

Structure-Aware Shape Processing

Niloy J Mitra Michael Wand Hao Zhang Daniel Cohen-Or Martin Bokeloh
 UCL MPI SFU TAU Stanford Univ.

Abstract

Shape structure is about the arrangement and relations between shape parts. Structure-aware shape processing goes beyond local geometry and low level processing, and analyzes and processes shapes at a high level. It focuses more on the global inter and intra semantic relations among the parts of shape rather than on their local geometry. With recent developments in easy shape acquisition, access to vast repositories of 3D models, and simple-to-use desktop fabrication possibilities, the study of structure in shapes has become a central research topic in shape analysis, editing, and modeling. A whole new line of structure-aware shape processing algorithms has emerged that base their operation on an attempt to understand such structure in shapes. The algorithms broadly consist of two key phases: an analysis phase, which extracts structural information from input data; and a (smart) processing phase, which utilizes the extracted information for exploration, editing, and synthesis of novel shapes. In this survey paper, we organize, summarize, and present the key concepts and methodological approaches towards efficient structure-aware shape processing. We discuss common models of structure, their implementation in terms of mathematical formalism and algorithms, and explain the key principles in the context of a number of state-of-the-art approaches. Further, we attempt to list the key open problems and challenges, both at the technical and at the conceptual level, to make it easier for new researchers to better explore and contribute to this topic. Our goal is to both give the practitioner an overview of available structure-aware shape processing techniques, as well as identify future research questions in this important, emerging, and fascinating research area.

1. Introduction

For it is not a bundle of parts but an organization of parts, of parts in their relative arrangement, . . . the coordinated parts, now as related and fitted *to the end or function of the whole*, and now as related to or resulting *from the physical causes* inherent in the entire system of forces to which the whole has been exposed. Chapter XVI [Tho92].

Realistic modeling, rendering, and animation of physical and virtual shapes have matured significantly over the last few decades. Yet, the creation and subsequent modeling of three-dimensional shapes remains a tedious task which requires not only artistic and creative talent, but also significant technical skills. Much perfection witnessed in computer-generated feature films requires extensive manual processing and touchups. Every researcher working in graphics and related fields has likely experienced the difficulty of creating even a moderate-quality 3D model, whether based on a mental concept, a hand sketch, or inspirations from one or more photographs or existing 3D designs.

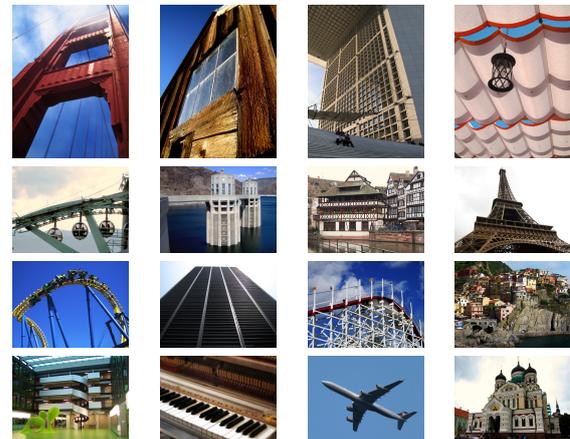


Figure 1: Structure in man-made objects arises from a multitude of factors, such as physical, aesthetical, or economical constraints. This paper surveys methods for structure-aware processing of 3D shapes.

This situation, frequently referred to as the *content creation bottleneck*, is arguably the major obstacle to making computer graphics as ubiquitous as it could be and it has recently become a dominant problem in graphics. Classical modeling techniques have primarily dealt with local or low-level geometric entities (e.g., points or triangles) and criteria (e.g., smoothness or detail preservation), lacking the freedom necessary to produce novel and creative content. In contrast, recent research effort has shifted its focus to modeling which works at a more global and higher-level characterization of 3D shapes, namely, at the *structural* level.

Shape structure is about the arrangement and relations between shape parts. *Structure-aware* shape modeling goes beyond local geometry refinement and detail-preserving shape deformation. It involves structure-preserving shape manipulation, as well as the generation of new shape structures. For example, new shapes can be produced by editing an input shape while maintaining structural relations such as symmetry. Novel shapes can be generated by assembling parts from shapes in an existing collection, while retaining the original structure. Such data-driven approaches create new shapes by *reusing* existing 3D models; they are further supported by the increasing volume of 3D shapes from on-line shape repositories. The major ensuing challenge is that structure-aware shape processing necessitates a proper *understanding* of the reused shapes at the structural level, bringing in the need for *semantic* or even *functional* analysis of shapes (see Figure 2).

Structures of real-world objects exhibit great variability and complexity (Figure 1). Even seemingly simple objects such as household items and furniture obey a multitude of complex relations. Such relations (or constraints) arise from various practical considerations, which can be categorized as: *semantic* considerations (e.g., table-tops are horizontal); *functional* considerations (e.g., chair legs support the seat and keep the chair stable); and *fabrication* or economic considerations (e.g., repeated object parts are easier to replicate by reusing machining or molding setups). Such considerations lead to characteristic object structures, as reflected by the geometric arrangements and constraints among the object parts. Violating such structural constraints (which we are all intuitively familiar with) during model creation leads to implausible or unnatural results. Further, as digital prototyping and physical fabrication becomes commonplace the synthesized objects have to conform to such constraints to be functional or also to be just physically realizable.

In classical shape modeling, the user remains in charge of manually recognizing and asserting any higher-level structural constraints, and resolving any dependencies. To reduce the burden on designers and artists, we need methods that automatically recognize structural properties and invariants of a shape or shape families, that understand their interdependencies, and that assist the user in creating structurally



Figure 2: A family of sofas of varying dimensions and functions reuses parts and their relations (IKEA sofa collection EKTORP). Although the models have very different geometric representations, at a high-level, they have similar style. We focus on extracting structure from low-level geometric descriptions to characterize such a notion of style leading towards a semantic understanding of shapes.

plausible shapes efficiently. Thus, the processing pipeline is necessarily structure-aware throughout.

In this survey, we review methods that support structure-aware shape processing. We focus in particular on techniques designed for *man-made* objects. These objects often have simple structural invariants across larger classes of shapes, such as symmetry, coplanarity, orthogonality, or regular arrangements, allowing us to characterize whole families of shapes concisely. Even though the low-level geometric descriptions of each individual object might be detailed and complex, the subparts of the objects share many such local and non-local relations across a larger family of related shapes. Thus, they can be compactly represented and parameterized as few part geometries along with the relationship among the parts, e.g., as a graph or hyper-graph.

A historical perspective. In the context of computer aided geometry, the use of algebraic methods to handle constraints was pioneered in the 1980-90s. We refer the readers to the survey by Huffmann et al. [HJA05] for key aspects of geometric constraint solving with characterization of the different methods used for constraint driven modeling. Instead, in this work we focus on the challenges arising from having to deal with real unorganized models, which naturally require geometric data understanding (e.g., 3D scans; non-manifold and heterogeneous quality 3D models).

Up until the mid-2000's, geometry modeling in graphics was dominated by techniques focusing on low-level shape analysis and processing. Typical examples include smoothing, subdivision, surface reconstruction, and Laplacian - based detail-preserving editing. A notable exception is the work of Funkhouser et al. [FKS*04] on modeling by example, where new shapes are generated by part composition based on a database of existing shapes (see Figure 3). This modeling paradigm still plays the dominant role in novel shape creation today, with the latest variants including structure-aware part shuffling [KJS07], probabilistic synthesis [CKGK11, KCKK12], set evolution [XZCOC12], etc.



Figure 3: Modeling-by-example proposed a novel model synthesis possibility by directly composing parts retrieved from a database of 3D shapes [FKS*04]. As 3D model repositories continue to grow, such a design possibility has increasingly gained research focus.

The analyze-and-edit approach introduced in the iWires work [GSMCO9] has set the tone for numerous works on structure-preserving editing, e.g., [ZFCO*11, LCOZ*11, BWSK12]. The analysis phase often takes the center stage as it presents the taller challenge of shape or structural understanding. Such works include inverse procedural modeling [SBM*10, BWS10] and symmetry-driven generative analysis [PMW*08, WXL*11]. Note early work by Stiny [SG71] on shape grammars has been an inspiration in this area, with current works focusing on recovering such grammars directly from the input examples.

Symmetry indeed plays a key role in structure-aware processing of man-made objects, since symmetries are abundant in these objects and from a functional point of view, symmetric structures are expected to share the same functionality. Methods for structural symmetry detection [MGP06, PSG*06, SKS06] have been made accessible to the geometric modeling community and since then, a large body of work has been developed on symmetry-driven shape processing (cf., survey [MPWC12]). An attempt to understand functionality of man-made objects was the notion of upright orientation [FCODS08], which can be seen as one of the early efforts in structural analysis of man-made objects. Both symmetry and upright orientation represent early attempts at extracting high-level semantic information from geometry.

Part analysis lies at the core of structure-aware shape processing. Extracting parts from a shape is the classical segmentation problem (c.f., survey [Sha08]). Equally important is the part correspondence problem, not only between a pair of shapes, but among a shape collection (c.f., survey [vKZHC011]). Early works on both the problems focused exclusively on local geometric analysis including those on geodesics and curvature. An influential effort was

made by the ShapeAnnotator framework [ARSF09] which helps the user create semantic, part-based shape ontology. Recent developments have taken the data-driven approach, in particular by learning structural invariants from a set of exemplars. Such a *co-analysis* approach started with the work of Golovinskiy and Funkhouser [GF09] on consistent segmentation and gained momentum on several fronts including style content separation [XLZ*10], joint segmentation [HKG11], spectral clustering [SvKK*11, KLM*12], active learning [WAvK*12], co-abstraction [YK12], and discovery of functional substructures [ZCM13]. Going beyond object modeling, Fisher et al. [FSH11, FRS*12] analyze scene data using spatial contexts and apply the results for novel scene synthesis.

Organization. We organize this survey on structure-aware shape processing techniques by the underlying structure models as primary sorting criterion. We believe that formalizing our understanding of “structure” in shapes is the intellectual key challenge in creating more powerful and generally applicable modeling tools. Beside discussing the broad types of structure analysis approaches, we also summarize key application areas including smart acquisition, structure-aware model editing, novel model synthesis, and design space exploration. We conclude by listing some of the key challenges to be solved and their potential implications.

2. Overview of Structure

Starting with early scientific efforts to relate object form to its intended functions, the inter-relation between form and function has played an important role in shape understanding and design. In nature, the geometric forms, such as the skeleton, of many organisms can be explained by physical considerations arising from their environment, operational efficiency, and functional utility. For example, the horns and shells of animals are often spiral-shaped due to growth pattern and rate considerations [Tho92]. Hence, it is expected that functionally related natural objects would share a similar form, or as we shall call in this survey, structure.

In man-made artifacts, such similarities can be even more pronounced since most man-made objects are designed to serve singular functions. Structures of man-made objects arise due to physical and economic, i.e., cost, constraints during design and manufacturing, or more pragmatically, due to semantic considerations. But, what is a *structure*?

The Oxford dictionary defines structure as

(Definition): The arrangement of and relations between the parts or elements of something complex.

In this view, structure constitutes a collection of parts and how the parts are mutually related. It should not be confused with the notion of structure in structural engineering. In this survey, we follow this abstract notion for composite objects. Such a characterization is not surprising given that

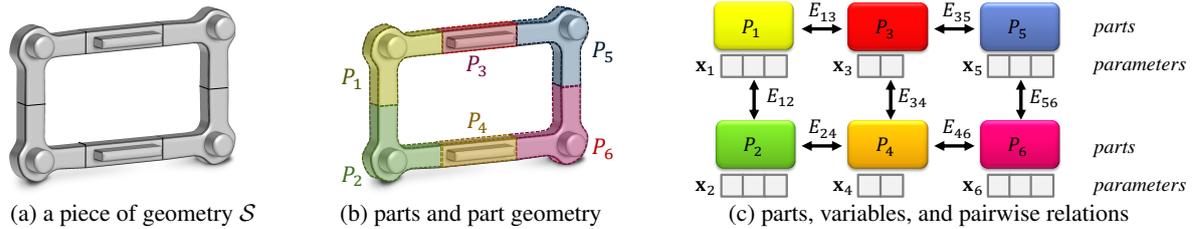


Figure 4: An example of a structured shape (a) that is decomposed into parts. Each part controls a portion of geometry (b), which we call part geometry. The parts (c) have parameters and a constraint energy that controls the parameters as well as the decomposition itself. The example shows a pairwise relations (a part graph). In this particular case, it has a Markovian structure (only neighbors interact). Such graphs would for example reflect the structure of enforcing connectivity or smoothness across neighboring pieces.

most man-made objects (e.g., chairs, tables, lamps, shelves) are a constellation of parts, where the constellations are often characteristic of the corresponding model collection. We assume that the global geometry is captured by a composition of (abstract) *parts*, each of which has *parameters* that define the parts, and a set of *relations* among these parts that impose structure on the composite. A unique feature of the setup is that the relations can arise among parts restricted to any particular model, or more generally, across different but functionally related models (e.g., a collection of chairs).

2.1. Modeling Structures

We describe a generic model for structuring shapes, which we later use to compare and contrast apparently different structure-aware shape processing techniques, and better understand their scope and design choices. A shape S can be seen as a collection of parts, their parameters, and most importantly, the relations that characterize the arrangement of the parts (see Figure 4).

