

Data-Driven Shape Analysis and Processing

Kai Xu^{1,2} Vladimir G. Kim³ Qixing Huang⁴ Niloy Mitra⁵ Evangelos Kalogerakis⁶

¹ National University of Defense Technology ² Shenzhen VisuCA Key Lab / SIAT ³ Adobe Research

⁴ Toyota Technological Institute at Chicago ⁵ University College London ⁶ University of Massachusetts Amherst

Abstract

Data-driven methods serve an increasingly important role in discovering geometric, structural, and semantic relationships between shapes. In contrast to traditional approaches that process shapes in isolation of each other, data-driven methods aggregate information from 3D model collections to improve the analysis, modeling and editing of shapes. Through reviewing the literature, we provide an overview of the main concepts and components of these methods, as well as discuss their application to classification, segmentation, matching, reconstruction, modeling and exploration, as well as scene analysis and synthesis. We conclude our report with ideas that can inspire future research in data-driven shape analysis and processing.

Keywords: machine learning, geometry processing, geometry analysis

Concepts: •Computing methodologies → Image manipulation; Computational photography;

1 Introduction

As the availability of 3D data increases, due to the developments in both 3D sensing technology as well as 3D modeling software, data-driven approaches become increasingly applicable and useful to 3D shape processing. In contrast to traditional approaches [Levy and Zhang 2011], data-driven methods look beyond single objects, instead analyzing sets of shapes jointly to extract meaningful mappings and correlations between them. In addition, these methods are able to learn from data computational models that effectively reason about properties and relationships of shapes without relying on hard-coded rules or explicitly programmed instructions. Leveraging shared information across multiple objects, data-driven methods are able to facilitate high-level shape understanding through discovering geometric and structural patterns among collections of shapes, patterns which serve as strong priors in various geometry processing applications.

The idea of utilizing data to support geometry processing has been exploited and practiced for many years. However, most existing works based on this idea are confined to the example-based paradigm, mostly leveraging only one core concept of data-driven techniques – *information transfer*. Typically, the input to these problems includes one or multiple exemplar shapes with some prescribed or precomputed information of interest, and a target shape that needs to be analyzed or processed. These techniques usually establish a *correlation* between the source and the target shapes and transfer the interesting information from the source to the target. The applications of such approaches include a variety of methods

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s). © 2016 Copyright held by the owner/author(s).

SIGGRAPH Asia 2016 Courses, July 24-28, 2016, Anaheim, CA

ISBN: 978-1-4503-ABCD-E/16/07

DOI: <http://doi.acm.org/10.1145/9999997.9999999>

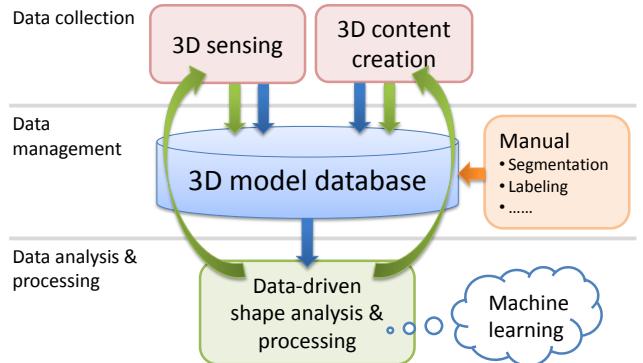


Figure 1: Data-driven shape processing and modeling provides a promising solution to the development of “big 3D data”. The two major ways of 3D data generation, 3D sensing and 3D content creation, populate 3D databases with fast growing amount of 3D models. The database models are sparsely augmented with manual segmentation and labeling, as well as reasonably organized, to support data-driven shape analysis and processing, based on, e.g., machine learning techniques. The learned knowledge can in turn support efficient 3D reconstruction and 3D content creation, during which the knowledge can be transferred to the newly generated data. Such 3D data with semantic information can be included into the database to enrich it and facilitate further data-driven applications.

in shape analysis (e.g. [Schaefer and Yuksel 2007a]) and shape synthesis (e.g. [Merrell 2007; Ma et al. 2014]).

As the availability of 3D data increases, several new concepts in data-driven methods are emerging, opening space for new developments in shape analysis and content creation. *First*, the rich variability of 3D content in existing shape repositories makes it possible to directly reuse the shapes or parts for constructing new 3D models [Funkhouser et al. 2004]. *Content reuse* for 3D modeling is perhaps the most straightforward application of big 3D geometric data, providing a promising approach to address the challenging 3D content creation problem. *Second*, high-level shape understanding can benefit from co-analyzing collections of shapes. Several analysis tools demonstrate that shape analysis is more reliable if it is supported by observing certain attributes across a set of semantically related shapes instead of just focusing on a single object. *Co-analysis* requires a critical step of finding correlations between multiple shapes in the input set, which is substantially different from building pair-wise correlation. A key concept to co-analysis is the *consistency* of the correlations across the entire set, which has both semantic [Kalogerakis et al. 2010; Sidi et al. 2011; Wang et al. 2012] and mathematical [Huang and Guibas 2013a] justifications. *Third*, aside from analyzing patterns from a set of shapes, it is also possible to endorse a subset of the shapes with some semantic information (e.g., part labeling), which can be propagated to the other shapes through learned mappings. This information propagation evolves the concept of *knowledge transfer* between shapes.

Relation to knowledge-driven shape processing. Prior to the emergence of data-driven techniques, high-level shape understanding and modeling was usually achieved with knowledge-driven methods. In the knowledge-driven paradigm, geometric and structural patterns are extracted and interpreted with the help of explicit rules or hand-crafted parameters. Such examples include heuristics-based shape segmentation [Shamir 2008] and procedural shape modeling [Müller et al. 2006]. Although these approaches have certain empirical success, they exhibit several inherent limitations. First, it is extremely difficult to hard-code explicit rules and heuristics that can handle the enormous geometric and structural variability of 3D shapes and scenes in general. As a result, knowledge-driven approaches are often hard to generalize well to large and diverse shape collections. Another issue is that non-experts find it difficult to interact with knowledge-driven techniques that require as input “low-level” geometric parameters or instructions.

In contrast to knowledge driven methods, data-driven techniques learn representations and parameters from data. They usually do not depend on hard-coded prior knowledge, and consequently do not rely on hand-crafted parameters, making these techniques more data-adaptive and thus lead to significantly improved performance in many practical settings. The success of data-driven approaches, backed by machine learning techniques, heavily relies on the accessibility of large data collections. We have witnessed the successful performance improvement of machine learning algorithms by increasing the training set size [Banko and Brill 2001]. In light of this, the recent developments in 3D modeling tools and acquisition techniques for 3D geometry, as well as availability of large repositories of 3D shapes (e.g., Trimble 3D Warehouse, Yobi3D , etc.), offer great opportunities for developing data-driven approaches for 3D shape analysis and processing.

Relation to structure-aware shape processing. This report is closely related to the recent survey on “structure-aware shape processing” by Mitra and co-workers [Mitra et al. 2014], which concentrates on techniques for structural analysis of 3D shapes, as well as high-level shape processing guided by structure-preservation. In that survey, shape structure is defined as the arrangement and relations between shape parts, which is analyzed through identifying shape parts, part parameters, and part relations. Each of the three can be determined through manual assignment, predefined model fitting and data-driven learning.

In contrast, our report takes a very different perspective—we focus on how the increasing availability of geometric data has changed the field of shape analysis and processing. In particular, we want to highlight several key distinctions: *First*, data-driven shape processing goes beyond structure analysis. For example, leveraging large shape collections may benefit a wider variety of problems in shape understanding and processing, such as parametric modeling of shape space [Allen et al. 2003], hypothesis generation for object and scene understanding [Zia et al. 2013; Satkin et al. 2012], and information transfer between multi-modal data [Wang et al. 2013b; Su et al. 2014]. Data-driven shape processing may also exploit the data-centered techniques in machine learning such as sparse representation [Ren and Ramanan 2013] and feature learning [Hinton et al. 2006; Bengio 2009; Yu and Ng 2010; Krizhevsky et al. 2012], which are not pre-conditioned on any domain-specific or structural prior beyond raw data. *Second*, even within the realm of structure-aware shape processing, data-driven approaches are arguably becoming dominant due to their theoretical and practical advantages, as well as the availability of large shape repositories and recent developments in machine learning.

Vision and motivation. With the emergence of “big data”, many scientific disciplines have shifted their focus to data-driven techniques. Although 3D geometry data is still far from being as ubiquitous as some other data formats (e.g., photographs), the rapidly growing number of 3D models, the recent developments in fusing 2D and 3D data, and the invention of commodity depth sensors, have made the era of “big 3D data” more promising than ever. At the same time, we expect data-driven approaches to take one of the leading roles in the reconstruction and understanding of acquired 3D data, as well as the synthesis of new shapes. Data-driven geometry processing will close the loop starting from acquisition, analysis, and processing all the way to the generation of 3D shapes (see Figure 1), and will be a key tool for manipulating big visual data.

Recent years have witnessed a rapid development of data-driven geometry processing algorithms, both in the computer graphics and computer vision communities. Given the research efforts and wide interests in the subject, we believe many researchers would benefit from a comprehensive and systematic survey. We also hope such a survey can stimulate new theories, problems, and applications.

Organization. This survey is organized as follows. Section 2 gives a high-level overview of data-driven approaches and classifies data-driven methods with respect to their application domains. This section also provides two representative examples for the readers to understand the general work-flow of data-driven geometry processing. The sections following survey the various data-driven shape processing problems in detail, and try to correlate the different methods through comparisons in various aspects. Finally, we conclude our survey by discussing a few key problems involved in designing a data-driven method for shape processing, listing a set of open challenges in this direction, as well as providing a vision on future research.

Accompanying online resources. In order to assist the reader in learning and leveraging the basic algorithms, we provide an online wikipage [Xu et al. 2016], which collects tools and source code, together with benchmark data for typical problems and applications of data-driven shape processing. This page will also maintain links and data mining tools for obtaining large data collections of shapes and scenes. This website could serve as a starting point for those who are conducting research in this direction. We also expect it to benefit a wide spectrum of researchers from related fields.

2 Overview

In this section, we provide a high-level overview of the main components and steps of data-driven approaches for processing 3D shapes and scenes. Although the pipeline of these methods vary significantly depending on their particular applications and goals, a number of components tend to be common: the input data collection and processing, data representations and feature extraction, as well as learning and inference. *Representation, learning and inference* are critical components of machine learning approaches in general [Koller and Friedman 2009b]. In the case of shape and scene processing, each of these components poses several interesting and unique problems when dealing with 3D geometric data. These problems have greatly motivated the research on data-driven geometry processing, and in turn have brought new challenges to the computer vision and machine learning communities, as reflected by the increased interest in 3D visual data from these fields. Below, we discuss particular characteristics and challenges of data-driven 3D shape and scene processing algorithms. Figure 2 provides a schematic overview of the most common components of these algorithms.

