover University. Our algorithm detects the repeated windows, even under non-uniform scaling, and subsequently establishes dense correspondences across the instances.

Next, we use a scan of a small clay replica of a statue (Figure 7, top), which provides a challenging example due to variations in clay instances. Again we automatically discover most of the ornaments below the neck of the statue. However, to resolve the *grid ambiguity*, we manually designate the "cut-out" of one instance (shown in purple). Otherwise, the algorithm detects larger instances with several pieces combined that are mapped in groups. Although correct, this creates a large, overlapping set of detected parts. A discrete group reduction algorithm [MGP06, PMW*08] acting on the discrete permutation group of the detected feature correspondences could resolve this issue automatically. As grid detection is not the main focus of this work, we leave this for future work.

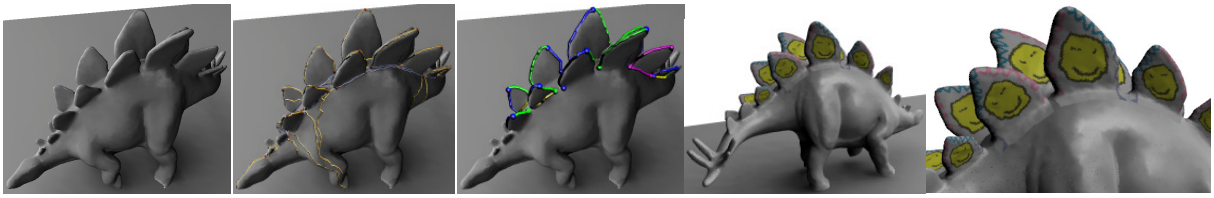
The automatic mode fails on the most complicated example, as shown in Figure 7 (bottom), since we fail to get a good set of seeds for initial subspace construction. Therefore, we allow the user to click on example features (two different tips of the scales, two more examples at the bottom). Afterwards, one example graph (one scale) is selected. This information is sufficient to remove distracting features and recognize all the major scales on the back of the dinosaur. We obtain a subspace with overall four main directions. In particular, the first two are intuitively interpretable, encoding size and skewness of the scales (see Figure 8a-b and supplementary video). Subsequent dense correspondence establishment works well.

Shape completion and denoising. We demonstrate denoising and hole filing on the Hannovarian church data set (Figure 6e,f). The algorithm detects a partial match and fills in geometry that is closest in a least-squares projection to the learned subspace. Similarly, we combine the geometry information across all the instances and perform a non-local, non-rigid denoising. This reproduces fine details and sharp edges better than in any single instance of the original scan. However, some subtle details that vary across the instances are lost by our algorithm when working with the available scan resolution (by looking very closely at the original data, one can guess that there are a different number of glass elements in large vs. small windows).

All the examples ran in the order of 5-10 minutes on a 2.5 GHz Core2Duo laptop with 8GB RAM with unoptimized code with the following breakup: each RANSAC step inner loop ran in about 5-30 seconds depending on graph complexity, line PCA building took around 30 seconds, and instance refinement order of few seconds, and finally, the most expensive dense correspondence ran in around 3-8 minutes depending on the complexity of the symmetries and the grid resolution of thin plate spline solver, with dense PCA taking less than a minute.

Limitations. The main limitation of our subspace symmetry analysis algorithm stems from the assumption that the chosen feature detector can detect enough nodes to seed a subspace model search. In presence of significant noise or large missing parts, this assumption breaks down for our choice of feature curves, thus forcing the algorithm to switch to a semi-supervised mode. Even then, our model does not capture all possible cases. For example, the structures on the wing of the "Gargoyle" model in Figure 9d do not lead to intersecting graph edges. The other line pattern on the rest of the model are also not clear enough to seed subspace models (even with some user interaction). Furthermore, the features guided matching in general has the drawback that it involves manual setting of parameter values. For models without a clear crease-line structure, fully automatic matching is usually not possible. However, this problem is reduced if we

(a) input data (b) feature graph (c) discrete matches: color encodes corresp. parts, purple = designated grid element (d) dense correspondences



(e) input model (mesh) (f) feature graph (g) discrete matches (h) dense correspondences

Figure 7: *Top: Statue scan; matching is fully automatic. However, we chose one grid element manually to resolve grid ambiguities. Bottom: Dinosaur model scan. The user specifies* 4 *example features and one subgraph(purple) to bootstrap the detection.*

(a) chairs eigenmodes (b) dino eigenmodes (c) coordinate of first eigenvector (shown as rainbow color map)

Figure 8: *(a, b) Shapes along the two dominant eigenmodes for the detected parts on the chair and dino datasets, respectively. (c) Visualization of the normalized signed coefficients along top eigen-direction, using the rainbow map, blue being minimum.*

allow the user to click on a few example features in order to prune out irrelevant features and spurious matches. Also the success of the approach is dependent on the richness of the subspace symmetry present in models. We found many man-made models to be a rich source of such symmetries, as shown in our experiments. Note, there remains a fundamental tradeoff between the compactness and variability of the subspace, the amount of noise that can be handled, and the number of feature points that are required to seed the candidate set of discrete sub-graph matches.