Parts and parameters. A part P_i of a shape is a logical entity of semantic significance that controls the appearance of part geometry. Note that in this abstraction, parts are not necessarily disjoint, i.e., they can overlap. Further, each part has a finite set of parameters that affect the shape of the part. Note that unlike in traditional geometry processing, by part we do not necessarily mean a surface patch arising from segmentation. Instead, a part can simply be an abstraction for a region of the object and act as a proxy for a semantic part. Figure 4 shows a schematic example of a shape that is decomposed into parts with multiple parameters per part.

Generally speaking, a part a vector of all the vertices forming a shape space (e.g., as in [YYPM11]), a feature curve (e.g., as in [BBW*09, GSMCO09]), a bounding box (e.g., as in [XLZ*10, OLG11]), a fitted proxy such as a box or generalized cylinder (e.g., as in [ZFCO*11, XZZ*11, XZ-COC12]), a surface or volumetric segment [Sha08], or a variational proxy, etc. Note that in most cases the choice of parts automatically determines the choice of the respective

parameters. Further, in some cases, the parts can be completely specified by the user, i.e., semantic parts can be provided as part of the input as an input template. Most methods covered in this survey operate on parts which are meaningful components of a shape [Sha08], e.g., a leg of a chair, a table top, a wing of an airplane, or a window over a facade, etc.

Relations. Relations capture how parts, and hence their parameters, are correlated. Such relations can be between a pair of parts (i.e., a pairwise relation) or among a set of parts (i.e., higher order relations). The relations are the key element behind any structure. Formally, relations can be represented mathematically by a *constraint energy* E that must be zero for a valid structure (relations enforced as hard constraints) or that should be minimized (soft constraints). Let us consider few examples:

- In a constraint-based modeling setup, the relations would require elementary, pairwise relations such as parallelity, coplanarity, bilateral symmetry, etc.
- Relations can also link a set of parts by considering higher-order relations. For example, the windows on a building facade can be arranged on a 2-parameter regular grid. In general, symmetry relations with more than two elements involved will lead to higher-order relations. Figure 5 shows an example – the blue parts are coupled by a 4-ary symmetry energy.
- A set of relations can capture the part configuration of a functional model collection, e.g., a set of chairs. Such constraints should capture both coupled part-level continuous variations and also discrete changes like chairs with or without arms. Thus, the relations capture not only part-arrangements, but also how they can vary in a coupled fashion, i.e., a corresponding deformation model.

2.2. Discovering Structures

For most methods covered, the input shapes come in the form of low-level geometric representations, such as polygonal meshes, thus obscuring the underlying structure of the shapes. For example, a 3D scanner returns a collection of

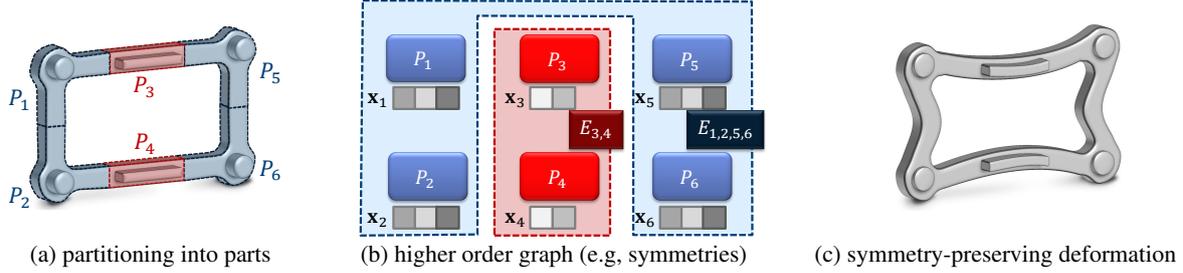


Figure 5: We can also use higher-order relations that couple larger cliques of parts (a), including non-local (non-Markovian) relations. In this example, we show a part (hyper-) graph (b) that aims at preserving symmetry. It couples symmetric pieces using hyper-edges (red/blue), leading to symmetric parameter changes. Such a setup can for example be used for symmetry preserving shape deformation (c).

points, possibly with normals; or, models from 3D model repositories (e.g., Trimble 3D warehouse, Turbosquid) often come as polygon meshes (manifold, or polygon soups), mostly without access to any semantic information. A key challenge is to discover the structure from a single model, or a set of related example shapes (i.e., a model collection). Thus, we have to determine (i) the parts, (ii) their parameters (i.e., representation), and most importantly, (iii) their relations. These are in fact mutually dependent, so ideally they should be learned in conjunction to each other.

A key aspect of structure-aware processing is to *discover* or learn the underlying structure from input examples — it is an analysis problem we refer to as structural analysis (Section 4). Starting from generic assumptions of how parts and relations are formed, the methods semi- or fully-automatically extract a structure model from one or a set of shapes. In Section 4.2, we categorize the main strategies used to discover structure models, which include symmetry-driven analysis, statistical modeling, and *co-analysis*, which learns a structure model from a shape collection.

2.3. Structure-aware Shape Processing

Shape or part structures, once extracted, can be used for a range of structure-aware shape processing tasks (Section 5). A recurring theme in these tasks is structure preservation, e.g., during shape editing and synthesis. The working hypothesis is that preserving structures also ensures plausibility of the manipulated or synthesized models. Complementary to local differential operators, structure models might also provide *non-local relations* and thus couples distant parts of the shapes. Interestingly, focusing on structure over low-level details allows us to couple models even with large geometric and topological variations. Thus, structure or part constellations become the key focus, with geometry playing only a secondary role. Preserving structures in course of scanning, editing, modeling, or even design exploration, then leads to meaningful and intuitive shape processing. The structures provide high-level handles for model manipula-

tion that often reflect object semantics, thus vastly simplifying the modeling interface.

3. Types of Structures

In order to discover a structure, one has to solve three problems: (i) identify the relevant parts which is akin to a segmentation problem; (ii) identify the relevant parameters or representations for the parts which is akin to part parameterization problem; and (iii) identify relations (i.e., correlation) among the parts both inside a shape and also across different shapes in a model collection. The challenge is perform these steps in conjunction.

We now categorize different approaches mainly based on how each of parts, their parameters, and their relations are discovered (see Figure 6). In each case, depending on the richness of the input data, broadly there are three options: (i) Models are manually prescribed by the user; (ii) models are extracted based on some prescribed model amounting to model-fitting; and (iii) models are learned directly from the data in an unsupervised or supervised setting.

3.1. Identifying Parts

The first task is to devise a scheme for splitting objects into parts. Ideally, such parts should relate to semantic object parts, rather than being constructed from low-level geometric details.

User specified parts. The simplest idea is to let the user explicitly create or specify parts in the input data sets. For example, in a classic modeling system, the user creates parts such as spheres, cubes, etc. To this end, many approaches use information such as connected components of a triangle mesh [JTRS12, ZCM13] or the scene-graph hierarchy provided in a database of 3D models [FSH11].

Parts are obtained according to a fixed model. In this case, a fixed segmentation model is constructed and applied. Popular choices are for example the detection of crease

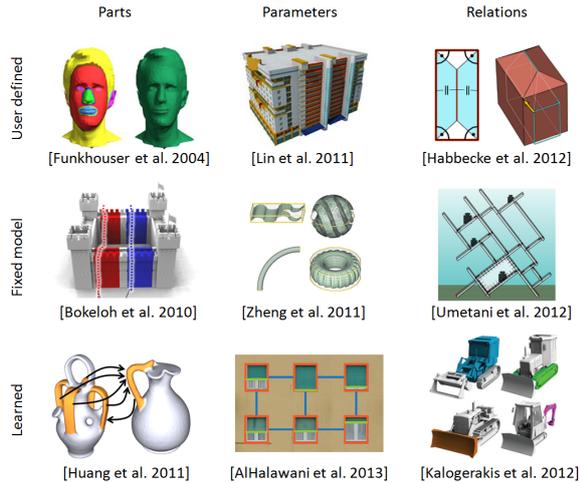


Figure 6: Examples of various methods that define or discover parts, parameters, and relations, i.e., extract structure for the respective collections.

lines [GSMCO09], of objects with a certain symmetry properties [MY^Y*10, MY^Y*13, BWS10, KBW*12], or generic shape segmentation approaches [Sha08].

Parts are discovered directly from the data. Instead of using a model that is completely based on a priori assumptions, it is also possible to define a meta-model that describes how the part structure should be extracted using a learning approach. The difference here is that the concrete extraction model will be adapted according to some set of training data. If this process requires user annotations, this is a supervised learning technique [ATC*05, vKTS*11, KHS10, SJWS13]. Unsupervised methods use clustering in order to find structural similarities in the input data, typically a collection of shapes belonging to the same family, automatically [GF09, XLZ*10, SvKK*11, HKG11, HFL12, KLM*12].

3.2. Identifying Parameters

Having established a part decomposition, the next task is to extract its parameters.

User specified parameters. This option is frequently used in procedural modeling schemes. By permitting the user to write scripts or shape grammars, custom parameters can be specified interactively. There are also more traditional systems (from the 1980s) where the whole modeling process is driven by specifying a scene hierarchy with user defined parameters used for communication between modules that instantiate geometry [SG71].

Parameters are obtained according to a fixed model. There are a large number of methods where the set of parameters is fixed a priori. For examples, parts are arranged using rigid transforms [FKS*04, GSMCO09], isometric variations [HMC09], or restricted based on fabrication require-

ments [YYPM11]. Very frequently, parts can be arranged with rigid motions or general homogeneous transformation matrices. Controlling parts with axis-aligned scaling of the bounding boxes is another option [ZFCO*11]. Parametric models also include specifying the geometric properties of fixed primitives (radius of a sphere, width/height/depth of a cuboid).

Parameters are learned directly from the data. In the most general setting, the set of parameters is not known a priori but the degrees of freedom have to be extracted from training geometry. This is more complicated and structurally very related to the extraction of global constraints from training data, which we discuss more in detail in the next subsection. Common models for learning parameter sets from data are dimensionality reduction techniques, such as principle component analysis [DHS00], or its non-linear variants (kernel PCA, spectral embedding techniques with custom kernels) [STC04].

3.3. Identifying Relations

The key aspect of structure is relation among parts. However, to relate parts often amounts to establishing a global correspondence among the parts. This is challenging as models often have significant geometric and topological variations even with the same model collection. In such scenarios, point-to-point correspondence is ill-defined. An alternative is to assign a probabilistic or fuzzy correspondence among shape collections [KLM*12]. Alternately, explicit correspondence extraction can be avoided as proposed by Ovsjanikov et al. [OLGM11] who analyze model shape collections as a manifold of characteristic descriptors. The descriptors are chosen such that they smoothly depend on the shape. Assuming continuous variability, exploring the shape space can be mapped to navigating a non-linear manifold.

Relations can capture how a part relates to itself (i.e., symmetry) and thus can be modeled as an unary term. More interestingly, pairwise part relations can be captured as binary relations; while more generally interaction between set of n parts can be captured as n -ary relations. Since n -ary relations can be difficult to track computationally, they are often split into lower-order relations, which is strongly related to graphical models in machine learning (c.f., [KF09]). Pairwise relations lead to graphs where nodes represent parts and their parameters and edges correspond to the constraint energy induced by pairwise relations. Higher order models correspond to hypergraphs, where each hyper-edge might connect more than two parts. If the graph is sparse and only connects locally neighboring parts, we have a Markov-random field (assuming a soft constraint energy that has a probabilistic interpretation, for example as the log-likelihood of an actual probability density). For the interpretation as a (classic) probabilistic graphical model, we have to assume that the set of employed parts is fixed a priori. The graphical model in Figure 4 is an example for a Markovian model with pairwise

edges. In contrast, Figure 5 is not a Markovian model, because there are non-local couplings. Furthermore, and independently of this fact, higher-order edges are used (the graph is a hypergraph).

User specified relations. In the context of computer-aided design and sketching interfaces, a number of systems have been developed where the user can specify general geometric constraints that are checked and maintained automatically by the modeling system [ZHH07, IMT99]. This is not the central topic of our survey; we will focus on automatic methods.

Relations are obtained by a priori model. In this case, the constraints are fixed, independent of the data. In many cases, these are physical constraints: For example, properties such as connectivity (dynamic stability), center of mass (for balance), or constraints on the maximum stress within the material (for static stability) can be imposed [UIM12, WSW*12]. Deformation models also fall into this category — they prescribe priors on differential properties of a deformation function acting upon a piece of geometry. These soft-constraints are given a priori by analogy to physical processes such as the elastic or plastic deformation of rods, shells, or solids.

Relations are discovered from the data. The final, most interesting, case is to learn constraints from data. This can be done with machine-learning approaches of different strength: A weakly learned model would employ a rather rigid a priori assumption and only apply it to example data in order to obtain concrete rules. An example of such an approach is the learning of shape grammars based on local similarity [MP08, BWS10]: Here, the invariant is fixed a priori (all shapes must be similar to an example shape within local regions). The learning is restricted to computing the rules that arise from applying this model to a concrete piece of example geometry. A much more flexible model is for example the famous morphable face model of Blanz et al. [BV99]: It only assumes low-dimensional linear subspaces for models and learns the geometric variability automatically from corresponding 3D scans. Such a model can also be extracted directly using abstracted part geometry and proxies [OLGM11, KCKK12].

4. Structure Analysis

In this section, we discuss recent techniques developed for learning structure models, which is an analysis task. The learned structures are then utilized for a variety of processing tasks, in particular, shape synthesis. Works on structure-aware processing are surveyed in Section 5.