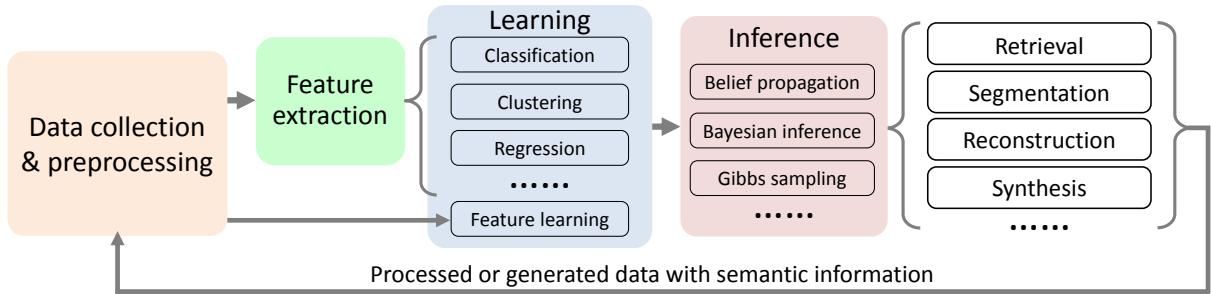


Figure 2: The general pipeline of data-driven geometry processing contains four major stages: data collection and preprocessing, feature extraction (or feature learning), learning and inference. The inference supports many applications which would produce new shapes or scenes through reconstruction modeling or synthesis. These new data, typically possessing labels for shapes or parts, can be used to enrich the input datasets and enhance the learning tasks in future, forming a data-driven geometry processing loop.

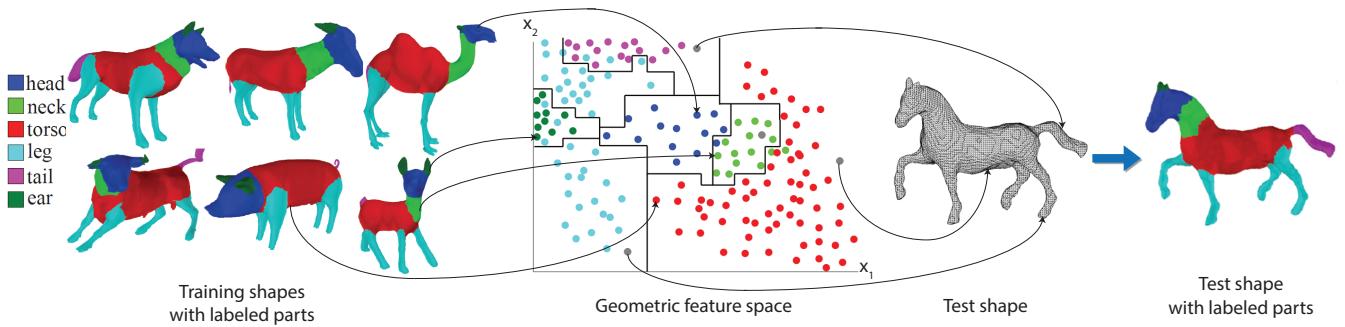


Figure 3: Pipeline of a supervised segmentation algorithm [Kalogerakis et al. 2010]. Given a set of shapes with labeled parts, the points of each shape are embedded in a common feature space based on their local geometric descriptors (a color is assigned to points depending on their given part label). A classifier is learned to split the feature space into regions corresponding to each part label. Given a test shape, its points (shown in grey) are first embedded in the same space. Then part labels are inferred for all its points based on the learned classifier and an underlying structured probabilistic model (Section 4).

2.1 3D data collection

Shape representation. A main component of data-driven approaches for shape and scene processing is data collection, where the goal is to acquire a number of 3D shapes and scenes depending on the application. When shapes and scenes are captured with scanners or depth sensors, their initial representation is in the form of *range data* or *unorganized point clouds*. Several data-driven methods for reconstruction, segmentation and recognition directly work on these representations and do not require any further processing. On the other hand, online repositories, such as the Trimble 3D Warehouse, contain millions of shapes and scenes that are represented as *polygon meshes*. A large number of data-driven techniques are designed to handle complete shapes in the form of polygon meshes created by 3D modeling tools or re-constructed from point clouds. Choosing which representation to use depends on the application. For example, data-driven reconstruction techniques aim for generating complete shapes and scenes from noisy point clouds with missing data. The reconstructed shapes can then be processed with other data-driven methods for categorization, segmentation, matching and so on. Developing methods that can handle any 3D data representation, as well as jointly reconstructing and analyzing shapes is a potential direction for future research we discuss in Section 11.

When polygon meshes are used as the input representation, an important aspect to consider is whether and how data-driven methods will deal with possible “defects”, such as non-manifold and non-orientable sets of polygons, inverted faces, isolated elements, self-intersections, holes and topological noise. The vast majority of

meshes available in online repositories have these problems. Although there is a number of mesh repairing tools (see [Campen et al. 2012] for a survey), they may not handle all different types of “defects”, and can take a significant amount of time to process each shape in a large dataset. To avoid the issues caused by these “defects”, some data-driven methods uniformly sample the input meshes and work on the resulting point-based representation instead (e.g., [Chaudhuri et al. 2011a; Kim et al. 2013a]).

Datasets. Although it is desirable to develop data-driven methods that can learn from a handful of training shapes or scenes, this is generally a challenging problem in machine learning [Fei-Fei et al. 2006]. Several data-driven methods in computer vision have been particularly successful due to the use of very large datasets that can reach the size of several millions of images [Torralba et al. 2008]. In contrast, data-driven approaches for 3D shape and scene processing approaches have mostly relied on datasets that reach the order of a few thousands so far (e.g., Princeton Shape Benchmark [Shilane et al. 2004b], or datasets collected from the web [Kim et al. 2013a]). Online repositories contain large amount of shapes, which can lead to the development of methods that will leverage datasets that are orders of magnitudes larger than the ones currently used. One significant example is the recently available ShapeNet [Su et al. 2015c], which provides a richly-annotated, large-scale dataset of 3D shapes. Similar to ImageNet, a well-known image database in the computer vision community, ShapeNet is organized based on the WordNet hierarchy. It has indexed about 3 million models, out of which 220 thousand models are classified into 3,135 WordNet

synsets (a synset refers to a meaningful concept in WordNet).

Another possibility is to develop synthetic datasets. A notable example is the pose and part recognition algorithm used in Microsoft’s Kinect that relies on 500K synthesized shapes of human bodies in different poses [Shotton et al. 2011]. In general, large datasets are important to capture the enormous 3D shape and scene variability, and can significantly increase the predictive performance and usability of learning methods. A more comprehensive summary of existing online data collections can be found on our wikipage [Xu et al. 2016].

2.2 3D data processing and feature representation

It is common to perform some additional processing on the input representations of shapes and scenes before executing the main learning step. The reason is that the input representations of 3D shapes and scenes can have different resolutions (e.g., number of points or faces), scale, orientation, and structure. In other words, the input shapes and scenes do not initially have any type of common parameterization or alignment. This is significantly different from other domains, such as natural language processing or vision, where text or image datasets frequently come with a common parameterization beforehand (e.g., images with the same number of pixels and objects of consistent orientation).

To achieve a common parameterization of the input shapes and scenes, one popular approach is to embed them in a *common geometric feature space*. For this purpose a variety of shape descriptors have been developed. These descriptors can be classified into two main categories: *global shape descriptors* that convert each shape to a feature vector, and *local shape descriptors* that convert each point to a feature vector. Examples of global shape descriptors are Extended Gaussian Images [Horn 1984], 3D shape histograms [Ankerst et al. 1999; Chaudhuri and Koltun 2010], spherical functions [Saupe and Vrancic 2001], lightfield descriptors [Chen et al. 2003a], shape distributions [Osada et al. 2002], symmetry descriptors [Kazhdan et al. 2004a], spherical harmonics [Kazhdan et al. 2003b], 3D Zernicke moments [Novotni and Klein 2003], and bags-of-words created out of local descriptors [Bronstein et al. 2011]. Local shape descriptors include surface curvature, PCA descriptors, local shape diameter [Shapira et al. 2008], shape contexts [Belongie et al. 2002; Kalogerakis et al. 2010; Kokkinos et al. 2012], spin images [Johnson and Hebert 1999b], geodesic distance features [Zhang et al. 2005], heat-kernel descriptors [Bronstein et al. 2011], and depth features [Shotton et al. 2011]. Global shape descriptors are particularly useful for shape classification, retrieval and organization. Local shape descriptors are useful for partial shape matching, segmentation, and point correspondence estimation. Before using any type of global or local descriptor, it is important to consider whether the descriptor will be invariant to different shape orientations, scales, or poses. In the presence of noise and irregular mesh tessellations, it is important to robustly estimate local descriptors, since surface derivatives are particularly susceptible to surface and sampling noise [Kalogerakis et al. 2007]. It is also possible to use several different descriptors as input, and let the learning step decide which ones are more relevant for each class of shapes [Kalogerakis et al. 2010].

A different approach, which has attracted large attention in the computer vision community, is to avoid manually engineered features and instead directly learn features them from raw data. This approach has been enlightened by the recent developments in deep learning [Bengio 2009; Yu and Ng 2010], and in particular Convolutional Neural Networks (CNNs) [Krizhevsky et al. 2012; Szegedy et al. 2015]. A number of deep learning architectures have been recently proposed to learn 3D shape and scene descriptors, operating

on either voxel-based representations [Wu et al. 2015], view-based projections [Su et al. 2015a; Xie et al. 2015], spectral representations [Boscaini et al. 2015], or RGB-D data [Socher et al. 2012; Blum et al. 2012; Lai et al. 2013; Bo et al. 2014].

Instead of embedding shapes in a common geometric feature space, several methods instead try to directly align shapes in Euclidean space. We refer the reader to the survey on dynamic geometry processing for a tutorial on rigid and non-rigid registration techniques [Chang et al. 2012a]. An interesting extension of these techniques is to include the alignment process in the learning step of data-driven methods, since it is inter-dependent with other shape analysis tasks such as shape segmentation and correspondences [Kim et al. 2013a].

Some data-driven methods require additional processing steps on the input. For example, learning deformation handles or fully generative models of shapes usually rely on segmenting the input shapes into parts with automatic algorithms [Huang et al. 2011; Sidi et al. 2011] and representing these parts with surface abstractions [Yumer and Kara 2012a] or descriptors [Kalogerakis et al. 2012a]. To decrease the amount of computation required during learning, it is also common to represent the shapes as a set of patches (superfaces) [Huang et al. 2011] inspired by the computation of superpixels in image segmentation.

2.3 Learning and Inference

The processed representations of shapes and scenes are used to perform learning and inference for a variety of applications: shape classification, segmentation, matching, reconstruction, modeling, synthesis, and scene analysis. The learning procedures significantly vary depending on the application, thus we discuss them individually in each of the following sections on these applications. As a common theme, learning is viewed as an *optimization* problem that runs on a set of variables representing geometric, structural, semantic or functional properties of shapes and scenes. There is usually a single or multiple objective (or loss) functions for quantifying preferences for different models or patterns governing the 3D data. After learning a model from the training data, inference procedures are used to predict values of variables for new shapes or scenes. Again, the inference procedures vary depending on the application, and are discussed separately in the following sections. It is common that inference itself is an optimization problem, and sometimes is part of the learning process when there are latent variables or partially observed input shapes or scene data.

A general classification of the different types of algorithms used in data-driven approaches for shape and scene processing can be derived from the type of input information available during learning:

- **Supervised learning** algorithms are trained on a set of shapes or scenes annotated with labeled data. For example, in the case of shape classification, these labeled data can have the form of tags, while in the case of segmentation, the labeled data have the form of segmentation boundaries or part labels. The labeled data can be provided by humans or generated synthetically. After learning, the learned models are applied on different sets of shapes (test shapes) to produce results relevant to the task.
- **Unsupervised** algorithms co-analyze the input shapes or scenes without any additional labeled data i.e., the desired output is unknown beforehand. The goal of these methods is to discover correlations in the geometry and structure of the input shape or scene data. For example, unsupervised shape segmentation methods usually perform some type of clustering in the feature space of points or patches belonging to the

input shapes.