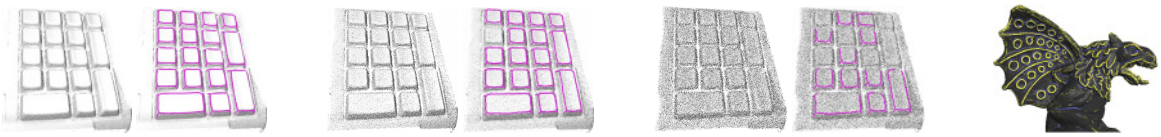
Conclusion and future work. We introduced subspace symmetries to capture similarity between surface geometry, which are related by non-rigid transformations that are not arbitrary but span a low-rank subspace. The resultant symmetry subspace then has a natural compact description, and effectively captures the variations of the underlying surface. We presented an algorithm to detect such symmetries, both automatically and also in a semi-supervised mode. The subspace symmetry detection algorithm was tested on various classes of inputs. Further, the extracted symmetry subspaces enable a range of interesting geometry processing tasks including non-local non-rigid denoising, model completion, simultaneous instance replacement, while factoring out the underlying subspace variations.

In the future, it will be interesting to further explore means to compactly encode model variations, and find low-rank subspaces of interest in the general space of shapes. Capturing the model variations and the distributions of the embedding parameters can allow efficient generation of statistical variations in shape spaces, thus producing subtle data-driven variations in shape families. Finally, further efforts are needed to explore alternate feature descriptors that can robustly initialize and capture subspace symmetries.

Acknowledgements. This work has been partially funded by the Cluster of Excellence "Multi-Modal Computing and Interaction" and the IMPECS collaboration network. The authors wish to thank the anonymous reviewers for their valuable comments and Martin Bokeloh for his help.

References

[ACP03] ALLEN B., CURLESS B., POPOVIĆ Z.: The space of human body shapes: reconstruction and parameterization from range scans. In *Proc. ACM SIGGRAPH* (2003). 3, 6



(a) original scan (b) noise level 0.2% BB (c) noise level 0.5% BB (d) "Gargoyl" data set

Figure 9: *(a)-(c) noise test; noise levels are the standard deviation of Gaussian additive noise as percentage of the largest side of an axis aligned bounding box of the whole scene. (d) The "Gargoyl" statue is a failure case (see text).*

[ASK*05] ANGUELOV D., SRINIVASAN P., KOLLER D., THRUN S., RODGERS J., DAVIS J.: SCAPE: shape completion and animation of people. *Proc. ACM SIGGRAPH 24*, 3 (2005). 3

[ATCO*10] AU O. K.-C., TAI C.-L., COHEN-OR D., ZHENG Y., FU H.: Electors voting for fast automatic shape correspondence. In *CGF (EUROGRAPHICS)* (2010), vol. 29. 3

[BBW*08] BERNER A., BOKELOH M., WAND M., SCHILLING A., SEIDEL H.-P.: A graph-based approach to symmetry detection. In *Proc. Symp. Point-Based Graphics 2008* (2008). 2

[BBW*09a] BERNER A., BOKELOH M., WAND M., SCHILLING A., SEIDEL H.-P.: *Generalized intrinsic symmetry detection*. Research Report MPI-I-2009-4-005, Max-Planck-Institut für Informatik, August 2009. 4

[BBW*09b] BOKELOH M., BERNER A., WAND M., SEIDEL H.-P., SCHILLING A.: Symmetry detection using line features. *Computer Graphics Forum (Eurographics 2009)* (2009). 2, 4

[BV99] BLANZ V., VETTER T.: A morphable model for the synthesis of 3d faces. In *Proc. SIGGRAPH* (1999). 3

[DB02] DOUROS I., BUXTON B.: Three-dimensional surface curvature estimation using quadric surface patches. In *Proc. Scanning* (May 2002). 4

[GCO06] GAL R., COHEN-OR D.: Salient geometric features for partial shape matching and similarity. *ACM Trans. Graph. 25*, 1 (2006), 130–150. 2

[HSKK01] HILAGA M., SHINAGAWA Y., KOHMURA T., KUNII T. L.: Topology matching for fully automatic similarity estimation of 3d shapes. In *Proc. ACM SIGGRAPH* (2001), pp. 203–212. 3

[HSS*09] HASLER N., STOLL C., SUNKEL M., ROSENHAHN B., SEIDEL H.-P.: A statistical model of human pose and body shape. *CGF (EUROGRAPHICS) 28*, 2 (2009). 3