4.1. Challenges in Structure Discovery

Although, in the previous sections, we independently considered parts, parameters, and their mutual relations, ideally they should be learned simultaneously. This makes the problem challenging, especially in the fully unsupervised setting.

For example, what are good object parts depends on their mutual relations, but the mutual relations can only be explicitly extracted when the parts along with their correspondences are known. In order to break this cyclic dependency, either training data is used, or the user prescribes candidate parts, or auxiliary model information (e.g., keywords, tags, scene-graphs) are usually used. Further, in many settings, parts can have significant variations across even related objects, and hence a corresponding deformation model has to be additionally extracted or factored out.

Instead of focusing on the variations (i.e., deformation), the general strategy is to look for consistency and patterns. This is motivated by statistical learning theory, which requires the repeated observation of objects originating from a common process. Only then is it possible to gather statistics about the outcome of the process and predict future events of this type. It is, in general, impossible to learn information from one process by observing a second, completely unrelated process [DHS00]. In terms of the part-based model defined above, any function of a subset of parts can be such a property if it (i) is observed repeatedly, and (ii) shows a significant statistical relation across these observations.

Thus, the definition of structural invariants requires identifying objects that correspond to the same underlying process. Please note that this abstract notion of correspondence does not necessarily refer to correspondence of points or regions in shapes. Correspondence is meant as an abstract association of properties that are related. For example, this could include generic properties such as elastic properties of local surface elements, or non-local properties such as the overall surface area of an object. Broadly, there are three main approaches, which we describe next.

(i) *Global point-wise correspondences:* These techniques consider a family of shapes with global, point-wise correspondences between the considered shapes. In our setting, the objects have only one part, with a number of corresponding parameters essentially representing a shape space (e.g., [KMP07]). If applied directly to geometry (for example, every vertex of a mesh being a parameter), this implies that the topology of the created or edited shapes cannot be changed from what was determined in the analysis phase. A classic example is a morphable model (providing a linear subspace of possible configurations) (e.g., [ACP03]). Physically based models such as elastic deformation also use global correspondences, but the constraint energy is not learned but rather specified a priori (e.g., [UIM12]).

(ii) *Partial point-wise correspondences:* These techniques are more general and do not require a one-to-one identification of all points in all shapes, but permit association across partial data. Hence, they correspond to techniques that permit multiple parts, where each part has its own parameter set. Only parts of compatible types, however, can be put into correspondences. Typical examples of part-based models are texture synthesis, (inverse) procedural models,



Figure 7: Models, even from the same family, can have large variations in geometry and topology. As a result, the notion of point-to-point correspondence is not always well defined in this context. Hence, Kim et al. [KLM*12] introduce the notion of fuzzy correspondence, which can then be used for interactive exploration of heterogeneous model collections.

or regularity-based shape resizing. Further, in the notion of model collections, Kim et al. [KLM*12] introduce the notion of fuzzy correspondence, where instead of a point-to-point correspondence a fuzzy or probabilistic notion of correspondence is proposed (see Figure 7).

(iii) *Correspondence across derived properties or “descriptors”*: These models do not identify actual surface points, but derived properties that summarize larger portions of the shape. Such models are typically *discriminative* [KH03]. They can detect changes of a property of a shape in a collection of shapes, but they are typically not *generative*; they cannot generate new shapes from scratch, just from knowing the derived property, because there is a loss of information in computing this property. Typical examples are methods that identify objects by descriptors, possibly combined with discriminative, supervised learning algorithms that structure the space of descriptors according to labeled examples. Such learning is for example performed in the data-driven shape segmentation approach by Kalogerakis et al. [KHS10].

4.2. Discovering Structures

We now survey works developed for structural analysis. There are different ways to classify these techniques. In terms of the learning mechanism, they can be supervised, unsupervised, or semi-supervised. In terms of what input a technique takes, they can be based on analyzing a single shape or studying a whole collection, i.e., co-analysis. From a computational point of view, we can examine the techniques by what optimization strategies are employed.

In what follows however, we categorize the surveyed methods in terms of the broad strategies employed in the analysis; a key question is what prior knowledge is relied upon. Supervised techniques rely on training data [KHS10]

or user-specified templates. We focus on unsupervised approaches.

4.2.1. Symmetry as Source of Structure

Symmetry is a purely geometric notion, i.e., it can be computed based purely on geometric analysis without any additional prior knowledge. Yet it carries powerful semantic information and plays a critical role in structural analysis of man-made objects, whether the input is an individual shape or a shape collection. In the case of analyzing individual models, symmetry provides the key source of structure. The most important reason is that symmetric parts in an object are expected, in fact, they were designed, to perform the same function. Hence, detected symmetries provide the first cues for functional analysis. Symmetry-driven analysis is primarily based on the prior knowledge that symmetric parts are to be grouped together [Wer23].

Symmetry is the absence of information (i.e., redundancy) within a shape. Typically, a group \mathcal{G} of allowed transformations are specified. Most often, \mathcal{G} is a subset of rigid, similarity, or isometric transforms. In order to have a symmetry, the group \mathcal{G} of operations must not change at least a subset $\mathcal{P} \subseteq \mathcal{S}$ of a geometric object \mathcal{S} . In this case, we have found a (partial) symmetry: All of the operations $g \in \mathcal{G}$ are symmetry transformations that keep \mathcal{P} invariant.

In general there are two different types of symmetries: The weak form are *instances* or *building blocks* [KBW*12]: The set of operations \mathcal{G} refers to all possibilities of exchanging parts of the object with one another (for example by rigid transformations) without affecting the shape itself (in this case, \mathcal{G} is not a subgroup of the rigid motions). The stronger form are *regular transformations* [LCT04, PMW*08]: In this case, the transformations acting on the geometry themselves form an algebraic group. This means, that not just the geometry is redundant in the sense of transformations of the same piece showing up repeatedly, but the way these building blocks are instantiated are regular themselves. Slippability [GG04] can be considered as a special case (continuous case) of such regularity with respect to rigid motions.

Various methods have been proposed for detecting symmetry [AMWW88, GCO06, LE06, MSHS06, MGP06, PSG*06, SKS06, OSG08, PMW*08, BBW*09, XZT*09, RBBK10, XZJ*12]. For a detailed survey and in-depth discussion of symmetry, see for example the recent surveys for images [LHOKG10] and for 3D geometry [MPWC12].

Symmetry is attractive as a structure model because it is agnostic of the actual geometry that shows a certain symmetry; it can therefore abstract from the concrete shape and can be used to formulate complex, high-level structure assumptions on larger classes of geometry. For example, symmetry has been used to complete partial shapes by inferring the most symmetric one that is consistent with the observation [TW05]. Mitra et al. [MGP07] have proposed a method to fully automatically make shapes more symmetric.

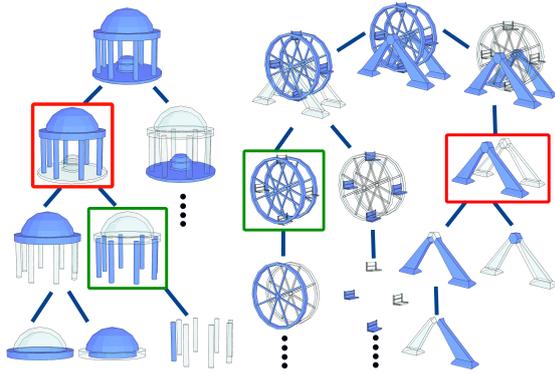


Figure 8: Symmetry hierarchy provides a structural organization of an object's parts. Parts associated with a node are colored blue. Each node represents either a grouping by symmetry (green) or assembly (red) by connectivity.

Symmetry can also be used as an invariant for controlling object deformation. The iWires system is based on the assumption that a free-form edit of an object should maintain the original symmetry properties of the input shape [GSMCO09] (Figure 14). The interaction and structure detection is based on salient line features [KMWJ96, OBS04], which are named 'wires' [SE98]. Similar thoughts have been used by Bokeloh et al. [BWKS11, BWSK12] and Xu et al. [XZZ*11], among others, in order to deform objects in a (symmetry) structure preserving manner.

Hierarchical nesting of symmetries has also been explored for shape representation and modeling [Mar07, WXL*11]. In particular, Wang et al. [WXL*11] constructs a symmetry hierarchy from a given man-made object (Figure 8). The object is assumed to have been segmented into meaningful parts. Inter-part symmetries as well as self-symmetries of the parts are first detected. Symmetry and (part) connectivity relations are encoded into a graph representation. Then recursive graph contraction is performed to form a hierarchical representation. The order of the contraction operations is determined by a set of hand-crafted rules which reflect the perceptual laws of grouping and the compactness principle (i.e., striving for the simplest representation). The resulting symmetry hierarchies are shown to reflect object semantics and support structure-aware hierarchical shape editing. However, consistency between the hierarchical structural representations obtained on a set of related shapes is not ensured.

4.2.2. Statistical Models of Structure

In the image domain, assembling images from local neighborhoods has been used in non-parametric texture synthesis. The underlying idea is to statistically model structure locally by assuming a Markov-random field (MRF). Instead of modeling global statistics over complete images only image pieces are considered and their potential connection to neighboring pieces. Non-parametric texture synthe-

sis was introduced by [EL99] and inspired further research for images [WL00, HJO*01, KSE*03, KEBK05, SCSIO8, WHZ*08]. Also, the idea has been tried for 3D geometry: [BIT04, SACO04, ZG04, LDD05, LHGM05, NYC05, ZG06, ZHW*06, Mer07, MM08, CM09, MM09, AZL12]. MRF-based approaches are powerful tools for modeling local texture but lack the ability to produce globally consistent results (e.g., closing loops). This is due to the MRF assumption that captures only local statistics while the global model structure is ignored. Assembling globally consistent pieces would result in a NP-hard optimization problem. Typically, non-parametric methods damp this problem by minimizing inconsistencies rather than avoiding them completely.

Kalogerakis and co-workers learn a generative probabilistic model describing a component-based structure from a pre-segmented set of shapes [KCKK12]. Their key observation is that structural variability heavily depends on the particular style of a shape and its components. Sailing ships, for example, typically differ vastly from cargo ships and components such as sails or containers naturally point to a particular style. This relationship is modeled with a probabilistic model consisting of observable random variables and latent variables that model the cause for these observations. The observable variables can be extracted from the input shapes such as the number of components, shape descriptors, and adjacency information (part parameters). The style of a shape and the style of sub components are modeled as latent variables. The model is organized hierarchically: The shape style conditions on the number of components and their style which in turn conditions on the observable variables describing a component. Additionally, lateral conditional dependencies are learned between observed random variables that capture relationships between attributes of different components.

A different approach is to model the global structure directly and thereby restrict the shape space to globally consistent shapes. In traditional procedural modeling the structure is encoded into a procedure that the user has to provide [SG71, PM01, WWSR03, MWH*06, LWW08]. Various methods were proposed to guide procedural modeling to achieve the anticipated result [TLL*11, BvMM11]. Inverse procedural modeling poses the problem of creating procedures from examples. In an early work [HF97] find fractal representations for input shapes. Recently, this line of research was picked up again for vector graphics [IOI06, IMIM08, HMC09, YM09, SBM*10] and shapes [ARB07, MP08, BWS10, BWKS11, BWSK12, VDA*12, TYK*12].

4.2.3. Co-analysis of Model Collections

In recent years there have been increasing interests in the co-analysis of sets of shapes. The premise is that understanding a shape, and in particular its parts, can only be reliable by observing a set of shapes possessing the same semantic or functionality, and not from a single observation. While this

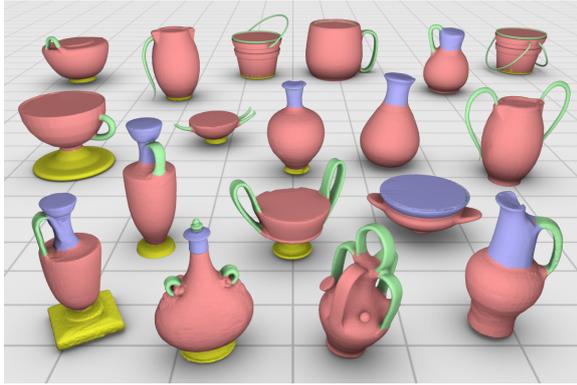


Figure 9: Unsupervised co-segmentation of a highly varied set of container objects [SvKK*11].

claim is straightforward, that more information can be extracted by simultaneously analyzing a set, it is unclear how to exploit the distributed knowledge of a set efficiently to achieve a coherent and reliable analysis of structures.

Unsupervised co-analysis only relies on the weak prior that the input set of shapes are related, that is, they belong to the same object class, e.g., chairs, airplanes, or lamps. The central problem of co-analysis is that of *co-segmentation*, where the challenge is to simultaneously segment all the shapes in the input set in a consistent manner (Figure 9). Thus, besides partitioning the shapes into segments, we also obtain a labeling of the segments across the set, where the parts with the same label serve the same semantic purpose, albeit possibly being geometrically dissimilar.

Existing attempts to co-segmentation can be broadly classified into supervised and unsupervised methods. In the supervised setting [KHS10, vKZHC011], a training set with enough pre-analyzed shapes is assumed to be given. The training set is then used to probabilistically label a set of unknown shapes. Although supervised methods are not strictly speaking a co-analysis (since the shapes are not simultaneously analyzed), the result of the labeling leads to a consistent segmentation for shapes in each object category.