- **Semi-supervised** algorithms make use of shapes (or scenes) with and without any labeled data. Active learning is a special case of semi-supervised learning in which a learning algorithm interactively queries the user to obtain desired outputs for more data points related to shapes.

In general, supervised methods tend to output results that are closer to what a human would expect given the provided labeled data. However, they may fail to produce desirable results when the training shapes (or scenes) are geometrically and structurally dissimilar from the test shapes (or scenes). They also tend to require a substantial amount of labeled information as input, which can become a significant burden for the user. Unsupervised methods can deal with collections of shapes and scenes with larger variability and require no human supervision. However, they sometimes require parameter tuning to yield the desired results. Semi-supervised methods represent a trade-off between supervised and unsupervised methods: they provide more direct control to the user about the desired result compared to unsupervised methods, and often produce considerable improvements in the results by making use of both labeled and unlabeled shapes or scenes compared to supervised methods.

The data-driven loop. An advantageous feature of data-driven shape processing is that the output data, produced by learning and inference, typically come with rich semantic information. For example, data-driven shape segmentation produces parts with semantic labels [Kalogerakis et al. 2010]; data-driven reconstruction is commonly coupled with semantic part or shape recognition [Shen et al. 2012; Nan et al. 2012]; data-driven shape modeling can generate readily usable shapes inheriting the semantic information from the input data [Xu et al. 2011]. These processed and generated data can be used to enrich the existing shape collections with both training labels and reusable contents, which in turn benefit subsequent learning. In a sense, data-driven methods *close the loop of data generation and data analysis* for 3D shapes and scenes; see Figure 2. Such concept has been practiced in several prior works, such as the data-driven shape reconstruction framework proposed in [Pauly et al. 2005a] (Figure 12).

Pipeline example. To help the reader grasp the pipeline of data-driven methods, a schematic overview of the components is given in Figure 2. Depending on the particular application, the pipeline can have several variations, or some components might be skipped. We discuss the main components and steps of algorithms for each application in more detail in the following sections. A didactic example of the pipeline in the case of supervised shape segmentation is shown in Figure 3. The input shapes are annotated with labeled part information. A geometric descriptor is extracted for each point on the training shapes, and the points are embedded in a common feature space. The learning step uses a classification algorithm that non-linearly separates the input space into a set of regions corresponding to part labels in order to optimize classification performance (more details are provided in Section 4). Given a test shape, a probabilistic model is used to infer part labels for each point on that shape based on its geometric descriptor in the feature space.

2.4 A comparative overview

Before reviewing the related works in detail, we provide a comparative overview of them in Table 5, and correlate them under a set of *criteria*:

- **Training data.** Data-driven methods can be categorized according to the shape or scene representations they operate

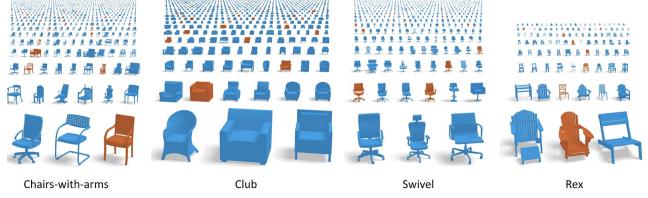


Figure 4: Fine-grained classification of 3D models [Huang et al. 2013a], where text labels are propagated from brown to blue models.

on, the scale (size) of the training datasets they use, and the type of pre-processing applied to these datasets. The most common representation for shapes are polygon meshes and point clouds. 3D scenes are typically represented as an arrangement of individual shapes, usually organized in a scene graph. Pre-processing includes pre-segmentation, over-segmentation, pre-alignment, initial correspondence, or/and labeling.

- **Features.** Roughly speaking, there are two types of feature representations involved in data-driven shape processing. The most commonly used feature representations are low-level ones, such as local geometric features (e.g., local curvature) and global shape descriptors (e.g., shape distribution [Osada et al. 2002]). If the input shapes are pre-segmented into meaningful parts, high-level structural representations (spatial relationships of parts) can be derived. Generally, working with high-level feature representations enables the learning of more powerful models for more advanced inference tasks, such as structural analysis [Mitra et al. 2014], on complex man-made objects and scenes.
- **Learning model/approach.** The specific choice of learning method is often application-dependent. In most cases, machine learning techniques are adapted or developed from scratch to process geometric data. For some problems, such as shape correspondence, the core problem is to extract geometric correlations between different shapes in an unsupervised manner, which itself can be seen as a learning problem specific to geometry processing.
- **Learning type.** As discussed above, there are three basic types of data-driven methods, depending on the use of labeled training data: supervised, semi-supervised and unsupervised methods.
- **Learning outcome.** Learning can produce different types of outputs: parametric or non-parametric models (classifiers, clusterings, regressors, etc.), distance metrics which can be utilized for further analysis, and/or feature representations learned from raw data.
- **Application.** The main applications of data-driven shape analysis and processing include classification, segmentation, correspondence, modeling, synthesis, reconstruction, exploration and organization.

3 Shape classification

Data-driven techniques commonly make assumptions about the size and homogeneity of the input data set. In particular, existing analysis techniques often assume that all models belong to the same class of objects [Kim et al. 2013a] or scenes [Fisher et al. 2011], and cannot directly scale to entire repositories such as the Trimble 3D

Method	Input Data	Shapes	Classes	Acc
[Huang et al. 2013a]	3D Warehouse	1206-5850	26	86
[Golovinskiy et al. 2009]	LIDAR	1063	16	58
[Shilane and Funkhouser 2007]	PSB	1814	90	75
[Funkhouser and Shilane 2006]	PSB	1814	90	83
[Barutcuoglu and DeCoro 2006]	PSB	1814	90	84
[Bronstein et al. 2011]	SG	715	13	89
[Litman et al. 2014]	SG	715	13	91
[Li et al. 2012]	SHREC12	1200	60	88
[Li et al. 2014]	SHREC14	400-10 ⁴	1352	87
[Wu et al. 2015]	ModelNet40	48,000	40	77
[Su et al. 2015a]	ModelNet40	48,000	40	90
[Su et al. 2016]	ModelNet40	48,000	40	94

Table 1: Performance of several methods for shape classification (the accuracy in the right-most column as measured as fraction of correctly-labeled shapes). Huang et al. [Huang et al. 2013a] predict fine-grained tag attributes for big collections of similar shapes. Golovinskiy et al. [Golovinskiy et al. 2009] propose a method for classifying point clouds of objects in urban environments. The methods aimed at classifying meshes are evaluated on Princeton Shape Benchmark (PSB) [Shilane and Funkhouser 2007; Funkhouser and Shilane 2006; Barutcuoglu and DeCoro 2006]. To evaluate performance of the method in the presence of non-rigid deformations ShapeGoogle (SG) dataset is also commonly used [Bronstein et al. 2011; Litman et al. 2014]. Several recent techniques use uniformly sampled representations (volumetric and view-based images) of 3D shapes in conjunction with neural networks [Wu et al. 2015; Su et al. 2015a; Su et al. 2016].

Warehouse [Trimble 2014]. Similarly, techniques for data-driven reconstruction of indoor environments assume that the input data set only has furniture models [Nan et al. 2012], while modeling and synthesis interfaces restrict the input data to particular object or scene classes [Chaudhuri et al. 2011a; Kalogerakis et al. 2012a; Fisher et al. 2012]. Thus, as a first step these methods need to query a 3D model repository to retrieve a subset of relevant models.

Most public shape repositories such as 3D Warehouse [Trimble 2014] rely on the users to provide tags and names of the shapes with little additional quality control measures. As a result, the shapes are sparsely labeled with inconsistent and noisy tags. This motivates developing automatic algorithms to infer text associated with models. Existing work focuses on establishing class memberships for an entire shape (e.g. this shape is a chair), as well as inferring finer-scale attributes (e.g. this chair has a rocking leg).

Classification methods assign a class membership for unlabeled shapes. One approach is to retrieve for each unlabeled shape the most similar shape from a database of 3D models with known shape classes. There has been a large number of shape descriptors proposed in recent years that can be used in such a retrieval task, and one can refer to various surveys (e.g., [Tangelder and Veltkamp 2008b]) for a thorough overview and comparisons. One can further improve classification results by leveraging machine learning techniques to learn classifiers that are based on global shape descriptors [Frome et al. 2004; Golovinskiy et al. 2009]. Barutcuoglu et al. [Barutcuoglu and DeCoro 2006] demonstrate that Bayesian aggregation can be used to improve classification of shapes that are a part of a hierarchical ontology of objects. Geometry matching algorithms also facilitate distinguishing important features for classification [Funkhouser and Shilane 2006; Shilane and Funkhouser 2007]. Bronstein et al. [Bronstein et al. 2011] leverage “bag of features” to learn powerful descriptor-space metrics for non-rigid shapes. These techniques can be further improved by using sparse

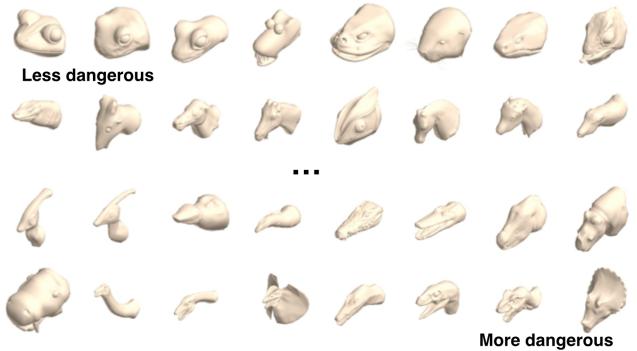


Figure 5: Ranking of parts with respect to “dangerous” attribute (image from [Chaudhuri et al. 2013])

coding techniques [Litman et al. 2014] and perform well on benchmarks [Li et al. 2012; Li et al. 2014]. Motivated by the success of deep neural networks in image classification, Wu et al. [Wu et al. 2015] represent a shape by a volumetric occupancy grid and train neural network to classify them. Su et al. [Su et al. 2015a] demonstrate that by rendering the shape from multiple viewpoints and analyzing the views as images enables leveraging the power of existing neural networks that were trained for image analysis. Su et al. [Su et al. 2016] further combine both shape representations by sampling volumetric occupancy grids with anisotropic view-dependent kernels. See Table 1 for a brief summary of some methods.

Tag attributes often capture fine-scale attributes of shapes that belong to the same class. These attributes can include presence or absence of particular parts, object style, or comparative adjectives. Huang et al. [Huang et al. 2013a] developed a framework for propagating these attributes in a collection of partially annotated 3D models. For example, only brown models in Figure 4 were labeled, and blue models were annotated automatically. To achieve automatic labeling, they start by co-aligning all models to a canonical domain, and generate a voxel grid around the co-aligned models. For each voxel they compute local shape features, such as spin images, for each shape. Then, they learn a distance metric that best discriminates between different tags. All shapes are finally embedded in a weighted feature space where nearest neighbors are connected in a graph. A graph cut clustering is used to assign tags to unlabeled shapes. Tag attributes can also be used to describe semantics, function, or style of parts in shapes. Data-driven consistent segmentation and labeling techniques can be applied to propagate part tags across shapes (see Section 4). An alternative approach is to partition shapes into multiple sets of parts, then extract descriptors to define part similarity. A characteristic example of such an approach was demonstrated in Shapira et al. [Shapira et al. 2010]. Given the hierarchical segmentations of 3D shapes as input, part tagging was achieved by comparing local geometric features of parts as well as their context within the whole shape.