[KFS*07] KÓKAI I., FINGER J., SMITH R. C., PAWLICKI R., VETTER T.: Example-based conceptual styling framework for automotive shapes. In *Eurographics Workshop on Sketch-Based Interfaces and Modeling* (2007). 3

[KLCF10] KIM V. G., LIPMAN Y., CHEN X., FUNKHOUSER T.: Mobius transformations for global intrinsic symmetry analysis. *Proc. Eurographics Symp. on Geometry Processing* (2010). 2

[KS90] KIRBY M., SIROVICH L.: Application of the karhunen-loeve procedure for the characterization of human faces. *IEEE PAMI 12*, 1 (1990), 103–108. 3

[LE06] LOY G., EKLUNDH J.: Detecting symmetry and symmetric constellations of features. In *Proc. Euro. Conf. on Comp. Vis.* (2006), pp. 508–521. 2

[LTSW09] LASOWSKI R., TEVS A., SEIDEL H.-P., WAND M.: A probabilistic framework for partial intrinsic symmetries in geometric data. In *IEEE International Conference on Computer Vision (ICCV)* (September 2009). 2

[MBB10] MITRA N. J., BRONSTEIN A., BRONSTEIN M.: Intrinsic regularity detection in 3d geometry. In *Proc. Euro. Conf. on Comp. Vis.* (2010), p. to appear. 2

[MGP06] MITRA N. J., GUIBAS L. J., PAULY M.: Partial and approximate symmetry detection for 3d geometry. *Proc. ACM SIGGRAPH 25*, 3 (2006), 560–568. 1, 2, 8

[MGP07] MITRA N. J., GUIBAS L., PAULY M.: Symmetrization. In *Proc. ACM SIGGRAPH* (2007), vol. 26. 1

[MSHS06] MARTINET A., SOLER C., HOLZSCHUCH N., SILLION F.: Accurate detection of symmetries in 3d shapes. *ACM Trans. on Graphics 25*, 2 (2006), 439 – 464. 2

[OBS04] OHTAKE Y., BELYAEV A., SEIDEL H.-P.: Ridge-valley lines on meshes via implicit surface fitting. In *Proc. ACM SIGGRAPH* (2004), pp. 609–612. 4

[OSG08] OVSJANIKOV M., SUN J., GUIBAS L.: Global intrinsic symmetries of shapes. In *Proc. Eurographics Symp. on Geometry Processing* (2008). 2

[PMW*08] PAULY M., MITRA N. J., WALLNER J., POTTMANN H., GUIBAS L.: Discovering structural regularity in 3D geometry. *Proc. ACM SIGGRAPH 27*, 3 (2008). 1, 2, 8

[PSG*06] PODOLAK J., SHILANE P., GOLOVINSKIY A., RUSINKIEWICZ S., FUNKHOUSER T.: A planar-reflective symmetry transform for 3D shapes. *Proc. ACM SIGGRAPH 25*, 3 (2006). 1, 2

[SKS06] SIMARI P., KALOGERAKIS E., SINGH K.: Folding meshes: hierarchical mesh segmentation based on planar symmetry. In *Proc. Eurographics Symp. on Geometry Processing* (2006), pp. 111–119. 2

[SSM98] SCHÖLKOPF B., SMOLA A., MÜLLER K.-R.: Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation 10*, 5 (July 1 1998), 1299–1319. 3

[SWWK08] SCHNABEL R., WESSEL R., WAHL R., KLEIN R.: Shape recognition in 3d point-clouds. In *Proc. Conf. in Central Europe on Computer Graphics, Visualization and Computer Vision* (2008). 2

[SZGP05] SUMNER R. W., ZWICKER M., GOTSMAN C., POPOVIĆ J.: Mesh-based inverse kinematics. *Proc. ACM SIGGRAPH 24*, 3 (2005), 488–495. 3

[TP91] TURK M., PENTLAND A.: Face recognition using eigenfaces. In *Proc. IEEE Conf. on Comp. Vis. and Pat. Rec.* (1991), pp. 586–591. 3

[XZT*09] XU K., ZHANG H., TAGLIASACCHI A., LIU L., LI G., MENG M., XIONG Y.: Partial intrinsic reflectional symmetry of 3d shapes. *ACM Transactions on Graphics, (Proceedings SIGGRAPH Asia 2009) 28*, 5 (2009), Article 138. 2

[ZSCO*08] ZHANG H., SHEFFER A., COHEN-OR D., ZHOU Q., VAN KAICK O., TAGLIASACCHI A.: Deformation-drive shape correspondence. *Proc. Eurographics Symp. on Geometry Processing 27*, 5 (2008), 1431–1439. 3

[ZST*10] ZHENG Q., SHARF A., TAGLIASACCHI A., CHEN B., ZHANG H., SHEFFER A., COHEN-OR D.: Consensus skeleton for non-rigid space-time registration. *CGF (EUROGRAPHICS) 29*, 2 (2010). 3