The unsupervised setting is more challenging, since the entire knowledge must be extracted from the input set itself [GF09, XLZ*10, SvKK*11, HKG11, HFL12]. It should be noted that no method, be it supervised or unsupervised, can guarantee a perfect co-segmentation of a set (see Figure 9), since the geometry alone cannot always fully convey shape semantics. In particular, no descriptors can capture all possible geometric variations of a part.

Early work by Golovinskiy and Funkhouser [GF09] rigidly pre-aligns all the shapes in the set and then clusters the shape faces according to an underlying graph. The graph links faces that are adjacent in the models and faces that are close-by after the alignment. The resulting clusters provide a natural co-segmentation of the shapes.



Figure 10: Style-content separation [XLZ*10] by anisotropic part scales facilitates part correspondence among a diverse set of chairs (top). The separation is shown by the table with rows representing identified styles. The correspondences allow automatic synthesis of novel shapes (shaded in gold) from the example set via style transfer.

Xu et al. [XLZ*10] factor out the scale variation in the shape parts by first clustering the shapes into different styles, where style is defined by the anisotropic part scales of the shapes. In this manner, they are able to co-segment shapes with more variability, compared to [GF09], and also synthesize new shapes by transferring (the part proportion) styles, as shown in Figure 10. However, the technique is designed to only deal with one specific style.

Sidi et al. [SvKK*11] pose the co-segmentation problem as that of clustering segments resulting from an initial over-segmentation of the set of shapes using diffusion maps. Compared to the two previous works [GF09, XLZ*10], the clustering is performed in a space of shape descriptors rather than on the spatial coordinates of the shapes themselves. This allows the handling of corresponding parts differing in pose, location, and even cardinality, factors which would challenge any technique based on spatial alignment or direct clustering of shape geometry. In addition, the descriptor clustering approach allows to exploit a key enabling feature of the input set, namely, third-party connections. Even if two shapes possess parts that are significantly dissimilar, we can still establish a link between them if there are other parts in the set (third parties) that create such a connection, resulting in a successful co-segmentation. In a follow-up work, Hu et al. [HFL12] improve the feature modeling aspect of the algorithm. Instead of concatenating the different features into one feature descriptor, they pose and solve a subspace clustering problem in multiple feature spaces.

Huang et al. [HKG11] develop a joint segmentation method where a set is used to assist in the segmentation of *individual* shapes. They can handle rich shape variations by

using shape descriptors, but without aiming at a consistent segmentation of the entire set. Thus, strictly speaking, it is not a co-segmentation method.

Semi-supervised methods can be viewed as unsupervised methods assisted by input coming out of the set, typically through user interaction. In the recent work by Wang et al. on active learning for co-segmentation [WAVK*12], the system automatically suggests constraints which would most effectively refine the results while the user interactively adds constraints as appropriate. Specifically, the user adds either a cannot-link constraint or a must-link constraint between currently obtained segments. The constraints are propagated to the set and the co-segmentation is refined. Their work indeed demonstrates that relatively few user constraints can lead to almost perfect co-segmentation results. Their work also made available the COSEG dataset for evaluating algorithms on co-segmentation.

4.3. Structure from Physical Constraints

Some recent techniques analyze an input man-made object based on domain-specific priors arising from physical or manufacturing considerations. For example, physical laws (e.g., gravity, stability) can be used as algebraic constraints or relations to guide geometric optimization towards form-finding [UIM12, WSW*12]. Similarly, manufacturing constraints can determine structural preferences. For example, in the context of architectural buildings with glass facades [YYPM11], planar faces (e.g., quads) are more desirable as they are cheaper to build. A corresponding constraint amounts to planarity constraint for the respective faces of the surface mesh (e.g., for quad meshes). Other constraints can be horizontal floor lines, or equal height for different floors, etc. Finally, motion considerations for example in automata for toys, the respective parts (e.g., gears, links, etc.) should satisfy certain contact and kinematic constraints, again specified in the form of algebraic constraints [MY*10, ZXS*12].

5. Structure-aware Shape Processing

Sematic shape processing is a long-standing goal in model creation and synthesis. Structure, either prescribed as a part of the input or discovered by analyzing model collections, gets us closer to this goal. We defined structure as relations between parts (and their parameters) with the relations encoded as constraints or energy functions. As a result deformation, synthesis, and design exploration can all be cast as various forms of constrained-editing.

Optimization Strategies. Various optimization strategies have been employed for structural-aware processing as instances of constrained optimization: (i) greedy progressive projection to the multiple constraints [GSMCO09]; (ii) gradient descent [UIM12, WSW*12]; (iii) tangent space navigation using a level-set approach [YYPM11, BWSK12];

(iv) modal analysis by studying second order information of the constraints [YYPM11]; or sparse basis (e.g., using basis pursuit) to locally represent the solution space [HK12, DBD*13]. The main considerations in deciding among these various options are simplicity of implementation; speed and interactive-response; the order of the constraints; and the dimension of the shape space determined by the structure model.

5.1. Structure-aware Shape Editing

In this section, we discuss a few example systems that aim at *editing* existing shapes. All of these have in common that they base their structure model on an analysis of a single input shape. The discovered structure is then maintained as a soft or hard constraint. This could also be viewed as removing degrees of freedom in comparison to general, unconstrained shape modification, with the goal of making it easier to perform plausible changes.

We consider three classes of examples: (i) As a baseline, we consider traditional free-form deformation. (ii) We then discuss methods that extend these ideas towards the preservation of more general relations, such as salient shape features or symmetry. We pick the “iWires” system [GSMCO09] as a canonical example of such approaches. (iii) The last example concerns methods that are actually able to change the topology of the object. We discuss algebraic regularity models as one example of such methods [BWSK12].

5.1.1. Baseline: Free-form deformation

If we assume a triangle mesh as input, the simplest method for editing is certainly performing low-level mesh operations (moving vertices, possibly creating and deleting triangles). Obviously, this provides the most degrees of freedom and the least structural constraints. In practice, editing of shapes in this way is impossible for any but the simplest 3D models.

Instead of moving individual vertices, several methods have been proposed to perform higher-level edits that affect multiple vertices at the same time in a useful way. The first methods were free-form deformations [SP86, Coq90] that use a low-dimensional, band-limited, volumetric basis to impose smooth, low-frequency deformations to the geometry (such as a B-spline / Bezier basis of $3 \times 3 \times 3$ control points). The vertices $\mathbf{x} \in \mathbb{R}^3$ are deformed by a function $f: \mathbb{R}^3 \rightarrow \mathbb{R}^3$ that is composed out of such low-frequency basis functions $b_i: \mathbb{R}^3 \rightarrow \mathbb{R}^3$:

$$f(\mathbf{x}) = \sum_{i=1}^n \mathbf{x}_i b_i(\mathbf{x})$$

The underlying structure model is that high-frequency details of the geometry should be preserved while the low-frequency bands are subject to direct editing. According to our taxonomy, the parts in this approach are the scalar basis

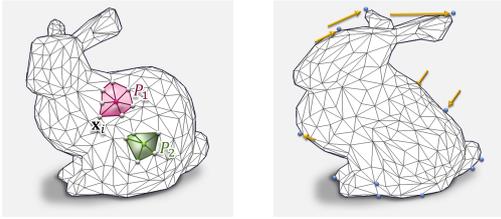


Figure 11: Left: Setting up a differential soft-constraint energy, every one-ring neighborhood is a part P_i , and every vertex is a parameter x_i , shared by multiple parts. From the configuration of vertices in the one-ring neighborhood, an estimate for differential surface properties is obtained. Right: Deformation result for an elastic deformation model.

functions b_i that represent localized, low-frequency degrees of freedom in the global deformation field f . The parameters are the coefficients x_i (which are 3D vectors). The part-geometry is the set of triangles that have vertices within the support of the part b_i . There is no constraint energy — the user has to adjust all parameters manually. Structure is only implicitly imposed by using only a few, low-frequency basis functions.

The requirement to adjust all control points of a low-frequency deformation basis explicitly creates some avoidable burden to the designer. More recent free-form deformation techniques use physics-based priors: The method uses an elastic energy as constraint energy. Additional handles are added to give the user control. In order to formulate elastic deformation models, differential properties of the deformed shape are computed and deformations that stretch or bend the surface are penalized using a soft constraint energy. If the input is a triangle mesh, the differential properties are typically read off a one-ring neighborhood in the mesh [SA07], so that the parts of such a model are all one ring neighborhoods and the parameters are the vertex positions of the mesh. Figure 11 illustrates this notion (left), along with a result from an elastic free-form deformation of the Stanford bunny model (right).

There are a large number of variants of this idea, dating back to the seminal paper of Terzopoulos et al. [TPBF87]. A survey is provided by Botsch and Sorkine [BS08]. From a birds-eye-view, the structure model always involves a constraint energy that penalizes deviations from the original shape. By formulating the penalty in a differential domain, local details are preserved more strongly than low-frequency shape properties.

5.1.2. Structure-Aware Deformation

Free-form deformations have a *local* and *non-adaptive* way of preserving structure: The shape of local pieces is preserved independently of the content (lack of adaptivity). Further, no global relations are considered; only a tendency towards low-frequency bending arises implicitly from chain-

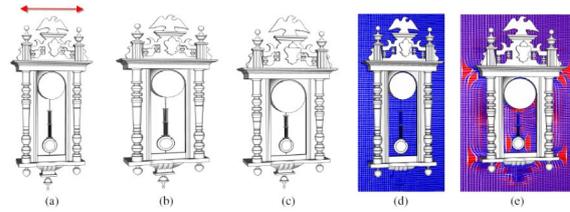


Figure 12: Non-homogeneous resizing protects salient regions [KSSCO08].

ing differential parts. Both of these aspects open up room for new methods.

Local adaptivity. Local but adaptive deformation is considered for example by [KSSCO08], see Figure 12. Their approach uses a differential free-form deformation energy that prefers axis-aligned stretch. Further, it estimates the “vulnerability” of local regions by looking at differential properties (curvature, slippage analysis [GG04]), and the elasticity of the model is adaptively reduced in vulnerable regions. Xu et al. [XWY*09] introduce adaptivity by adjusting the deformation penalties locally to match the slippability properties of the object (again based on slippage analysis); see Figure 13. This creates a deformation behavior that often mimics the behavior of mechanical systems (joints, cylinders etc.).

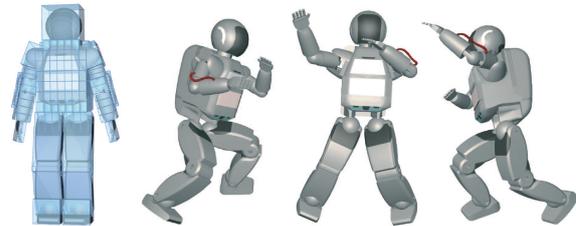


Figure 13: Possible joint locations are extracted by a local slippability analysis and subsequently used in enabling joint-aware deformation [XWY*09]. Note that structure in the form of the kinematic motion chains is discovered by analyzing the input model.

In both of these cases, the same concept of parts and parameters is used. The only difference is how the constraint energy is created. In both cases, the constraints only react to local differential properties of the deformation field, but the behavior is more adaptive than in a standard deformation approach.

Non-local relations. Global relations are used by iWires [GSMCO09], see Figure 14. The method first detects crease lines in a triangle mesh, which the authors call “wires”. These elements are the parts of the deformation model. The parameters are the vertices of the wires. Among these parts, all salient properties of Euclidean geometry (parallelity, orthogonality, different types of symmetry)



Figure 14: The *iWires* system models shape structure by symmetry relations within the object (specifically, among line features named “wires”). These properties are preserved during interaction [GSMCO09].

are enumerated in the analysis phase. These properties are then considered invariants of the deformation. Again, the user can control the deformation by handles, and at first a conventional elastic solution is computed. However, after that, the system tries to restore the invariants iteratively by a greedy algorithm that bends the wires such that they still meet the user constraints but better preserve the original structure. Parallel lines should remain parallel, and objects that were symmetric under a Euclidean (rigid) transformation should maintain this property.

A similar idea is examined by Zheng et al. [ZFCO*11], see Figure 15. Here, the parts are object-aligned bounding boxes of shape components (obtained from segmentation). Again, Euclidean invariants (symmetries) are used to propagate edits to affect all symmetric elements similarly.

Habbecke and Kobbelt [HK12] observe that fully constrained systems are of limited interest in the context of interactive modeling as they lack any design freedom. Hence they focus on under-constrained systems. They linearize the constraint functions (i.e., energies) and examine their nullspace (see also [BWSK12]) as an instance of the Cardinality Minimization Problem and efficiently solve the optimization using ideas from compressed sensing (see Figure 16). Unlike image-based constrained reconstruction approaches, their proposed system interactively supports manipulation even with hundreds of vertices and constraints.

5.1.3. Topological Changes

All of the methods so far apply homeomorphisms to the original shape, i.e., continuous, bijective mappings that preserve the topology of the input. More recently, structure-aware editing methods have appeared that are able to change the topology of the input.