While the above method works well for discrete tags, they do not capture more continuous relations, such as animal A is more dangerous than animal B. Chaudhuri et al. [Chaudhuri et al. 2013] focus on estimating ranking based on comparative adjectives. They use crowdsourcing to gather pairwise comparisons of shape parts with respect to different adjectives, and use a Support Vector Machine ranking method to predict attribute strengths from shape features for novel shape parts (Figure 5).

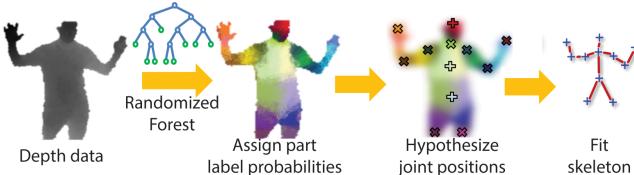


Figure 6: A random forest classifier applied on depth data representing a human body shape (image from [Fossati et al. 2013])

Style similarity methods have recently been proposed to classify shapes into style-related categories e.g., buildings can be classified into architectural styles, such as Gothic, Baroque, Byzantine and so on. In contrast to the previously discussed approaches that rely on generic visual similarity measures to compare shapes, these methods learn distance functions for style elements [Lun et al. 2015] or common feature spaces [Liu et al. 2015] to quantify the stylistic similarity of shapes. The methods can be used to compare the style similarity of shapes, even when these belong to different classes (e.g., chairs and lamps). To align the style similarity measures with the human perception of style, style comparisons of shapes are gathered through crowdsourcing. The learned similarity measures can be used to retrieve stylistically similar shapes to populate a scene, or associate shapes with style-related tags.

While the techniques described above are suitable for retrieving and classifying shapes, a large number of applications require a more involved structural analysis to infer semantic and functional properties of shapes or their parts. The following two sections will discuss methods that perform structural analysis in collections of shapes based on segmentation and local matching.

4 Shape segmentation

The goal of data-driven shape segmentation is to partition the shapes of an input collection into parts, and also estimate part correspondences across these shapes. We organize the literature on shape segmentation into the following three categories: supervised segmentation, unsupervised segmentation, and semi-supervised segmentation following the main classification discussed in Section 2. Table 2 summarizes representative techniques and reports their segmentation and part labeling performance based on established benchmarks. Table 3 reports characteristic running times for the same techniques.

4.1 Supervised shape segmentation

Classification techniques. Supervised shape segmentation is frequently formulated as a classification problem. Given a training set of shapes containing points, faces or patches that are labeled according to a part category (see Figure 3), the goal of a classifier is to identify which part category other points, faces, or patches from different shapes belong to. Supervised shape segmentation is executed in two steps: during the first step, the parameters of the classifier are learned from the training data. During the second step, the classifier is applied on new shapes. A simple linear classifier has the form:

$$c = f\left(\sum_j \theta_j \cdot x_j\right) \quad (1)$$

where x_j is a geometric feature of a point (face, or patch), such as the ones discussed in Section 2. The parameters θ_j serve as weights for each geometric feature. The function f is non-linear and maps to a discrete value (label), which is a part category, or to probabilities per category. In general, choosing a good set of

geometric features that help predicting part labels, and employing classifiers that can discriminate the input data points correctly are important design choices. There is no rule of thumb on which is the best classifier for a problem. This depends on the underlying distribution and characteristics of the input geometric features, their dimensionality, amount of labeled data, existence of noise in the labeled data or shapes, training and test time constraints - for a related discussion on how to choose a classifier for a problem, we refer the reader to [Manning et al. 2008]. Due to the large dimensionality and complexity of geometric feature spaces, non-linear classifiers are more commonly used. For example, to segment human bodies into parts and recognize poses, the Microsoft’s Kinect uses a random forest classifier trained on synthetic depth images of humans of many shapes and sizes in highly varied poses sampled from a large motion capture database [Shotton et al. 2011] (Figure 6).

Structured models. For computer graphics applications, it is important to segment shapes with accurate and smooth boundaries. For example, to help the user create a new shape by re-combining parts from other shapes [Funkhouser et al. 2004], irregular and noisy segmentation boundaries can cause problems in the part attachment. From this aspect, using a classifier per point/face independently is usually not enough. Thus, it is more common to formulate the shape segmentation problem as an energy minimization problem that involves a unary term assessing the consistency of each point/face with each part label, as well as a pairwise term assessing the consistency of neighboring points/faces with pairs of labels. For example, pairs of points that have low curvature (i.e., are on flat surface) are more likely to have the same part label. This energy minimization formulation has been used in several single-shape and data-driven segmentations (unsupervised or supervised) [Katz and Tal 2003; Anguelov et al. 2005c; Shapira et al. 2010; Kalogerakis et al. 2010]. In the case of supervised segmentation [Kalogerakis et al. 2010], the energy can be written as:

$$E(\mathbf{c}; \theta) = \sum_i E_{\text{unary}}(c_i; \mathbf{x}_i, \theta_1) + \sum_{i,j} E_{\text{pairwise}}(c_i, c_j; \mathbf{y}_{ij}, \theta_2) \quad (2)$$

where $\mathbf{c} = \{c_i\}$ is a vector of random variables representing the part label per point (or face) i , \mathbf{x}_i is its geometric feature vector, i, j are indices to points (or faces) that are considered neighbors, \mathbf{y}_{ij} is a geometric feature vector representing dihedral angle, angle between normals, or other features, and $\theta = \{\theta_1, \theta_2\}$ are the energy parameters. The important difference of supervised data-driven methods with previous single-shape segmentation methods is that the parameters θ are automatically learned from the training shapes to capture complex feature space patterns per part [Anguelov et al. 2005c; Kalogerakis et al. 2010]. We also note that the above energy of Equation 2, when written in an exponentiated form and normalized, can be treated as a probabilistic graphical model [Koller and Friedman 2009b], called Conditional Random Field [Lafferty et al. 2001] that represents the joint probability distribution over part labels conditioned on the input features:

$$P(\mathbf{c}|\mathbf{x}, \mathbf{y}, \theta) = \exp(-E(\mathbf{c}; \theta))/Z(\mathbf{x}, \mathbf{y}, \theta) \quad (3)$$

where $Z(\mathbf{x}, \mathbf{y}, \theta)$ is a normalization factor, also known as partition function. Minimizing the energy of Equation 2, or correspondingly finding the assignment \mathbf{c} that maximizes the above probability distribution is known as a Maximum A Posteriori inference problem that can be solved in various manners, such as graph cuts, belief propagation, variational or linear programming relaxation techniques [Koller and Friedman 2009b].

The parameters θ can be jointly learned through maximum likelihood (ML) or maximum a posteriori (MAP) estimates [Koller and Friedman 2009b]. However, due to high computational complexity

Segmentation method	Learning type	Type of manual input	PSB rand index (# train. shapes if applicable)	L-PSB accuracy (# train. shapes if applicable)	COSEG accuracy
[Kalogerakis et al. 2010]	supervised	labeled shapes	9.4% (19) / 14.8% (3)	95.3% (19) / 89.2% (3)	91.9% (12) / 89.0% (3)
[Benhabiles et al. 2011]	supervised	segmented shapes	8.8% (19) / 9.7% (6)	not applicable	not applicable
[Huang et al. 2011]	unsupervised	none	10.1%	not applicable	not applicable
[Sidi et al. 2011]	unsupervised	none	unknown	unknown	87.7%
[van Kaick et al. 2011a]	supervised	labeled shapes	unknown	88.7% (12), see caption	unknown
[Hu et al. 2012a]	unsupervised	none	unknown	88.5%	91.4%
[Lv et al. 2012]	semi-supervised	labeled shapes	unknown	92.3% (3)	unknown
[Wang et al. 2012]	semi-supervised	link constraints	unknown	unknown	‘close to error-free’
[Wang et al. 2013b]	supervised	labeled images	unknown	88.0% (19), see caption	unknown
[Kim et al. 2013a]	semi-/unsupervised	box templates	unknown	unknown	92.7% (semi-superv.)
[Huang et al. 2014b]	unsupervised	none	unknown	unknown	90.1%
[Xu et al. 2014b]	supervised	labeled shapes	10.0%	86.0%	unknown
[Xie et al. 2014]	supervised	labeled shapes	10.2% (19)	94.2 (19) / 88.6 (5)	unknown
[Yi et al. 2016]	semi-supervised	labeled views and verifications	unknown	unknown	100% human-verified

Table 2: Performance of data-driven methods for segmentation in the Princeton Segmentation Benchmark (PSB) and COSEG datasets. Left to right: segmentation method, learning type depending on the nature of data required as input to the method, type of manual input if such required, segmentation performance expressed by the rand index metric [Chen et al. 2009], labeling accuracy [Kalogerakis et al. 2010] based on the PSB and COSEG datasets. We report the rand index segmentation error metric averaged over all classes of the PSB benchmark. The labeling accuracy is averaged over the Labeled PSB (L-PSB) benchmark excluding the “Bust”, “Mech”, and “Bearing” classes. The reason is that there are no clear semantic correspondences between parts in these classes, or the ground-truth segmentations do not sufficiently capture semantic parts in their shapes. We report the labeling accuracy averaged over the categories of the COSEG dataset used in [Sidi et al. 2011]. The COSEG classes “iron”, “large chairs”, “large vases”, “tele-aliens” were added later and are excluded here since most papers frequently do not report performance in those. We note that van Kaick et al. [van Kaick et al. 2011a] reported the labeling accuracy in ten of the L-PSB classes, while Wang et al. [Wang et al. 2013b] reported the labeling accuracy in seven of the L-PSB classes. The method by Kim et al. [Kim et al. 2013a] can run in either semi-supervised or unsupervised mode. In unsupervised mode, the corresponding labeling accuracy is 89.9% in the COSEG dataset on average.

of ML or MAP learning and the non-linearity of classifiers used in shape segmentation, it is common to train the parameters θ_1 and θ_2 of the model separately i.e., train the classifiers of the unary and pairwise term separately [Sutton and Mccallum 2005]. The exact form of the unary and pairwise terms vary across supervised shape segmentation methods: the unary term can have the form of a log-linear model [Anguelov et al. 2005c], cascade of JointBoost classifiers [Kalogerakis et al. 2010], Gentleboost [van Kaick et al. 2011a], or feedforward neural networks [Xie et al. 2014]. The pairwise term can have the form of a learned log-linear model [Anguelov et al. 2005c], label-dependent GentleBoost classifier [Kalogerakis et al. 2010], or a smoothness term based on dihedral angles and edge length tuned by experimentation [Shapira et al. 2010; van Kaick et al. 2011a; Xie et al. 2014]. Again the form of the unary and pairwise terms depend on the amount of training data, dimensionality and underlying distribution of geometric features used, and computational cost.

Joint labeling. Instead of applying the learned probabilistic model to a single shape, an alternative approach is to find correspondences between faces of pairs of shapes, and incorporate a third “inter-shape” term in the energy of Equation 2 [van Kaick et al. 2011a]. The “inter-shape” term favors pairs of corresponding faces on different shapes to have the same label. As a result, the energy can be minimized jointly over a set of shapes to take into account any additional correspondences.