The modeling by example system [FKS*04] introduced the concept of model synthesis as putting together parts from different models to form a novel and coherent shape (see Figure 3). The challenge, however, is how to define a family of shapes that is similar to the input but actually changes in composition. A line of work by Bokeloh et al. [BWKS11, BWSK12] uses the invariant of maintaining the regularity properties of the input shape. Regular patterns



Figure 15: Controllers (shown in cyan), either a cuboid or a generalized cylinder [ZFCO*11], serve as high-level deformation handles for structure-aware deformation of man-made shapes [XZZ*11]. The numbers indicate the number of controllers needed, showing that a small number of controllers usually suffice for the modeling tasks.

are a special kind of symmetry property: A part of the shape is repeated multiple times in a regular fashion, for example by a rigid transformation. Regularly repeating elements are important structural features of a shape; they can communicate direction, grouping, emphasis, layout, order, etc. Other important structural and visual properties of a shape such as straight lines, planes, cylinders, and spheres can be represented as continuous regular patterns. Using this regularity structure as invariant for shape deformation the resulting shape will resemble the original structure but with different proportions.

Both methods focus on discrete and continuous translational patterns but differ fundamentally in their representation. In [BWKS11] an elastic deformation field is augmented with anisotropic elasticity constraints penalizing deviation orthogonal to translational patterns while allowing stretch in pattern direction. Non-pattern regions are considered to be salient features and are assigned a high stretch resistance. Thereby, occurring stretch will be preferably diffused within regular patterns. Additionally, the method extracts elements from discrete 1-parameter groups and restructures the object by inserting or removing elements in order to minimize the distortion in pattern direction for discrete patterns. Similar anisotropic energies have been proposed in [KSSCO08] allowing uniform scaling but minimize non-uniform scaling

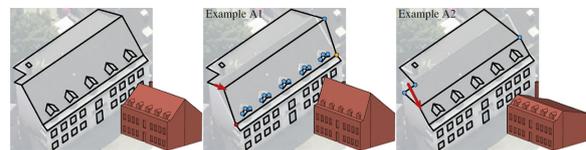


Figure 16: Modeling and editing a roof structure in a constrained-optimization setup. The vertices are relaxed in the analysis stage, and later automatically updated by the editing system [HK12].

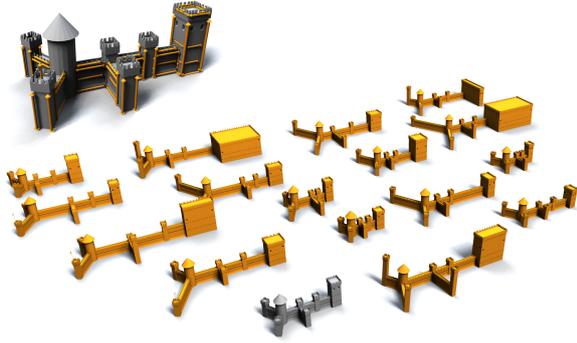


Figure 17: Regularity preserving shape editing by algebraically enforcing translative regularity patterns. Left: Degrees of freedom. Right: Shape variants [BWSK12].

in salient regions or in [XWY*09] using local slippage to simulate mechanical joints.

The model of [BWSK12] follows a stricter interpretation of maintaining regular patterns (see Figure 17). Instead of constraining an elastic deformation field in a least-squares sense to preserve regularity the method parameterizes shape deformation directly over regular patterns. Each pattern is represented by a small number of variables allowing only elementary changes such as moving the entire pattern or changing the number of elements. The method extracts all possible degrees of freedom by analyzing the nullspace of a linear system of hard constraints linking adjacent patterns and thereby guaranteeing a consistent shape. The resulting model typically offers a small number of changeable parameters that correspond to global changes in the shape and represent the algebraic structure w.r.t. regularity. Quadratic programming is used to combine the convex linear space of shapes with least-squares user constraints and regularization. The key difference to [BWKS11] is the inability to distort patterns at all. Users can only select valid shapes from the shape space which can be a great advantage in some applications but also be a disadvantage in others.

Formally, in [BWSK12], a part in their model is a region of geometry that has a fixed regularity pattern, i.e., (i) the same geometry shows up repeatedly, (ii) only differing by a rigid transformation, and (iii) these transformations form a commutative group (or a large enough excerpt of a commutative Euclidean symmetry group of rigid motions). By demanding maximal groups and area, the part decomposition is uniquely defined. Commutative symmetry groups are isomorphic to integer (or continuous) lattices; therefore, they can be characterized by simple repetition counts, which are the parameters of the model. The constraint energy then just aims at minimizing stretch to regularize the solution.

A different way of structure-aware deformation with topological changes was proposed by Lin et al. [LCOZ*11]. They explicitly deal with resizing of irregular architecture where the regularity-based approaches would not work. The

input mesh is decomposed into a hierarchy of parts (bounding boxes) manually defined by the user. Part parameters are also defined by the user and specify if a part must stay intact, can be replicated/deleted, or can be scaled. The method analyzes a compatibility graph between adjacent parts and constructs a series of 1D sequences which can be easily resized in a structure-aware manner and constrain the remaining sequences.

5.2. Structure-aware Shape Synthesis

In this section, we survey methods for structure-aware modeling that aim at creating new variants of shapes based on structural similarity to one input shape or a collection of shapes. Methods have been proposed that learn relations of part decompositions from a single exemplar using a fixed model to define parts like regular patterns [PMW*08, MP08, BWKS11, BWSK12] and/or partial symmetry [SBM*10, BWS10]. Other methods operate on segmented shape collections to learn part parameters and relations between them for a class of shapes [XLZ*10, KCKK12, XZCOC12].

5.2.1. Variation from Single Example

Inverse procedural modeling. Bokeloh et al. [BWS10] introduced inverse procedural modeling, wherein parameters of a procedural model is extracted starting from a single example (see Figure 18). Algorithmically, a shape grammar is constructed from a single input shape that produces r -similar shapes meaning that every r -neighborhood of geometry in the output maps exactly to an r -neighborhood in the input shape. The algorithm decomposes an input shape into building blocks at symmetric regions that allow docking of different building blocks at these cut lines forming so called *docking sites*. These docking sites can be systematically found by investigating partial symmetries of the input shape. A docking site follows the boundary of an r -symmetric region, thereby having a corresponding docking site due to symmetry properties, and separates the model into two disjoint pieces. Intuitively, the non-symmetric part can be replaced by the corresponding non-symmetric part enclosed by the corresponding docking site. The method samples symmetry transformations, computes all docking sites,



Figure 18: Inverse procedural modeling: Shape grammar computed from input shape (red) synthesizes variations of input (grey) [BWS10].



Figure 19: Structure-preserving retargetting of irregular 3D architectures [LCOZ*11]. The original input model is shown at the bottom left corner.

and combines them into a context-free shape grammar. Although the context-free grammar is actually a limitation, it does enable fast synthesis of similar models without requiring to recompute all available docking sites in each step. Subsequently, Kalojanov et al. [KBW*12] investigated the space of r -similar shapes further. Intuitively, one can overlay all docking sites and fracture the model simultaneously into micro-tiles. As it turns out, this is a canonical decomposition that can construct the entire space of r -similar models. Both methods are examples for fixed part parameters since every part remains rigid.

Earlier, Merrell [Mer07] used a fixed grid to decompose a model into cubic parameterless parts, analyzes geometric compatibility between these parts along the cut boundaries, and synthesizes new variants of the input shape by recombining pieces with compatible boundaries, similarly to texture synthesis. The method was further extended to handle more complex grid structures as in for example the Sierpinski Tetrahedron [MM08].

Structure retargetting. In the context of facade modeling, the work of Lin et al. [LCOZ*11] mentioned above performs retargetting of irregular 3D architectural structures including those of facades. Bao et al. [BSW12] generate multiple variations starting from a single facade design. An input facade design is generated by hierarchically segmenting and labeling an input image. The user can then manually specify constraints that should be maintained in any variation, e.g., alignment constraints between windows. Subsequently, facade variations are generated for different facade sizes, where multiple variations can be produced for a certain size reminiscent of traditional media re-targeting approaches. Computing such facade variations is based on in-



Figure 20: Tile synthesis by utilizing a probabilistic graphical model learned from examples [YBY*13].



Figure 21: A matrix of photo-inspired creations [XZZ*11] (rows: 3D model candidates; columns: photographic inspirations). Structures of the 3D candidates are preserved.

terleaving heuristic search with quadratic programming. Yeh and co-workers propose a method to learn neighborhood relationships of tiles in a regular grid [YBY*13] that allows the use of multi-tiles (larger regions spanning over multiple grid cells) and takes statistical properties into account that were observed in the example data (see Figure 20).

Photo-inspired synthesis. Xu et al. [XZZ*11] introduce an algorithm for the modeling of 3D man-made objects where the user draws creative inspiration from a single photograph. Model creation is supported by utilizing an available set of 3D candidate models. Specifically, the user creates a digital 3D model as a geometric variation from a 3D candidate; see Figure 21. The variation is obtained by automatically deforming a retrieved 3D candidate to fit the photographed target under the guidance of silhouette correspondence. The set of candidate models have been pre-analyzed to possess useful high-level structural information, which is preserved by the geometric variation so that the final product is coherent with its inherited structural information readily usable for subsequent model refinement or processing. On the downside, the modeling technique is unable to create new part structures. Or, structures based on abstracted geometry can be used to directly enable intuitive image manipulation of indoor scenes [ZCC*12].

5.2.2. Synthesis from Shape Collection

With style-content separation achieved via co-analysis of a set of related shapes, Xu et al. [XLZ*10] perform straightforward style transfer by altering the scales of the shape parts, filling in a style-content table with the transfer results. New variations are generated which possess varying part proportion styles, while keeping the part structures entirely intact; no new part structures are created.

Ovsjanikov et al. [OLGM11] implicitly extract such a



Figure 22: Evolution paths leading to the generation of a chair (bottom) with a new structure [XZCOC12]. Note that new topologies can be generated by the crossover operator. The ancestors of the final chair belong to sets of shapes that are evolved together, i.e., a set evolution.

morphable model by searching for a low degree of deformation to explain an input model collection using a template shape (as collection of bounding boxes) to specify a representation. Later, Kim et al. [KLM*12] introduce the notion of fuzzy or probabilistic correspondence to facilitate a faceted shape retrieval system.

In contrast, Xu et al. [XZCOC12] introduce a *set evolution* technique, called fit-and-diverse, to create non-trivial 3D model variations, starting from a set of structured models belonging to the same object class (see Figure 22). Instead of synthesizing new models one at a time, as in the classical modeling by example paradigm [FKS*04, KCKK12] (see Figures 3 and 23), an entire set of models are evolved to produce novel shapes. Their method utilizes concepts from evolution theory to spawn new models while adapting to user intent in an interactive modeling setting. Specifically, in each evolution step, the user selects a subset of shapes considered to be fit which produce offsprings for the next generation. Thus, user preferences define the fitness for the evolution so that over time, the shape population is steered to keep good fitness.

To avoid overfitting which would result in an elite population with low diversity, two mechanisms are used to increase the diversity. First, a novel crossover operator is introduced

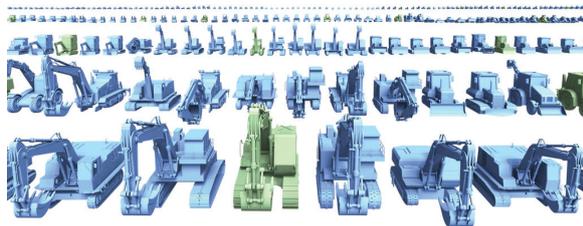


Figure 23: Recombination of existing shapes based on a probabilistic model (green=training data, blue=synthesized result) [KCKK12].

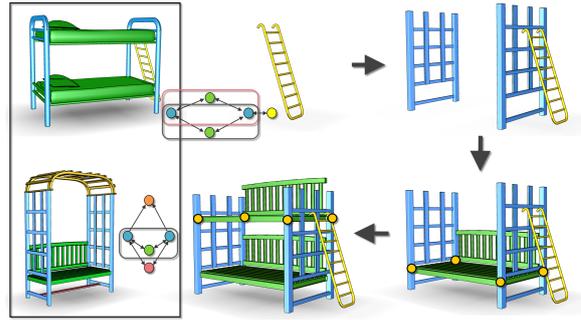


Figure 24: Parts from different models, even those from different model families, can still be combined based on the consistency of relations among the parts, i.e., matching sub-graphs in relation structures. This allows cross pollination of parts even across different model collections enabling functionally plausible model synthesis [ZCM13].

that shuffles parts from different shapes in the collection to create new mutations. Second, a small percentage of individuals with lower fitness score are added to the next generation. As a result, the evolution produces inspiring, and sometimes unexpected, shapes. Finally, the part crossover operator works at the fine level of shape parts and allows new topologies to be created while still preserving the coarse-level functional structure of the input shapes.

Jain et al. [JTRS12] present a system to interpolate and blend between shapes from a database. They treat shapes as a collection of parts, which are held together by symmetry and contact constraints. The system then synthesizes new models by combining parts with comparable symmetry and contact structures.

Zheng et al. [ZCM13] hypothesize that certain symmetric functional arrangements (i.e., part arrangements among symmetrically related parts) help capture object functions. Hence, preserving such arrangements in course of model synthesis, even when combining models from geometrically and topologically different sources (e.g., models from different collections) leads to functionally plausible models. Hence, they propose a purely geometric approach based on such substructures to match, replace, and position triplets of parts to create non-trivial, yet functionally plausible, model variations without requiring access to extensive training data (see Figure 24).

5.3. Structure-aware Shape Reconstruction

The GlobFit framework [LWC*11a] regularizes locally fitted primitives based on extracted global mutual relations. The main assumption is that man-made engineering objects primarily consist of basic primitives, possibly repeated and globally aligned under common relations. Hence, they introduce an algorithm to directly couple the local and global aspects of the problem. The fitted primitives become the parts,

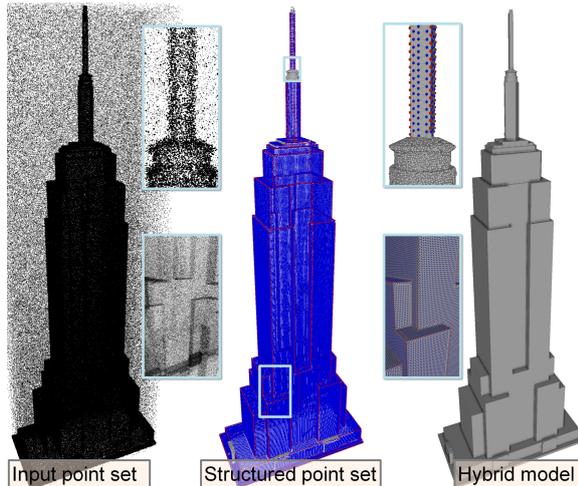


Figure 25: Point set structuring has been proposed for surface reconstruction [LA13] even from very noisy and sparse input pointsets by directly recovering local structure (c.f., [LWC* 11b]) before surface reconstruction.

while the relations are learned by testing for pairwise primitive relations. A set of feasible relations are extracted among the candidate relations, and then aligned to, while best fitting to the input data. Earlier, Li et al. [LZS* 11] extract structure (in the form of translational symmetry) from photographs to improve model reconstruction from sparse and noisy LiDAR scans.

More recently, Lafarge and Alliez [LA13] reconstruct surfaces from point sets via a structure-preserving approach. The key idea is to use local planar components to consolidate point sets before actual reconstruction. The final surface is then obtained via a novel graph-cut problem formulated on the 3D Delaunay triangulation of the structured point set. The approach gracefully combines detected primitives with free-form parts of the inferred shape to produce robust and high-quality final models (see Figure 25).



Figure 26: Context-based model search result using graph kernels [FRS* 12].



Figure 27: Structure-aware 3D model reconstruction from Kinect data. The reconstruction is based on retrieving and then assembling parts taken from multiple example shapes in a small shape repository, while preserving the built-in structures of the examples [SFCH12].

The notion of context as structure information among co-occurring models was introduced in recent efforts by Fisher et al. [FSH11, FRS* 12], see also Figure 26. Structure has also been used recently for semantic model reconstruction from sparse, noisy, and often incomplete depth scans (e.g., Kinect scans). For example, Shen et al. [SFCH12] convert low-quality scans to high-quality 3D models using semantic labeled datasets (see Figure 27). The algorithm runs bottom up and creates new structures by assembling existing labeled parts. Essentially the segmented input models become the parts, while the database suggests how they can be structurally linked. Alignment is performed to align the potential part arrangements to best explain the acquired data. In a parallel effort, Kim et al. [KMYG12] propose a two phase approach: (i) a learning phase wherein they acquire 3D models of frequently occurring objects and capture their variability modes from multiple scans, and (ii) a recognition phase wherein from a single scan of a new environment, they identify previously seen objects but in novel poses and locations to support recognition at interactive rates.

5.4. Structure-aware Design Exploration

In the context of design exploration, a direct optimization approach (e.g., gradient descent) is rarely desirable. For example, designers are more interested in exploring the space of acceptable solutions, rather than being presented with a single (best) solution. Thus, the goal is to expose the space of acceptable solutions as characterized by the solutions that satisfy the structure relations (or constraints) within an allowed threshold margin.

Yang et al. [YYP11] introduce the notion of *constrained meshes*, where the input is a single input mesh along with a set of non-linear constraints (in terms of the mesh vertices). The goal is to explore other meshes with the same connectivity, while respecting the prescribed constraints. They model



Figure 28: Constrained meshes, i.e., the shape space of meshes with a set of prescribed constraints involving the free vertices, preserve non-local part relations while allowing the users to directly navigate in the implicitly defined constrained shape space [YYPM11].

the problem by mapping the meshes to points $\mathbf{x} \in \mathbb{R}^D$, where D is 3 times the number of deformable vertices. Each constraint defines a hypersurface in \mathbb{R}^D , and the intersection of all these hypersurfaces forms the corresponding shape space, or mesh manifold, \mathcal{M} . Thus, each energy E_i manifests as such a hypersurface. For example, face planarity leads to the planar quad (PQ) mesh manifold. They locally navigate in \mathcal{M} (actually within a tolerance band around \mathcal{M}) with the help of local approximations of \mathcal{M} . These are tangent spaces and quadratically parameterized surfaces having second order contact with \mathcal{M} . They observe that in many settings additional quality measures (e.g., aesthetics) are available to discriminate among the valid solutions. Thus, while any point of the shape space \mathcal{M} represents a valid constrained mesh, only certain parts of \mathcal{M} are desirable according to application specific quality measures, e.g., fairness of selected mesh polygons. They enable such an exploration with the help of appropriate energy functions and their second order approximations that are intrinsic to \mathcal{M} . Eigen-analysis of the intrinsic Hessians of energy functions turns out to be a highly effective tool for the identification of the good parts of the shape space for subsequent exploration with applications in design exploration, optimization, and handle driven deformation with boundary conditions, while conforming to prescribed constraints (see Figure 28). The setup has subsequently been extended to handle curve-based model deformations [ZTY*12], and also deformations restricted to local modifications based on sparsity considerations [DBD*13].

Umetani et al. [UIM12] propose a guided modeling interface for exploration of valid shapes in the context of wood plank-based furniture. They propose an interactive design framework for efficient and intuitive exploration of geometrically and physically valid shapes. During geometric editing operation, the proposed system continuously visualizes the valid range of the parameter being edited. The planks (i.e., cuboids) become the parts, while the structures arise from the user-specified part layout. During manipulation, as one or more constraints (e.g., stability or toppling) are violated, the system generates multiple suggestions involving both

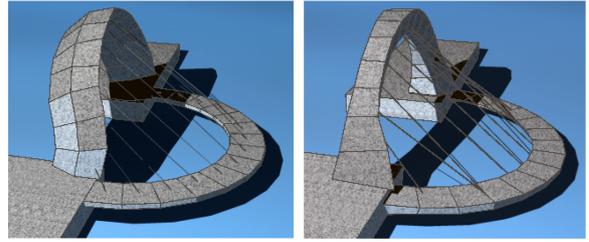


Figure 29: Improving the stability of a building using structural optimization, thus enabling a novel form-finding possibility [WSW*12].

discrete and continuous changes to restore validity. Thus, while the user focuses on the aesthetic aspects of the design, their computational framework helps to achieve physical realizability by providing active guidance to the user leading to fabrication- and physically-based form-finding possibility for conceptual design.

Whiting et al. [WSW*12] also use structural constraints to guide form-finding for shapes, especially to masonry structures. They explore how geometric changes can lead to improved structural stability. The approach is based on a novel measure of structural soundness, which is then optimized by taking derivatives with respect to changing geometry. Specifically, geometric form is refined via a gradient descent while introducing other constraints like volume preservation and uniform thickness (see Figure 29).

6. Conclusion

Inter-part relations within single shapes and across multiple shapes constitute structures of objects. In the context of man-made objects such structures capture characteristic non-local part relations and provide high-level information. Such part configurations along with their allowable variations arise out of various considerations: semantic factors, physical realizability, or fabrication convenience. Further, preserving such structures in course of any geometric manipulation help maintain the plausibility of the final geometric form. In recent years, there has been a flurry of research efforts to discover such structures from models, and subsequently preserve them in course of subsequent model manipulation. In this survey, we organize the various efforts in this emerging topic of structure-aware shape processing particularly focusing on extraction, representation, storage, and processing of structures in man-made shapes, with applications to various tasks including semantic reconstruction, scene understanding, intuitive deformation, plausible synthesis, and generally to design space exploration.

Future Challenges

We still consider structure-aware shape processing as a relatively young sub-field of computer graphics with many open

issues and problems to address. We present a list of important open problem (in our opinion) to conclude this survey.

Hierarchical models. Most existing structure models represent flat arraignments of shape parts or features without a hierarchical organization, except for symmetry hierarchies [Mar07, WXL*11]. However, symmetry hierarchies have so far been constructed by analyzing input shapes in isolation and the results do not ensure any consistency over a set of related shapes, limiting their utility. Since the way we perceive complex structures is generally thought of as being hierarchical, studies of hierarchical structure models for shape processing deserve more effort.

Functional shape analysis. The shift from low-level geometry processing to high-level structure-aware shape processing is posing new questions. Instead of focusing on measures that tell how similar two surface regions are geometrically, we are now interested in what makes two substructures functionally similar. Instead of producing shape variations which preserve local geometry details, we are now interested in how to maintain the same functionality during shape editing or creation. We expect functional shape analysis, i.e., how to properly map geometry to functionality, to draw more attention in research on geometry processing.

Scalability. To date, most works on structure-aware shape processing have only been demonstrated and evaluated on small to medium-sized datasets over a handful of object categories. Popular object classes include those of chairs, airplanes, lamps, and other household items. 3D scene models typically contain up to twenty items. Hence, both in terms of data size and structural complexity, the scalability of the proposed analysis and modeling techniques has not been sufficiently investigated. In addition to scalability analysis, it is also clearly beneficial to develop large-scale benchmarks for various structure-aware processing tasks.

The data challenge. Much work covered in this survey is data-driven, where existing 3D shapes, possibility pre-analyzed or manually labeled, serve as either training data or examples for analysis or modeling. However, such approaches are always limited by the scant availability of quality 3D shapes and quality 3D shape collections. While the issue may resolve by itself as the number of digital 3D models continue to grow, one may look for alternative rich data sources. One possibility is to tap into the massive amount of image data, however, how image data can be utilized for structure-aware processing of 3D shapes is the challenge.

Unified structural representation. We have discussed a number of structure models in use so far and most of them were developed for specific applications. For example, in iWires [GSMCO09], the model was built on feature curves and their relations and this was later extended for part-based analysis and manipulation via controllers [ZFCO*11]. Data-driven modeling through reuse of existing shapes and their structures will be greatly facilitated if there were a unified

structural representation. Such a representation requires the necessary formalism, compactness, and generality.

Conflicting constraints. As economical and accessible 3D data acquisition technologies, such as Kinect@home and LiDAR scans, propel the growth of 3D model collections, the quality of input data may become less unreliable and the structures extracted from such data may contain an increasing level of noise in the form of outliers or conflicting relations. Extracting a consistent set of relations from such inconsistent and conflicting relation sets will be another interesting and challenging research direction to pursue.

Author Background

Niloy J. Mitra is an associate professor (Reader) in the Department of Computer Science at University College London (UCL). Earlier, he co-founded the Geometric Modeling and Visualization Center (GMSV) center at KAUST. He received his Ph.D. degree and Masters in Electrical Engineering from Stanford University. His research interests include shape understanding, fabrication-aware design, geometric modeling, recreational art, and computer graphics in general. His work spans the area of symmetry detection, shape analysis, and their applications to semantic object manipulations. Currently he serves on the editorial boards of Transaction on Graphics, Computer Graphics Forum, Visual Computer, and Computer & Graphics.

Michael Wand is a senior scientist and junior research group leader at the Max-Planck Institut Informatik and Saarland University. He received his Ph.D. degree from Tübingen University in 2004 and his Diploma in Computer Science from Paderborn University in 2000. From 2005 to 2007, he was a postdoc at Stanford University. His research interests include statistical techniques for geometry processing and geometric correspondence problems.

Hao (Richard) Zhang directs the Graphics, Usability, and Visualization Lab at Simon Fraser University, Canada, where he is an associate professor in the School of Computing Science. He received his Ph.D. from the Dynamic Graphics Project (DGP), Department of Computer Science, University of Toronto in 2003 and M.Math. and B.Math. degrees from the University of Waterloo. His research interests include geometry processing, shape analysis, and computer graphics. Currently, he serves on the editorial boards of Computer Graphics Forum and Graphical Models. He is a co-chair for SGP 2013. He was a winner of the Best Paper Award from SGP 2008 and the Most Cited Paper Award for the journal Computer-Aided Design in 2010.

Daniel Cohen-Or is a professor at the School of Computer Science. He received his B.Sc. cum laude in both mathematics and computer science (1985), an M.Sc. cum laude in computer science (1986) from Ben-Gurion University, and his Ph.D. degree from the Department of Computer Science (1991) at State University of New York at Stony Brook.

He received the 2005 Eurographics Outstanding Technical Contributions Award. His research interests are in computer graphics, in particular, rendering and modeling techniques. His main interest right now is in few areas: image synthesis, motion and transformations, shapes and surfaces, and surface reconstruction.

Martin Bokeloh is a postdoc at Stanford University and part of the Max Planck Center for Visual Computing and Communication. He received his Ph.D. degree from Saarland University in 2011 and his Diploma in Computer Science from University of Tübingen in 2007. His research interests include geometry processing, shape understanding, and inverse procedural modeling.