Boundary learning. Instead of applying a classifier per mesh point, face or patch to predict a part label, a different approach is to predict the probability that a polygon mesh edge is a segmentation boundary [Benhabiles et al. 2011]. The problem can be formulated as a binary classifier (e.g., Adaboost) that is trained from human segmentation boundaries. The input to the classifier are geometric features of edges, such as dihedral angles, curvature, and shape diameter. The output is a probability for an edge to be a segmen-

tion boundary. Since the predicted probabilities over the mesh do not correspond to closed smooth boundaries, a thinning and an active contour model [Kass et al. 1988] are used in post-processing to produce the final segmentations.

Transductive segmentation. Another way to formulate the shape segmentation problem is to group patches on a mesh such that the segment similarity is maximized between the resulting segments and the provided segments in the training database. The segment similarity can be measured as the reconstruction cost of the resulting segment from the training ones. The grouping of patches can be solved as an integer programming problem [Xu et al. 2014b].

Shape segmentation from labeled images. Instead of using labeled training shapes for supervised shape segmentation, an alternative source of training data can come in the form of segmented and labeled images, as demonstrated by Wang et al. [Wang et al. 2013b]. Given an input 3D shape, this method first renders 2D binary images of it from different viewpoints. Each binary image is used to retrieve multiple segmented and labeled training images from an input database based on a bi-class Hausdorff distance measure. Each retrieved image is used to perform label transfer to the 2D shape projections. All labeled projections are then back-projected onto the input 3D model to compute a labeling probability map. The energy function for segmentation is formulated by using this probability map in the unary term expressed per face or point, while dihedral angles and Euclidean distances are used in the pairwise term.

4.2 Semi-supervised shape segmentation

Entropy regularization. The parameters θ of Equation 2 can be learned not only from the training labeled shapes, but also from the unlabeled shapes [Lv et al. 2012]. The idea is that learning should maximize the likelihood function of the parameters over the labeled

Segmentation method	Reported running times	Dataset size for reported running times	Reported processor
[Kalogerakis et al. 2010]	8h train. / 5 min test.	6 train. shapes / 1 test shape	Intel Xeon E5355 2.66GHz
[Benhabiles et al. 2011]	10 min train. / 1 min test.	unknown for train. / 1 test shape	Intel Core 2 Duo 2.99GHz
[Huang et al. 2011]	32h	380 shapes	unknown, 2.4 GHz
[Sidi et al. 2011]	10 min	30 shapes	AMD Opteron 2.4GHz
[van Kaick et al. 2011a]	10h train. / few min test.	20-30 train. shapes / 1 test shape	AMD Opteron 1GHz
[Hu et al. 2012a]	8 min (excl. feat. extr.)	20 shapes	Intel dual-core 2.93GHz
[Lv et al. 2012]	7h train. / few min test.	20 shapes	Intel I7 2600 3.4GHz
[Wang et al. 2012]	7 min user interaction	28 shapes	unknown
[Wang et al. 2013b]	1.5 min (no train. step)	1 test shape	unknown
[Kim et al. 2013a]	11h	7442 shapes	unknown
[Huang et al. 2014b]	33h	8401 shapes	unknown, 3.2GHz
[Xu et al. 2014b]	30 sec (no train. step)	1 test shape	Intel i5 CPU
[Xie et al. 2014]	15 sec train. (excl. feat. extr.)	6 train. shapes	Intel Quad-Core 3.2 GHz

Table 3: Running times reported for the data-driven segmentation methods of Table 2. We note that running times are reported in different dataset sizes and processors in the referenced papers, while it is frequently not specified whether the execution uses one or multiple threads or whether the running times include all the algorithm steps, such as super-face or feature extraction. Exact processor information is also frequently not provided. Thus, the reported running times of this table are only indicative and should not serve as a basis for a fair comparison.

shapes, and also minimize the entropy (uncertainty) of the classifier over the unlabeled shapes (or correspondingly maximize the negative entropy). The idea is that minimizing the entropy over unlabeled shapes encourages the algorithm to find putative labelings for the unlabeled data [Jiao et al. 2006]. However, it is generally hard to strike a balance between the likelihood and entropy terms.

Metric embedding and active learning. A more general formulation for semi-supervised segmentation was presented in [Wang et al. 2012]. Starting from a set of shapes that are co-segmented in an unsupervised manner [Sidi et al. 2011], the user interactively adds two types of constraints: “must-link” constraints, which specify that two patches (super-faces) should belong to the same cluster, and “cannot-link” constraints which specify that two patches must be in different clusters. These constraints are used to perform constrained clustering in an embedded feature space of super-faces coming from all the shapes of the input dataset. The key idea is to transform the original feature space, such that super-faces with “must-link” constraints come closer together to form a cluster in the embedded feature space, while super-faces with “cannot-link” constraints move away from each other. To minimize the effort required from the user, the method suggests to the user pairs of points in feature space that when constrained are likely to improve the co-segmentation. The suggestions involve points that are far from their cluster centers, and have a low confidence of belonging to their clusters. **Yi et al. [Yi et al. 2016] propose a method for achieving accurate segmentation of a dataset. Their method balances between manual mesh labeling, automatic propagation of these annotations, and verification of manual and automatic annotations. Their optimization explicitly minimizes the expected human work time.**

Template fitting. A different form of partial supervision can come in the form of part-based templates. Kim et al.’s method [Kim et al. 2013a] allows users to specify or refine a few templates made out of boxes representing expected parts in an input database. The boxes iteratively fit to the shapes of a collection through simultaneous alignment, surface segmentation and point-to-point correspondences estimated between each template and each input shape. Alternatively, the templates can be inferred automatically from the shapes of the input collection without human supervision based on single shape segmentation heuristics. Optionally, the user can refine and improve these estimated templates. From this aspect, Kim et al.’s method can run in either a semi-supervised or unsupervised method. It was also the first method to handle segmentation and

correspondences in collections with size on the order of thousands of shapes.

4.3 Unsupervised segmentation

Unsupervised data-driven shape segmentation techniques fall into two categories: clustering based techniques and matching based techniques. In the following, we highlight the key idea of each type of approach.

Clustering based techniques are adapted from supervised techniques. They compute feature descriptors on points or faces. Clustering is performed over all points/faces over all shapes. Each resulting cluster indicates a consistent segment across the input shapes. The promise of the clustering based approach is that when the number of shapes becomes large, the sampling density in the clustering space becomes dense enough, so that certain statistical assumptions are satisfied, e.g., diffusion distances between points from different clusters is significantly larger than those between points within each cluster. When these assumptions are satisfied, clustering based approaches may produce results that are comparable to supervised techniques (c.f. [Hu et al. 2012a]). In [Sidi et al. 2011], the authors utilize spectral clustering to perform clustering. In [Hu et al. 2012a], the authors employ subspace clustering, a more advanced clustering method, to obtain improved results.

Clustering methods can also be applied to shape parts. In [Xu et al. 2010], the authors perform co-analysis over a set of shapes via factoring out the part scale variation by grouping the shapes into different styles, where style is defined by the anisotropic part scales of the shapes. In [van Kaick et al. 2013], the authors introduce unsupervised co-hierarchical analysis of a set of shapes. They propose a novel cluster-and-select scheme for selecting representative part hierarchies for all shapes and grouping the shapes according to the hierarchies. The method can be used to compute consistent hierarchical segmentations for the input set.

Matching based methods [Golovinskiy and Funkhouser 2009b; Huang et al. 2011; Wang et al. 2013a; Huang et al. 2014b] build maps across shapes and utilize these maps to achieve consistency of segmentations. As shown in Figure 7, this strategy allows us to identify meaningful parts despite the lack of strong geometric cues on a particular shape. Likewise, the approach is able to identify coherent single parts even when the geometry of the individual shape suggests the presence of multiple segments. A challenge here is to find a suitable shape representation so that maps across diverse

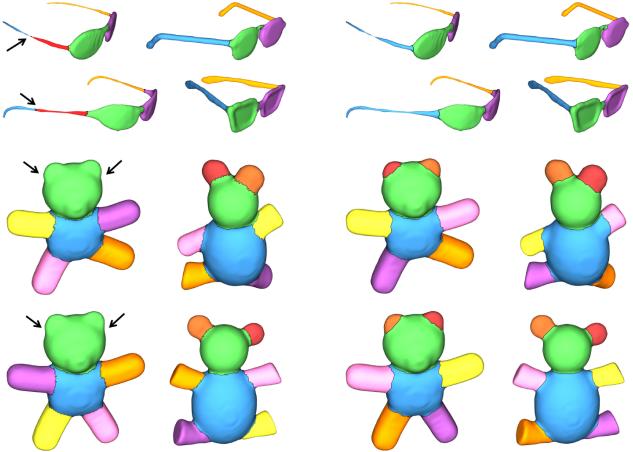


Figure 7: Comparison of single-shape segmentation (left) and joint shape segmentation (right) on models from the PSB benchmark [Chen et al. 2009]. Each segmentation on the left was produced by the top-performing algorithm in the benchmark for that shape. The segmentations on the right were produced by [Huang et al. 2011], which jointly optimized segmentations and correspondences across the entire dataset.

shapes are well-defined. In [Huang et al. 2011], Huang et al. introduce an optimization strategy that jointly optimizes shape segmentations and maps between optimized segmentations. Since the maps are defined at the part-level, this technique is suitable for heterogeneous shape collections. Experimentally, it generates comparable results with supervised method [Kalogerakis et al. 2010] on the Princeton segmentation benchmark. Recently, Huang et al.[Huang et al. 2014b] formulated the same idea under the framework of functional maps [Ovsjanikov et al. 2012a] and gain improved segmentation quality and computational efficiency.

5 Joint shape matching

Another fundamental problem in shape analysis is shape matching, which finds relations or maps between shapes. These maps allow us to transfer information across shapes and aggregate information from a collection of shapes for a better understanding of individual shapes (e.g., detecting shared structures such as skeletons or shape parts). They also provide a powerful platform for comparing shapes (i.e., with respect to different measures and at different places). As we can see from other sections, shape maps are widely applied in shape classification and shape exploration as well.

So far, most existing research in shape matching has focused on matching pairs of shapes in isolation. We refer to [van Kaick et al. 2011c] for a survey and to [Leordeanu and Hebert 2005; Lipman and Funkhouser 2009; van Kaick et al. 2011c; Ovsjanikov et al. 2010; Kim et al. 2011; Ovsjanikov et al. 2012a] for recent advances. Although significant progress has been made, state-of-the-art techniques are limited to shapes that are similar to each other. On the other hand, these techniques tend to be insufficient among shape collections that possess large geometric and topological variations.

The availability of large shape collections offers opportunities to address this issue. Intuitively, when matching two dissimilar shapes, we may utilize intermediate shapes to transfer maps. In other words, we can build maps between similar shapes, and use the composite maps to obtain maps between less similar shapes. As we will see shortly, this intuition can be generalized to enforcing a

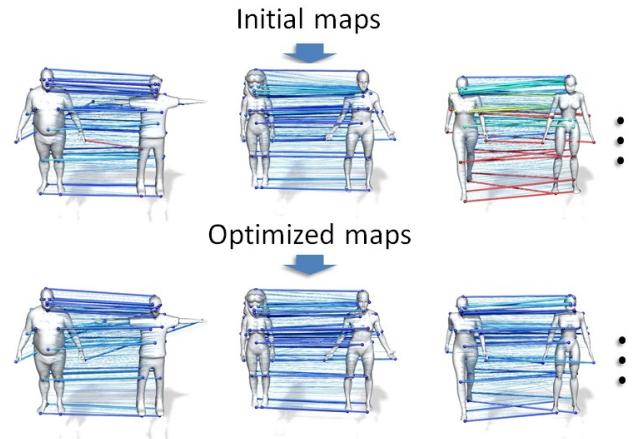


Figure 8: Joint shape matching takes as input maps computed between pairs of shapes in isolation and utilizes the cycle-consistency constraint to improve shape maps. This figure shows the result of Huang et al. [Huang et al. 2014b], which performs joint shape matching under the functional map setting.

cycle-consistency constraint; namely composite maps along cycles should be the identity map and the composite map between any two shapes is path-independent. In this section, we discuss joint shape matching techniques that take a shape collection and noisy maps as input, and output improved maps across the shape collection.

5.1 Model graph and cycle-consistency

To formulate the joint matching problem, we consider a model graph $\mathcal{G} = (\mathcal{S}, \mathcal{E})$ (c.f. [Huber 2002]). The vertex set $\mathcal{S} = \{S_1, \dots, S_n\}$ consists of the input shapes. The edge set \mathcal{E} characterizes the pairs of shapes that are selected for performing pair-wise matching. For small-scale datasets, we typically match all pairs of shapes. For large-scale datasets, the edge set usually connects shapes that are similar according to a pre-defined shape descriptor [Kim et al. 2012a; Huang et al. 2013a], thus generating a sparse shape graph.

The key component of a joint matching algorithm is to utilize the so-called cycle-consistency constraint. In particular, if all the maps in \mathcal{G} are correct, then composite maps along any loops should be the identity map. This is true for maps that are represented as transformations (e.g., rotations and rigid/affine transformations), or full point-wise maps that can be described as permutation matrices. We can easily modify the constraint to handle partial maps; namely, each point, when transformed along a loop, either disappears or goes back to the original point (See [Huang et al. 2014b] for details).

The cycle-consistency constraint is useful because the initial maps, which are computed between pairs of shapes in isolation, are not expected to satisfy the cycle consistency constraint. On the other hand, although we do not know which maps or correspondences are incorrect, we can detect inconsistent cycles. These inconsistent cycles provide useful information for us to detect incorrect correspondences or maps, i.e., an inconsistent cycle indicates that at least one of the participating maps or correspondences is incorrect. To incorporate this observation into algorithms, one has to formulate the cycle-consistency constraint properly. Existing works in data-driven shape matching fall into two categories: combinatorial techniques and matrix recovery based techniques. The reminder of this section provides the details.

5.2 Combinatorial techniques

Spanning tree optimization. Earlier works in joint matching aim at finding a spanning tree in the model graph. In [Goldberg et al. 2004; Huang et al. 2006], the authors propose to use the maximum spanning tree (MST) of the model graph. However, this strategy can easily fail since a single incorrect edge in the MST may break the entire matching result. In the seminal work [Huber 2002], Huber showed that finding the best spanning tree maximizing the number of consistent edges is NP-hard. Although finding the best spanning tree is not tractable, Huber introduced several local operations for improving the score of spanning trees. However, these approaches are generally limited to small-scale problems so that the search space can be sufficiently explored.

Inconsistent cycle detection. Another line of approaches [Zach et al. 2010; Roberts et al. 2011; Nguyen et al. 2011] applies global optimization to select cycle-consistent maps. These approaches are typically formulated as solving constrained optimization problems, where objective functions encode the scores of selected maps, and constraints enforce the consistency of selected maps along cycles. The major advantage of these approaches is that the correct maps are determined globally. However, as the cycle consistency constraint needs to apportion blame along many edges on a cycle, the success of these approaches relies on the assumption that correct maps are dominant in the model graph so that the small number of bad maps can be identified through their participation in many bad cycles.

MRF formulation. Joint matching may also be formulated as solving a second order Markov Random Field (or MRF) [Cho et al. 2010b; Cho et al. 2010a; Crandall et al. 2011; Huang et al. 2012b]. The basic idea is to sample the transformation/deformation space of each shape to obtain a candidate set of transformation/deformation samples per shape. Joint matching is then formulated as optimizing the best sample for each shape. The objective function considers initial maps. Specifically, each pair of samples from two different shapes would generate a candidate map between them. The objective function then formulates second-order potentials, where each term characterize the alignment score between these candidate maps and the initial maps [Huang et al. 2013a; Huang et al. 2012b].

The key challenge in the MRF formulation is generating the candidate samples for each shape. The most popular strategy is to perform uniform sampling [Crandall et al. 2011; Huang et al. 2013a], which works well when the transformation space is low-dimensional. To apply the MRF formulation on high-dimensional problems, Huang et al. [Huang et al. 2012b] introduce a diffusion-and-sharpening strategy. The idea is to diffuse the maps among the model graph to obtain rich samples of candidate transformations or correspondences and then perform clustering to reduce the number of candidate samples.

5.3 Matrix based techniques

A recent trend in map computation is to formulate joint map computation as inferring matrices [Singer and Wu 2011; Huang et al. 2012b; Kim et al. 2012a; Huang and Guibas 2013a; Wang and Singer 2013; Chen et al. 2014c; Huang et al. 2014b]. The basic idea is to consider a big map collection matrix

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_{11} & \mathbf{X}_{12} & \cdots & \mathbf{X}_{1n} \\ \mathbf{X}_{21} & \mathbf{X}_{22} & \cdots & \mathbf{X}_{2n} \\ \vdots & \ddots & \cdots & \vdots \\ \mathbf{X}_{21} & \cdots & \cdots & \mathbf{X}_{nn} \end{bmatrix},$$

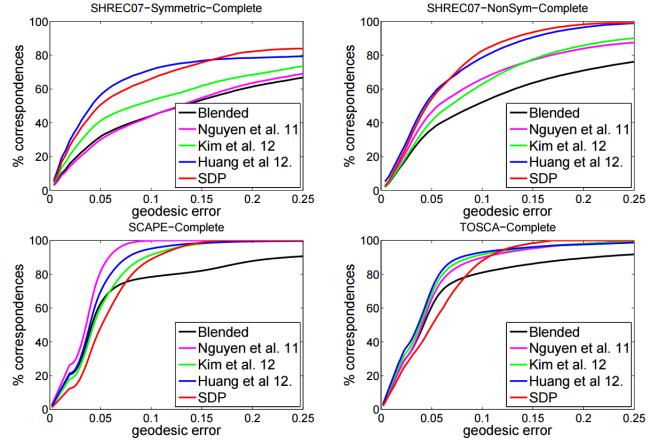


Figure 9: Comparison among various data-driven shape matching methods: optimized composite maps [Nguyen et al. 2011], fuzzy correspondences [Kim et al. 2012a], hub-and-spoke network [Huang et al. 2012b] and semidefinite programming relaxation [Huang and Guibas 2013a]. The input maps are given by blended intrinsic maps [Kim et al. 2011].

where each block \mathbf{X}_{ij} encodes the map from shape S_i to shape S_j . In this matrix representation, the cycle-consistency constraint can be equivalently described by simple properties of \mathbf{X} , i.e., depending on the types of maps, \mathbf{X} is either positive semidefinite or low-rank (c.f. [Huang and Guibas 2013a; Huang et al. 2014b]). In addition, we may view the initial pair-wise maps as noisy measurements of the entries of \mathbf{X} . Based on this perspective, we can formulate joint matching as matrix recovery from noisy measurements of its entries.

Spectral techniques. The initial attempts in matrix recovery are spectral techniques and their variants [Singer and Wu 2011; Kim et al. 2012a; Wang et al. 2013a]. The basic idea is to consider the map collection $\mathbf{X}^{\text{input}}$ that encodes initial maps in its blocks. Then, the recovered matrix is given by $\mathbf{X} = \mathbf{U}\Sigma\mathbf{V}^T$, where \mathbf{U} , Σ , \mathbf{V} represent the singular value decomposition (or SVD) of $\mathbf{X}^{\text{input}}$. Various methods have added heuristics on top of this basic procedure. For example, Kim et al. [Kim et al. 2012a] use the optimized maps to recompute initial maps.

This SVD strategy can be viewed as matrix recovery because \mathbf{X} is equivalent to the optimal low-rank approximation of $\mathbf{X}^{\text{input}}$ (with given rank) under the matrix Frobenius norm. However, as the input maps may contain outliers, employing the Frobenius norm for matrix recovery is sub-optimal. Moreover, it is hard to analyze these techniques, even in the very basic setting where maps are given by permutation matrices [Pachauri et al. 2013].

Point-based maps. In a series of works, Huang and coworkers [Huang and Guibas 2013a; Chen et al. 2014c; Huang et al. 2014a] consider the case of point-based maps and develop joint matching algorithms that admit theoretical guarantees. The work of [Huang and Guibas 2013a] considers the basic setting of permutation matrix maps and proves the equivalence between cycle-consistent maps and the low-rank or positive semi-definiteness of the map collection matrix. This leads to a semidefinite programming formulation for joint matching. In particular, the L1 norm is used to measure the distance between the recovered maps and the initial maps. The authors provide exact recovery conditions, which state that the ground-truth maps can be recovered if the percentage of incorrect correspondences in the input maps is below a constant. In a followup work, Chen et al. [Chen et al. 2014c]

extends this to partial maps and provide a better analysis in the case where incorrect correspondences in the input maps are random. The computational issue is addressed in [Huang et al. 2014a], which employs the alternating direction of multiplier methods for optimization. Figure 9 compares the SDP formulation of [Huang and Guibas 2013a] with several other data-driven shape matching methods. Note that all data-driven shape matching methods improve upon pair-wise matching, yet the SDP formulation leads to the largest improvement.

Rotations and functional maps. Maps that are represented by general matrices (e.g., rotations or functional maps) can also be handled in a similar fashion. In [Wang and Singer 2013], Wang and Singer consider the case of rotations between objects. Their formulation is similar to [Huang and Guibas 2013a] but utilize an L1 Frobenius norm for measuring the distance between initial rotations and recovered rotations. Recently, Huang et al. [Huang et al. 2014b] extend this idea to functional maps. The major difference between functional maps and point-based maps or rotations is that the map collection matrix is no-longer symmetric. Thus, their method is formulated to recover low-rank matrices.

5.4 Discussion and future directions

The key to joint shape matching is to have a proper formulation of the cycle-consistency constraint. We have witnessed the evolution of this constraint from earlier works on combinatorial search and inconsistent cycle detection to more recent works which use spectral techniques, MRF based methods and matrix recovery techniques. In particular, matrix recovery techniques admit theoretical guarantees, providing a fundamental understanding of why joint shape matching can improve upon isolated pair-wise matching.

One future direction is to integrate pair-wise matching and joint matching into one optimization problem. Since the major role of joint matching is to remove the noise presented in pair-wise matching, it makes sense to perform them together. Such unified approaches have the potential to further improve upon decomposed approaches (i.e., from pair-wise to joint). The technical challenge is to find map representations so that pair-wise matching and map consistency can be formulated in the same framework.

6 Shape reconstruction

Reconstructing geometric shapes from physical objects is a fundamental problem in geometry processing. The input to this problem is usually a point cloud produced by aligned range scans, which provides an observation of an object. The goal of a shape reconstruction algorithm is to convert this point cloud into a high-quality geometric model. In practice, the input point cloud data is noisy and incomplete. Thus, the key to a successful shape reconstruction algorithm is formulating appropriate shape priors. Traditional shape reconstruction algorithms usually utilize generic priors, such as surface smoothness [Diebel et al. 2006], and typically assume that the input data captures most of the object’s surface. To handle higher degrees of noise and partiality of the input data, it is important to build structural shape priors.