References

- [ACP03] ALLEN B., CURLESS B., POPOVIĆ Z.: The space of human body shapes: reconstruction and parameterization from range scans. *ACM Trans. Graph.* 22, 3 (2003), 587–594.
- [AMWW88] ALT H., MEHLHORN K., WAGENER H., WELZL E.: Congruence, similarity and symmetries of geometric objects. *Discrete Comput. Geom.* 3, 3 (1988), 237–256.
- [ARB07] ALIAGA D., ROSEN P., BEKINS D.: Style grammars for interactive visualization of architecture. *IEEE Trans. Vis. Comp. Graph.* 13, 4 (2007), 786–797.
- [ARSF09] ATTENE M., ROBBIANO F., SPAGNUOLO M., FALCIDIENO B.: Characterization of 3d shape parts for semantic annotation. *Computer Aided Design* 41, 10 (2009), 756–763.
- [ATC*05] ANGUELOV D., TASKAR B., CHATALBASHEV V., KOLLER D., GUPTA D., HEITZ G., NG A.: Discriminative learning of markov random fields for segmentation of 3d scan data. In *CVPR* (2005).
- [AZL12] ALHASHIM I., ZHANG H., LIU L.: Detail-replicating shape stretching. *The Visual Computer* 28, 12 (2012), 1153–1166.
- [BBW*09] BOKELOH M., BERNER A., WAND M., SEIDEL H.-P., SCHILLING A.: Symmetry detection using line features. *Computer Graphics Forum* 28, 2 (2009), 697–706.
- [BIT04] BHAT P., INGRAM S., TURK G.: Geometric texture synthesis by example. In *Symp. Geometry Processing* (2004), pp. 41–44.
- [BS08] BOTSCH M., SORKINE O.: On linear variational surface deformation methods. *IEEE Transactions on Visualization and Computer Graphics* 14, 1 (2008), 213–230.
- [BSW12] BAO F., SCHWARZ M., WONKA P.: Procedural facade variations from a single layout. *ACM Trans. Graph.* (2012).
- [BV99] BLANZ V., VETTER T.: A morphable model for the synthesis of 3d faces. In *Computer Graphics (Proc. SIGGRAPH)* (1999), pp. 187–194.
- [BvMM11] BENEŠ B., ŠŤAVA O., MĚCH R., MILLER G.: Guided Procedural Modeling. In *Comp. Graph. Forum* (2011), vol. 30, pp. 325–334.
- [BWKs11] BOKELOH M., WAND M., KOLTUN V., SEIDEL H.-P.: Pattern-aware shape deformation using sliding dockers. *ACM Trans. Graph.* 30 (Dec. 2011), 123:1–123:10.
- [BWS10] BOKELOH M., WAND M., SEIDEL H.-P.: A connection between partial symmetry and inverse procedural modeling. *ACM Trans. Graph.* 29 (July 2010), 104:1–104:10.
- [BWSK12] BOKELOH M., WAND M., SEIDEL H.-P., KOLTUN V.: An algebraic model for parameterized shape editing. *ACM Transactions on Graphics* 31, 4 (2012), 78:1–78:10.
- [CKGK11] CHAUDHURI S., KALOGERAKIS E., GUIBAS L., KOLTUN V.: Probabilistic reasoning for assembly-based 3D modeling. *ACM Transactions on Graphics (Proc. SIGGRAPH)* 30, 4 (2011), 35:1–35:10.
- [CM09] CHEN L., MENG X.: Anisotropic resizing of model with geometric textures. In *Conf. on Geometric and Physical Modeling (SPM)* (New York, NY, USA, 2009), ACM, pp. 289–294.
- [Coq90] COQUILLART S.: Extended free-form deformation: a sculpturing tool for 3d geometric modeling. In *Proc. Siggraph* (1990).
- [DBD*13] DENG B., BOUAZIZ S., DEUSS M., ZHANG J., SCHWARTZBURG Y., PAULY M.: Exploring local modifications for constrained meshes. *Computer Graphics Forum (Proceedings of Eurographics 2013)* 32, 2 (2013). to appear.
- [DHS00] DUDA R. O., HART P. E., STORK D. G.: *Pattern Classification (2nd Edition)*. Wiley-Interscience, 2000.
- [EL99] EFROS A. A., LEUNG T. K.: Texture synthesis by non-parametric sampling. In *Proc. Int. Conf. Comp. Vision* (1999).
- [FCODS08] FU H., COHEN-OR D., DROR G., SHEFFER A.: Upright orientation of man-made objects. *ACM Trans. Graph.* 27, 3 (Aug. 2008), 42:1–42:7.
- [FKS*04] FUNKHOUSER T., KAZHDAN M., SHILANE P., MIN P., KIEFER W., TAL A., RUSINKIEWICZ S., DOBKIN D.: Modeling by example. *ACM Trans. Graph.* 23, 3 (2004).
- [FRS*12] FISHER M., RITCHIE D., SAVVA M., FUNKHOUSER T., HANRAHAN P.: Example-based synthesis of 3d object arrangements. *ACM Transactions on Graphics (SIGGRAPH Asia)* 31, 6 (2012), 135:1–135:11.
- [FSH11] FISHER M., SAVVA M., HANRAHAN P.: Characterizing structural relationships in scenes using graph kernels. *ACM Trans. Graph.* 30, 4 (2011), 34:1–34:12.
- [GCO06] GAL R., COHEN-OR D.: Salient geometric features for partial shape matching and similarity. *ACM Trans. Graph.* 25, 1 (2006), 130–150.
- [GF09] GOLOVINSKIY A., FUNKHOUSER T.: Consistent segmentation of 3D models. *Computers & Graphics (Proc. of SMI)* 33, 3 (2009), 262–269.
- [GG04] GELFAND N., GUIBAS L.: Shape segmentation using local slippage analysis. In *Proc. Symp. Geometry Processing* (2004), pp. 214–223.
- [GSMCO09] GAL R., SORKINE O., MITRA N., COHEN-OR D.: iWires: An Analyze-and-Edit Approach to Shape Manipulation. *ACM Trans. Graph.* 28, 3 (2009), 33:1–33:10.
- [HF97] HART J., FLYNN O. C. P.: Similarity hashing: A computer vision solution to the inverse problem of linear fractals. *Fractals* 5 (1997), 35–50.
- [HFL12] HU R., FAN L., LIU L.: Co-segmentation of 3d shapes via subspace clustering. *Computer Graphics Forum (Proceedings of SGP)* 31, 5 (2012), 1703–1713.
- [HJA05] HOFFMANN C. M., JOAN-ARINYO R.: A brief on constraint solving. *Computer-Aided Design and Applications* 2, 5 (2005), 655–663.
- [HJO*01] HERTZMANN A., JACOBS C. E., OLIVER N., CURLESS B., SALESIN D. H.: Image analogies. In *Proc. Siggraph* 2001 (2001), pp. 327–340.
- [HK12] HABBECKE M., KOBELT L.: Linear analysis of nonlinear constraints for interactive geometric modeling. *Computer Graphics Forum* 31, 2 (2012).

- [HKG11] HUANG Q., KOLTUN V., GUIBAS L.: Joint shape segmentation with linear programming. *ACM Trans. Graph.* 30 (2011), 125:1–125:12.
- [HMC09] HUANG Q., MECH R., CARR N.: Optimizing structure preserving embedded deformation for resizing images and vector art. vol. 28, pp. 1887–1896.
- [IMIM08] IJIRI T., MÈCH R., IGARASHI T., MILLER G.: An example-based procedural system for element arrangement. *Computer Graphics Forum* 27, 3 (2008), 429–436.
- [IMT99] IGARASHI T., MATSUOKA S., TANAKA H.: Teddy: A sketching interface for 3d freeform design. In *Computer Graphics (Proc. ACM SIGGRAPH)* (1999), pp. 409–416.
- [IOI06] IJIRI T., OWADA S., IGARASHI T.: The sketch I-system: Global control of tree modeling using free-form strokes. In *Smart Graphics* (2006), pp. 138–146.
- [JTRS12] JAIN A., THORMÄHLEN T., RITSCHER T., SEIDEL H.-P.: Exploring shape variations by 3d-model decomposition and part-based recombination. *Comp. Graph. Forum (Proc. Eurographics 2012)* 31, 2 (2012), 631–640.
- [KBW*12] KALOJANOV J., BOKELOH M., WAND M., GUIBAS L., SEIDEL H.-P., SLUSALLEK P.: Microtiles: Extracting building blocks from correspondences. vol. 31, John Wiley & Sons, Inc., pp. 1597–1606.
- [KCKK12] KALOGERAKIS E., CHAUDHURI S., KOLLER D., KOLTUN V.: A probabilistic model for component-based shape synthesis. *ACM Transactions on Graphics* 31, 4 (2012), 55:1–55:11.
- [KEBK05] KWATRA V., ESSA I., BOBICK A., KWATRA N.: Texture optimization for example-based synthesis. *ACM Trans. Graph.* 24, 3 (2005), 795–802.
- [KF09] KOLLER D., FRIEDMAN N.: *Probabilistic Graphical Models*. MIT Press, 2009.
- [KH03] KUMAR S., HEBERT M.: Discriminative random fields: A discriminative framework for contextual interaction in classification. In *ICCV* (2003).
- [KHS10] KALOGERAKIS E., HERTZMANN A., SINGH K.: Learning 3d mesh segmentation and labeling. *ACM Trans. Graph.* 29, 3 (2010).
- [KJS07] KRAEVOY V., JULIUS D., SHEFFER A.: Shuffler: Modeling with interchangeable parts. In *Pacific Graphics 2007* (2007).
- [KLM*12] KIM V. G., LI W., MITRA N., DIVERDI S., FUNKHOUSER T.: Exploring collections of 3d models using fuzzy correspondences. *Trans. on Graphics (Proc. of SIGGRAPH 2012)* 31, 4 (2012), 54:1–54:11.
- [KMP07] KILIAN M., MITRA N. J., POTTMANN H.: Geometric modeling in shape space. *ACM Transactions on Graphics (SIGGRAPH)* 26, 3 (2007), #64, 1–8.
- [KMWJ96] KENT J. T., MARDIA K. V., WEST J. M., JT L. L.: Ridge curves and shape analysis. In *In The British Machine Vision Conference* (1996), pp. 43–52.
- [KMYG12] KIM Y. M., MITRA N. J., YAN D.-M., GUIBAS L.: Acquiring 3d indoor environments with variability and repetition. *ACM Transactions on Graphics (SIGGRAPH Asia 31)*, 6 (2012), 138:1–138:11.
- [KSE*03] KWATRA V., SCHÖDL A., ESSA I., TURK G., BOBICK A.: Graphcut textures: image and video synthesis using graph cuts. *ACM Trans. Graph.* 22, 3 (2003), 277–286.
- [KSSCO08] KRAEVOY V., SHEFFER A., SHAMIR A., COHEN-OR D.: Non-homogeneous resizing of complex models. *ACM Trans. Graph.* 27, 5 (2008), 1–9.
- [LA13] LAFARGE F., ALLIEZ P.: Surface reconstruction through point set structuring. In *Eurographics* (2013). to appear.
- [LCOZ*11] LIN J., COHEN-OR D., ZHANG H. R., LIANG C., SHARF A., DEUSSEN O., CHEN B.: Structure-preserving retargeting of irregular 3d architecture. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2011)* 30, 6 (dec 2011), 183:1–183:10.
- [LCT04] LIU Y., COLLINS R., TSIN Y.: A computational model for periodic pattern perception based on frieze and wallpaper groups. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26, 1 (March 2004), 354 – 371.
- [LDD05] LAGAE A., DUMONT O., DUTRÉ P.: Geometry synthesis by example. In *Conf. Shape Modeling and Applications* (2005).
- [LE06] LOY G., EKLUNDH J.: Detecting symmetry and symmetric constellations of features. In *Proc. Europ. Conf. Computer Vision* (2006), pp. 508–521.
- [LHGM05] LAI Y.-K., HU S.-M., GU D. X., MARTIN R. R.: Geometric texture synthesis and transfer via geometry images. In *Symp. Solid and Physical Modeling* (2005), pp. 15–26.
- [LHOKG10] LIU Y., HEL-OR H., KAPLAN C. S., GOOL L. V.: Computational symmetry in computer vision and computer graphics. *Foundations and Trends in Computer Graphics and Vision* 5 (2010), 1–195.
- [LWC*11a] LI Y., WU X., CHRYSANTHOU Y., SHARF A., COHEN-OR D., MITRA N. J.: Globfit: Consistently fitting primitives by discovering global relations. *ACM Transactions on Graphics* 30, 4 (2011), 52:1–52:12.
- [LWC*11b] LI Y., WU X., CHRYSANTHOU Y., SHARF A., COHEN-OR D., MITRA N. J.: Globfit: Consistently fitting primitives by discovering global relations. *ACM Transactions on Graphics* 30, 4 (2011), 52:1–52:12.
- [LWW08] LIPP M., WONKA P., WIMMER M.: Interactive visual editing of grammars for procedural architecture. *ACM Trans. Graph.* 27, 3 (2008), 1–10.
- [LZS*11] LI Y., ZHENG Q., SHARF A., COHEN-OR D., CHEN B., MITRA N. J.: 2d-3d fusion for layer decomposition of urban facades. In *ICCV* (2011).
- [Mar07] MARTINET A.: *Structuring 3D Geometry based on Symmetry and Instancing Information*. PhD thesis, INP Grenoble, 2007.
- [Mer07] MERRELL P.: Example-based model synthesis. In *Symp. Interactive 3D Graphics and Games* (2007), pp. 105–112.
- [MGP06] MITRA N. J., GUIBAS L. J., PAULY M.: Partial and approximate symmetry detection for 3d geometry. *ACM Trans. Graph.* 25, 3 (2006), 560–568.
- [MGP07] MITRA N. J., GUIBAS L., PAULY M.: Symmetrization. vol. 26, p. Article 63.
- [MM08] MERRELL P., MANOCHA D.: Continuous model synthesis. *ACM Trans. Graph.* 27, 5 (2008), 1–7.
- [MM09] MERRELL P., MANOCHA D.: Constraint-based model synthesis. In *SPM '09: 2009 SIAM/ACM Joint Conference on Geometric and Physical Modeling* (New York, NY, USA, 2009), ACM, pp. 101–111.
- [MP08] MITRA N. J., PAULY M.: Symmetry for architectural design. In *Advances in Architectural Geometry* (2008), pp. 13–16.
- [MPWC12] MITRA N. J., PAULY M., WAND M., CEYLAN D.: Symmetry in 3d geometry: Extraction and applications. Eurographics State-of-the-art Report (STAR), 2012.