Data-driven techniques tackle this challenge by leveraging shape collections to learn strong structural priors from similar objects, and use them to reconstruct high-quality 3D models. Existing approaches fall into two categories, based on how they represent the shape priors: *parametric* and *non-parametric*. The former usually builds a low-dimensional parametric representation of the underlying shape space, learning the representation from exemplars and enforcing the parameterization when reconstructing new models. Parametric methods typically require building correspondences

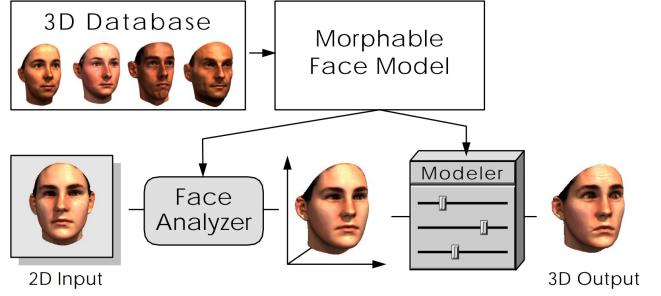


Figure 10: Derived from a dataset of prototypical 3D scans of faces, the morphable face model contributes to two main steps in face manipulation: (1) deriving a 3D face model from a novel image, and (2) modifying shape and texture in a natural way [Blanz and Vetter 1999].

across the exemplar shapes. In contrast, non-parametric methods directly operate on the input shapes by copying and deforming existing shapes or shape parts.

6.1 Parametric methods

Morphable face. The morphable face model [Blanz and Vetter 1999] is a representative work of parametric data-driven shape reconstruction, a technique which reconstructs 3D textured faces from photos and scans. The model is learned from a dataset of prototypical 3D shapes of faces, and can then be used to derive a new 3D face from a novel image. (See Figure 10).

In particular, the morphable face model represents the geometry of a face with a shape-vector $S = (\mathbf{p}_1^T, \dots, \mathbf{p}_n^T)^T \in \mathbb{R}^{3n}$, that contains the 3D coordinates of its n vertices. Similarly, it encodes the texture of a face by a texture-vector $T = (\mathbf{c}_1^T, \mathbf{c}_2^T, \dots, \mathbf{c}_n^T)^T \in \mathbb{R}^{3n}$, that contains the RGB color values of the corresponding vertices. A morphable face model is then constructed using a database of m exemplar faces, each represented by its shape-vector S_i and T_i . In [Blanz and Vetter 1999] the exemplar faces are constructed by matching a template to scanned human faces.

The morphable face model uses Principal Component Analysis (PCA) to characterize the shape space. A new shape and its associated texture are given by

$$\mathbf{S}_{\text{mod}} = \bar{\mathbf{S}} + \sum_{i=1}^{m-1} \alpha_i \mathbf{s}_i, \quad \mathbf{T}_{\text{mod}} = \bar{\mathbf{T}} + \sum_{i=1}^{m-1} \beta_i \mathbf{t}_i,$$

where $\bar{\mathbf{S}}$ and $\bar{\mathbf{T}}$ are the mean-shape and mean-texture, respectively, and \mathbf{s}_i and \mathbf{t}_i are eigenvectors of the covariance matrices. α_i and β_i are coefficients. PCA also gives probability distributions over coefficients. The probability for coefficient α_i is given by

$$p(\{\alpha_i\}) \sim \exp \left(-\frac{1}{2} \sum_{i=1}^{m-1} (\alpha_i / \sigma_i)^2 \right),$$

with σ_i^2 being the corresponding eigenvalue of the shape covariant matrix C_S (the probability $p(\{\beta_i\})$ is computed in a similar way).

With this morphable face model, reconstruction of textured models can be posed as a small-scale non-linear optimization problem. For example, given a 2D image of a human face I_{input} , one can reconstruct the underlying textured 3D model by searching for a similar rendered face $I(\{\alpha_i\}, \{\beta_i\}, p)$, parameterized by the shape and texture coefficients α_i and β_i , and the rendering parameters p .



Figure 11: Parameterizing the variation in human shapes can be used to synthesize new individuals or edit existing ones [Allen et al. 2003].

(e.g., camera configuration, lighting parameters). The optimization problem is formulated as minimizing a data term, which measures the distance between the input image and the rendered image, and regularization terms that are learned from exemplar faces. The success of the morphable model relies on the low-dimensionality of the solution space, and so is also applicable to several other data sets where this assumption holds, including the domain of human bodies and poses.

Morphable human bodies. Allen et al. [Allen et al. 2003] generalize the morphable model to characterize human bodies (Figure 11). Given a set of 250 scanned human bodies, the method first performs non-rigid registration to fit a hole-free, artist-generated mesh (template) to each of these scans. The result is a set of mutually consistent parameterized shapes based on the corresponding vertex positions originating from the template. Similar to [Blanz and Vetter 1999], the method employs PCA to characterize the shape space, which enables applications in shape exploration, synthesis and reconstruction.

In addition to variations in body shape, human models also exhibit variations in poses. The SCAPE model (Shape Completion and Animation for People) [Anguelov et al. 2005a] addresses this challenge by learning two separate models of body deformation – one accounting for variations in poses and one for differences in body shapes. The pose deformation component is acquired from a set of dense 3D scans of a single person in multiple poses. A key aspect of the pose model is that it decomposes deformation into a rigid and a non-rigid component. The rigid component is modeled using a standard skeleton system. The non-rigid component, which captures remaining deformations such as flexing of the muscles, associates each triangle with a local affine transformation matrix. These transformation matrices are learned from exemplars using a joint regression model. In [Hasler et al. 2009b], Hasler et al. introduce a unified model for parameterizing both shapes and poses. The basic idea is to consider the relative transformations between all pairs of neighboring triangles. These transformation matrices allow us to reconstruct the original shape by solving a least squares problem. In this regard, each shape is encoded as a set of edge-wise transformation matrices, which are fit into the PCA framework to obtain a statistical model of human shapes. The model is further extended to estimate shapes of dressed humans from range scans [Hasler et al. 2009a].

Recent works on statistical human shape analysis focus on combining learned shape priors with sparse observations and special effects. In [Loper et al. 2014], the authors introduce an approach that reconstructs high-quality shapes and poses from a sparse set of markers. The success of this approach relies on learning meaningful shape priors from a database consisting of thousands of shapes. In [Tsoli et al. 2014], the authors study human breathing from acquired data.

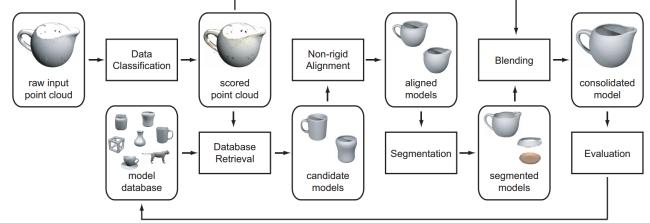


Figure 12: The data-driven shape reconstruction pipeline proposed in [Pauly et al. 2005a].

Data-driven tracking. Object tracking aims to analyze dynamic shapes and/or poses of physical objects. Successful tracking techniques (e.g., [Weise et al. 2009; Weise et al. 2011; Li et al. 2013; Cao et al. 2013; Cao et al. 2014]) typically utilize parametric shape spaces. These reduced shape spaces provide shape priors that improve both the efficiency and robustness of the tracking process. The way to utilize and construct shape spaces vary in different settings, and are typically tailored to the specific problem setting. Weise et al. [Weise et al. 2009] utilize a linear PCA subspace trained with a very large set of pre-processed facial expressions. This method requires an extended training session with a careful choice of facial action units. In addition, the learned face model is actor-specific. These restrictions are partially resolved in [Li et al. 2010], which introduces an example-based blendshape optimization technique, involving only a limited number of random facial expressions. In [Weise et al. 2011], the authors combine both blendshapes and data-driven animation priors to improve the tracking performance. In a recent work, Li et al. [Li et al. 2013] employ adaptive PCA to further improve tracking performance on nuanced emotions and micro-expression. The key idea is to combine a general blendshape PCA model and a corrective PCA model that is updated on-the-fly. This corrective PCA model captures the details of the specific actor and missing deformations from the initial blendshape model. Wei et al. [Wei et al. 2016] train a neural network to predict correspondences between depth scans of humans. They demonstrate that one can train a network on synthetic data using the existing morphable human bodies.

6.2 Non-parametric methods

Parametric methods require canonical domains to characterize the shape space, which have so far been demonstrated within domains of organic shapes, such as body shapes or faces. In this section, we discuss another category of methods that have shown the potential to handle more diverse shape collections.

Generally speaking, a non-parametric data-driven shape reconstruction method utilizes a collection of relevant shapes and combines three phases, i.e., a query phase, a transformation phase and an assembly phase. Existing methods differ in how the input shape collection is preprocessed and how these phases are performed.

Example-based scan completion. Pauly et al. [Pauly et al. 2005a] introduce one of the first non-parametric systems. As shown in [Pauly et al. 2005a], the method takes a point cloud and a collection of complete objects as input. The reconstruction procedure incorporates all three phases described above. The first phase determines a set of similar objects. The retrieval phase combines both text-based search and PCA signatures, which are then refined by rigid alignment. The second step performs non-rigid alignment between the retrieved shapes and the input point cloud. This step partitions the input point cloud into a set of patches, where each patch is associated with one retrieved shape (via the corresponding

region). The final phase merges the corresponding regions into a unified shape.

Nan et al. [Nan et al. 2012] introduce a similar system for indoor scene reconstruction. Given an input point cloud of an indoor scene that consists of a set of objects with known categories, the method searches in a database of 3D models to find matched objects and then deforms them in a non-rigid manner to fit the input point cloud. Note that this method treats complete 3D objects as building blocks, so the final reconstruction does not necessarily reflect the original scene. Shao et al. [Shao et al. 2012] adopt an interactive approach to labeled segmentations of RGBD images, followed by 3D model retrieval for scene modeling. Chen et al. [Chen et al. 2014b] learn contextual relationships from a 3D scene database to further constrain the reconstruction for semantic compatibility between both objects and parts.

In contrast to considering entire 3D shapes, Gal et al. [Gal et al. 2007a] utilize a dictionary of local shape priors (defined as patches) for shape reconstruction. The method is mainly designed for enhancing shape features, where each region of an input point cloud is matched to a shape patch in the database. The matched shape patch is then used to enhance and rectify the local region. Recently, Mattausch et al. [Mattausch et al. 2014] introduce a patch-based reconstruction system for indoor scenes. Their method considers recognizing and fitting planar patches from point cloud data.

Shen et al. [Shen et al. 2012] extends this idea for single object reconstruction by assembling object parts. Their method utilizes a collection consistently segmented 3D shapes. Given a scan of an object, the method recursively searches for parts in the collection to assemble the original object. The retrieval phase considers both the geometric similarity between the input and retrieved parts as well as part compatibility which is learned from the input shapes. Sung et al. [Sung et al. 2015] describe a framework for reliably estimating the part structure of partial scans by treating each part as a local coordinate system. Their method also utilizes symmetric properties of the target object and shape collection, providing more accurate reconstructions on their shape completion benchmark.