- [MSHS06] MARTINET A., SOLER C., HOLZSCHUCH N., SIL-LION F.: Accurate detection of symmetries in 3d shapes. *ACM Trans. Graph.* 25, 2 (2006), 439–464.
- [MWH*06] MÜLLER P., WONKA P., HAEGLER S., ULMER A., GOOL L. V.: Procedural modeling of buildings. *ACM Trans. Graph.* 25, 3 (2006), 614–623.
- [MYY*10] MITRA N. J., YANG Y.-L., YAN D.-M., LI W., AGRAWALA M.: Illustrating how mechanical assemblies work. *ACM Transactions on Graphics* 29, 3 (2010).
- [MYY*13] MITRA N. J., YANG Y.-L., YAN D.-M., LI W., AGRAWALA M.: Illustrating how mechanical assemblies work. *ACM CACM Research Highlight* (2013).
- [NYC05] NGUYEN M. X., YUAN X., CHEN B.: Geometry completion and detail generation by texture synthesis. *The Visual Computer* 21, 9–10 (2005), 669–678.
- [OBS04] OHTAKE Y., BELYAEV A., SEIDEL H.-P.: Ridge-valley lines on meshes via implicit surface fitting. In *Proc. Siggraph* (2004), pp. 609–612.
- [OLGM11] OVSJANIKOV M., LI W., GUIBAS L., MITRA N. J.: Exploration of continuous variability in collections of 3d shapes. *ACM Transactions on Graphics* 30, 4 (2011), 33:1–33:10.
- [OSG08] OVSJANIKOV M., SUN J., GUIBAS L.: Global intrinsic symmetries of shapes. In *Eurographics Symposium on Geometry Processing (SGP)* (2008).
- [PM01] PARISH Y. I. H., MÜLLER P.: Procedural modeling of cities. In *Proc. Siggraph 2001* (2001), pp. 301–308.
- [PMW*08] PAULY M., MITRA N. J., WALLNER J., POTTMANN H., GUIBAS L.: Discovering structural regularity in 3D geometry. *ACM Trans. Graph.* 27, 3 (2008).
- [PSG*06] PODOLAK J., SHILANE P., GOLOVINSKIY A., RUSINKIEWICZ S., FUNKHOUSER T.: A planar-reflective symmetry transform for 3D shapes. *ACM Trans. Graph.* 25, 3 (2006).
- [RBBK10] RAVIV D., BRONSTEIN A. M., BRONSTEIN M. M., KIMMEL R.: Full and partial symmetries of non-rigid shapes. *Intl. Journal of Computer Vision (IJCV)* 89, 1 (August 2010), 18–39.
- [SA07] SORKINE O., ALEXA M.: As-rigid-as-possible surface modeling. In *Proceedings of Eurographics/ACM SIGGRAPH Symposium on Geometry Processing* (2007), pp. 109–116.
- [SACO04] SHARF A., ALEXA M., COHEN-OR D.: Context-based surface completion. *ACM Trans. Graph.* 23, 3 (2004), 878–887.
- [SBM*10] STÁVA O., BENES B., MÉCH R., ALIAGA D., KRÍŠTOF P.: Inverse procedural modeling by automatic generation of l-systems. *Computer Graphics Forum* 29, 2 (2010), 665–674.
- [SCSI08] SIMAKOV D., CASPI Y., SHECHTMAN E., IRANI M.: Summarizing visual data using bidirectional similarity. In *IEEE Conf. Comp. Vision and Pattern Recognition (CVPR)* (2008), pp. 1–8.
- [SE98] SINGH K., EUGENE F.: Wires: a geometric deformation technique. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1998), SIGGRAPH '98, ACM, pp. 405–414.
- [SFCH12] SHEN C.-H., FU H., CHEN K., HU S.-M.: Structure recovery by part assembly. *ACM Transactions on Graphics (SIGGRAPH Asia)* 31, 6 (2012), 180:1–180:11.
- [SG71] STINY G., GIPS J.: Shape grammars and the generative specification of painting and sculpture. In *IFIP Congress 71* (Ljubljana, Yugoslavia, 1971).
- [Sha08] SHAMIR A.: A survey on mesh segmentation techniques. *Computer Graphics Forum* 27, 6 (2008), 1539–1556.
- [SJWS13] SUNKEL M., JANSEN S., WAND M., SEIDEL H.-P.: A correlated parts model for object detection in large 3d scans. *Computer Graphics Forum (Special Issue of Eurographics)* (2013). to appear.
- [SKS06] SIMARI P., KALOGERAKIS E., SINGH K.: Folding meshes: hierarchical mesh segmentation based on planar symmetry. In *Proc. Symp. Geometry Processing* (2006), pp. 111–119.
- [SP86] SEDERBERG T. W., PARRY S. R.: Free-form deformation of solid geometric models. In *Proc. Siggraph* (1986), pp. 151–160.
- [STC04] SHAWE-TAYLOR J., CRISTIANINI N.: *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.
- [SvKK*11] SIDI O., VAN KAICK O., KLEIMAN Y., ZHANG H., COHEN-OR D.: Unsupervised co-segmentation of a set of shapes via descriptor-space spectral clustering. *ACM Trans. on Graphics (Proc. SIGGRAPH Asia)* 30, 6 (2011), 126:1–126:10.
- [Tho92] THOMPSON D. W.: *On Growth and Form*. Dover reprint of 1942 2nd ed. (1st ed., 1917), 1992.
- [TLL*11] TALTON J. O., LOU Y., LESSER S., DUKE J., MÉCH R., KOLTUN V.: Metropolis procedural modeling. *ACM Trans. Graph.* 30 (April 2011), 11:1–11:14.
- [TPBF87] TERZOPOULOS D., PLATT J., BARR A., FLEISCHER K.: Elastically deformable models. In *Proc. SIGGRAPH '87* (New York, NY, USA, 1987), ACM, pp. 205–214.
- [TW05] THRUN S., WEGBREIT B.: Shape from symmetry. In *Proc. Int. Conf. Computer Vision* (2005).
- [TYK*12] TALTON J., YANG L., KUMAR R., LIM M., GOODMAN N. D., MÉCH R.: Learning design patterns with bayesian grammar induction. In *Proceedings of the 25th ACM Symposium on User Interface Software and Technology* (New York, NY, USA, 2012), UIST '12, ACM, pp. 63–74.
- [UIM12] UMETANI N., IGARASHI T., MITRA N. J.: Guided exploration of physically valid shapes for furniture design. *ACM Transactions on Graphics* 31, 4 (2012), 86:1–86:11.
- [VDA*12] VANEGAS C., DORADO I. G., ALIAGA D., BENES B., WADDELL P.: Inverse design of urban procedural models. *ACM Transactions on Graphics* 31, 6 (2012), 168:1–168:11.
- [vKTS*11] VAN KAICK O., TAGLIASACCHI A., SIDI O., ZHANG H., COHEN-OR D., WOLF L., HAMARNEH G.: Prior knowledge for shape correspondence. *Computer Graphics Forum (Special Issue of Eurographics)* 30, 2 (2011), 553–562.
- [vKZHC011] VAN KAICK O., ZHANG H., HAMARNEH G., COHEN-OR D.: A survey on shape correspondence. *Computer Graphics Forum* 30, 6 (2011), 1681–1707.
- [WAvK*12] WANG Y., ASAFI S., VAN KAICK O., ZHANG H., COHEN-OR D., CHEN B.: Active co-analysis of a set of shapes. *ACM Transactions on Graphics* 31, 6 (2012).
- [Wer23] WERTHEIMER M.: Untersuchungen zur lehre von der gestalt. ii. *Psychologische Forschung* 4 (1923), 301–350.
- [WHZ*08] WEI L., HAN J., ZHOU K., BAO H., GUO B., SHUM H.: Inverse texture synthesis. *ACM Trans. Graph.* 27, 3 (2008), 52:1–52:9.
- [WL00] WEI L., LEVOY M.: Fast texture synthesis using tree-structured vector quantization. In *Proc. Siggraph 2000* (2000), pp. 479–488.
- [WSW*12] WHITING E., SHIN H., WANG R., OCHSENDORF J., DURAND F.: Structural optimization of 3d masonry buildings. *ACM Transactions on Graphics* 31, 6 (2012).

- [WWSR03] WONKA P., WIMMER M., SILLION F., RIBARSKY W.: Instant architecture. *ACM Trans. Graph.* 22, 3 (2003), 669–677.
- [WXL*11] WANG Y., XU K., LI J., ZHANG H., SHAMIR A., LIU L., CHENG Z., XIONG Y.: Symmetry hierarchy of man-made objects. vol. 30, pp. 287–296.
- [XLZ*10] XU K., LI H., ZHANG H., COHEN-OR D., XIONG Y., CHENG Z.: Style-content separation by anisotropic part scales. *ACM Transactions on Graphics, (Proc. of SIGGRAPH Asia 2010)* 29, 5 (2010).
- [XWY*09] XU W., WANG J., YIN K., ZHOU K., VAN DE PANNE M., CHEN F., GUO B.: Joint-aware manipulation of deformable models. *ACM Trans. Graph.* 28, 3 (2009), 1–9.
- [XZCOC12] XU K., ZHANG H., COHEN-OR D., CHEN B.: Fit and diverse: Set evolution for inspiring 3d shape galleries. *ACM Transactions on Graphics (Special Issue of SIGGRAPH)* 31, 4 (2012), 57:1–57:10.
- [XZJ*12] XU K., ZHANG H., JIANG W., DYER R., CHENG Z., LIU L., CHEN B.: Multi-scale partial intrinsic symmetry detection. *ACM Transactions on Graphics (Special Issue of SIGGRAPH Asia)* 31, 6 (2012), 181:1–181:10.
- [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 (Special Issue of SIGGRAPH Asia)* 28, 5 (2009), 138:1–138:10.
- [XZZ*11] XU K., ZHENG H., ZHANG H., COHEN-OR D., LIU L., XIONG Y.: Photo-inspired model-driven 3d object modeling. *ACM Transactions on Graphics* 30, 4 (2011), 80:1–10.
- [YBY*13] YEH Y.-T., BREEDEN K., YANG L., FISHER M., HANRAHAN P.: Synthesis of tiled patterns using factor graphs. *ACM Transactions on Graphics* (2013), to appear.
- [YK12] YUMER M., KARA L.: Co-abstraction of shape collections. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2012)* 31 (2012), 158:1–158:11.
- [YM09] YEH Y.-T., MÈCH R.: Detecting symmetries and curvilinear arrangements in vector art. *Computer Graphics Forum* 28, 2 (2009), 707–716.
- [YYPM11] YANG Y.-L., YANG Y.-J., POTTMANN H., MITRA N. J.: Shape space exploration of constrained meshes. *ACM Trans. Graph.* 30, 6 (Dec. 2011), 124:1–124:12.
- [ZCC*12] ZHENG Y., CHEN X., CHENG M.-M., ZHOU K., HU S.-M., MITRA N. J.: Interactive images: Cuboid proxies for smart image manipulation. *ACM Transactions on Graphics* 31, 4 (2012), 99:1–99:11.
- [ZCM13] ZHENG Y., COHEN-OR D., MITRA N. J.: Functional substructures for part compatibility. *Computer Graphics Forum* (2013), to appear.
- [ZFCO*11] ZHENG Y., FU H., COHEN-OR D., AU O. K.-C., TAI C.-L.: Component-wise controllers for structure-preserving shape manipulation. In *Computer Graphics Forum (Proc. Eurographics)* (2011).
- [ZG04] ZELINKA S., GARLAND M.: Similarity-based surface modelling using geodesic fans. In *Proc. Symp. Geometry processing* (2004), pp. 204–213.
- [ZG06] ZELINKA S., GARLAND M.: Surfacing by numbers. In *Graphics Interface* (2006), pp. 107–113.
- [ZHH07] ZELEDNIK R. C., HERNDON K. P., HUGHES J. F.: Sketch: an interface for sketching 3d scenes. In *ACM SIGGRAPH 2007 courses* (New York, NY, USA, 2007), SIGGRAPH '07, ACM.
- [ZHW*06] ZHOU K., HUANG X., WANG X., TONG Y., DESBRUN M., BAINING GUO H.-Y. S.: Mesh quilting for geometric texture synthesis. *ACM Trans. Graph.* 25, 3 (2006), 690–697.
- [ZTY*12] ZHAO X., TANG C.-C., YANG Y.-L., POTTMANN H., MITRA N. J.: Intuitive design exploration of constrained meshes. In *Advances in Architectural Geometry* (2012).
- [ZXS*12] ZHU L., XU W., SNYDER J., LIU Y., WANG G., GUO B.: Motion-guided mechanical toy modeling. *ACM Trans. Graph.* 31, 6 (2012).