Data-driven SLAM. Non-parametric methods have also found applications in reconstructing temporal geometric data (e.g., the output of the Kinect scanner). The Simultaneous localization and mapping (or SLAM) method is a notable technique which jointly estimates the trajectory of the scanning device along side the geometry of the environment. In this case, shape collections serve as priors for the objects in the environment, which can be used to train object detectors. For example, the SLAM++ system proposed by Salas-Moreno et al. [Salas-Moreno et al. 2013] trained domain specific object detectors from shape collections. The learned detectors are integrated inside the SLAM framework to recognize and track those objects. Similarly, Kim et al. [Kim et al. 2012b] use learned object models to reconstruct dense 3D models from a single scan of an indoor scene. More recently, Sun et al. [Song and Xiao 2014] introduced a 3D sliding window object detector with improved performance and capability extending to a broader range of objects. Li et al. [Li et al. 2015a] propose a data-assisted online reconstruction technique which allows object retrieval from a 3D shape database while simultaneously scanning an environment in real-time.

Shape-driven reconstruction from images. Recently, there is a growing interest in reconstructing 3D objects directly from images (e.g., [Xu et al. 2011; Khogade et al. 2014; Aubry et al. 2014; Su et al. 2014]). This problem introduces fundamental challenges in both querying similar objects and deforming objects/parts to fit the input object. In terms of searching similar objects, successful methods typically render objects in the database from a dense

set of viewpoints and pick objects where one view is similar to the input image object. Since depth information is missing from the image object, it is important to properly regularize 3D object transformations; otherwise, a 3D object may be deformed arbitrarily even though its projection on the image domain matches the image object. Most existing techniques consider rigid transformations or user-specified deformations [Xu et al. 2011]. In a recent work, Su et al. [Su et al. 2014] propose to learn meaningful deformations of each shape from its optimal deformation to similar shapes. Huang et al. [Huang et al. 2015b] jointly analyze a large collection of images in object categories and a smaller collection of 3D models to achieve simultaneous analysis and reconstruction of 2D images.

7 Shape modeling and synthesis

So far, the creation of detailed three-dimensional content remains a tedious task which can mainly be performed by skilled artists. 3D content creation has been a major bottleneck hindering the development of ubiquitous 3D graphics. Thus, providing easy-to-use tools for casual and novice users to design and create 3D models has been a key challenge in computer graphics. To address this challenge, current literature has been focused on two main directions, i.e., intelligent interfaces for interactive shape modeling and smart models for automated model synthesis. The former strives to endow modeling interfaces with a higher-level understanding of the structure and semantics of 3D shapes, allowing the interface to reason around the incomplete shapes being modeled. The latter direction focuses on developing data-driven models to synthesize new shapes automatically. The core problem is to learn generative shape models from a set of exemplars (e.g., probability distributions, fitness functions, functional constraints, etc) so that the synthesized shapes are plausible and novel. It can be seen that both of the two paradigms depend on data-driven modeling of shape structures and semantics. With the availability of large 3D shape collections, the data-driven approach becomes a promising answer to the content creation bottleneck.

7.1 Interactive shape modeling and editing

Interactive 3D modeling software (3DS Max, Maya, etc.) provide artists with a large set of tools for creating and editing detailed 3D models. Unfortunately, this same software is often onerous to harness for non-professional users. For casual users, more intuitive modeling interfaces with a certain machine intelligence are to be preferred. Below, we discuss such methods for assembly-based modeling and guided shape editing.

Assembly-based modeling. Early works on 3D modeling based on shape sets are primarily driven by the purpose of *content reuse* in part-assembly based modeling approaches. The seminal work of modeling by example [Funkhouser et al. 2004] presents a pioneering system of shape modeling by searching a shape database for parts to reuse in the construction of new shapes. Kraevoy et al. [2007] describe a system for shape creation via interchanging parts between a small set of compatible shapes. Guo et al. [Guo et al. 2014] propose assembly-based creature modeling guided by a shape grammar.

Beyond content reuse through database queries or hand-crafted rules, Chaudhuri and Koltun [2010] propose a data-driven technique for suggesting the modeler with shape parts that can potentially augment the current shape being built. Such part suggestions are generated through retrieving a shape from a database based on partial shape matching. Although this is a purely geometric method without accounting for the semantics of shape parts, it represents

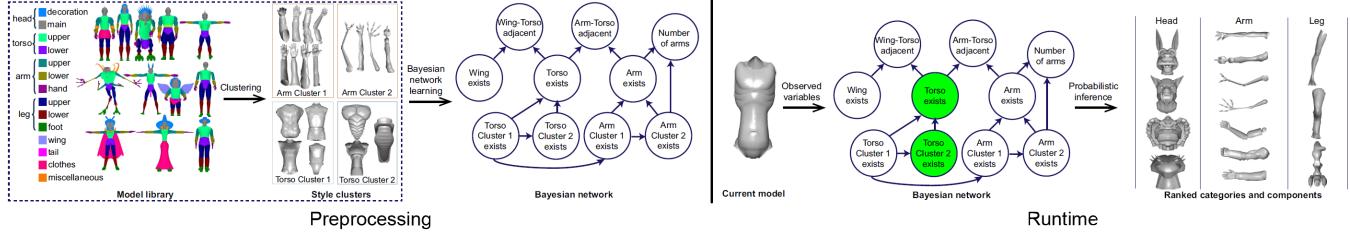


Figure 13: Given a library of models, a Bayesian network encoding semantic and geometric relationships among shape parts is learned [Chaudhuri et al. 2011a] (top). The modeling process (bottom) performs probabilistic inference in the learned Bayesian network to generate ranked lists of category labels and components within each category, customized for the currently assembled model.

the first attempt at utilizing a shape database to *augment the modeling interface*. Later, Chaudhuri et al. [2011a] show that the incorporation of semantic relationships increases the relevance of presented parts. Given a repository of 3D shapes, the method learns a probabilistic graphical model encoding semantic and geometric relationships among shape parts. During modeling, inference in the learned Bayesian network is performed to produce a relevance ranking of the parts.

A common limitation of the above techniques is that they do not provide a way to directly express a high-level design goal (e.g. “create a cute toy”). Chaudhuri et al. [2013] proposed a method that learns semantic attributes for shape parts that reflect the high-level intent people may have for creating content in a domain (e.g. adjectives such as “dangerous”, “scary” or “strong”) and ranks them according to the strength of each learned attribute (Figure 5). During an interactive session, the user explores and modifies the strengths of semantic attributes to generate new part assemblies.

3D shape collections can supply other useful information as well, such as contextual and spatial relationships between shape parts, to enhance a variety of modeling interfaces. Xie et al. [Xie et al. 2013] propose a data-driven sketch-based 3D modeling system. In the off-line learning stage, a shape database is pre-analyzed to extract the contextual information among parts. During the online stage, the user designs a 3D model by progressively sketching its parts and retrieving and assembling shape parts from the database. Both the retrieval and assembly are assisted by precomputed contextual information so that more relevant parts can be returned and selected parts can be automatically placed. Inspired by the ShadowDraw system [Lee et al. 2011], Fan et al. [Fan et al. 2013] propose 3D modeling by drawing with data-driven shadow guidance. The user’s strokes are used to query a 3D shape database for generating the shadow image, which in turn can guide the user’s drawing. Along the drawing, 3D candidate parts are retrieved for assembly-based modeling. Starting from a collection of expertly-created, fabricable 3D models, Schulz et al. [Schulz et al. 2014] extract parameterized design templates encoding all information necessary for fabrication. The templates can then be used to generate new fabricable models in an interactive design system.

Shape editing. The general idea of data-driven shape editing is to learn a model from a collection of closely related shapes that characterizes the plausible variations or deformations of the shapes in this collection. In this way, the learned model can be used to constrain a user’s edit to maintain plausibility. For organic shapes, such as human faces [Blanz and Vetter 1999; Chen et al. 2014a] or bodies [Allen et al. 2003], parametric models can be learned from a shape set characterizing its shape space. Such parametric models can be used to edit the shapes through exploring the shape space with the set of parameters.

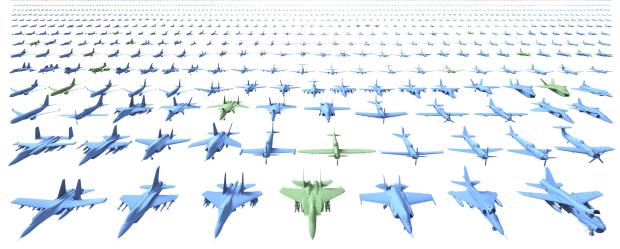


Figure 14: Given a hundred training airplanes (in green), the probabilistic model from [Kalogerakis et al. 2012a] synthesizes several hundreds of new airplanes (in blue).

An alternative widely-adopted approach is the analyze-and-edit paradigm. This technique first extracts the structure of the input shape, and then uses this structure to constrain the editing phase to be more tenable [Gal et al. 2009]. Instead of learning structure from a single shape, Fish et al. [Fish et al. 2014] learn it from a set of shapes that belong to the same family, resulting in a set of probability distributions characterizing the part arrangements. These distributions can be used to guide structure-preserving editing, where models can be edited while maintaining their familial traits. Yumer et al. [Yumer and Kara 2014] extract co-constrained handles from a set of shapes for shape deformation. The handles are generated based on co-abstraction [Yumer and Kara 2012a] of the set of shapes and the deformation co-constraints are learned statistically from the set. The deformation handles can also be controlled with continuous semantic attributes [Yumer et al. 2015]. **This approach also benefit from deep neural network trained to predict appropriate deformations for 3D models** [Yumer and Mitra 2016].

Based on learned structures from a database of 3D models, Xu et al. [Xu et al. 2011] propose photo-inspired 3D object modeling. Guided by the object in a photograph, the method creates a 3D model as a geometric variation of a candidate model retrieved from the database. Due to the pre-analyzed structural information, the method addresses the ill-posed problem of 3D modeling from a single 2D image via structure-preserving 3D warping. The final result is structurally plausible and is readily usable for subsequent editing. Moreover, the resulting 3D model, although built from a single view, is structurally coherent from all views.

7.2 Automated synthesis of shapes

Many applications such as 3D games and films require large collections of 3D shapes for populating their environments. Modeling each shape individually can be tedious even with the best interactive tools. The goal of data-driven shape synthesis algorithms is to generate several shapes automatically with no or very little user su-

pervision: users may only provide some preferences or high-level specifications to control the shape synthesis process. Existing methods achieve this task by using probabilistic generative models of shapes, evolutionary methods, or learned probabilistic grammars.

Statistical models of shapes. The basic idea of these methods is to define a parametric shape space and then fit a probability distribution to the data points that represent the input exemplar shapes. Since the input shapes are assumed to be plausible and desired representatives of the shape space, high-probability areas of the shape space which tend to become associated with new, plausible shape variants. This idea was first explored in the context of parametric models [Blanz and Vetter 1999; Allen et al. 2003], discussed in Section 6. By associating each principal component of the shape space defined by these methods with a Gaussian distribution, this distribution can be sampled to generate new human faces or bodies (Figure 11). Since the probability distribution of plausible shapes tends to be highly non-uniform in several shape classes, Talton et al. [Talton et al. 2009a] use kernel density estimation with Gaussian kernels to represent plausible shape variability. The method is able to generate new shapes for tree and human body parametric spaces.

Shapes have structure i.e., shapes vary in terms of their type and style, different shape styles have different number and type of parts, parts have various sub-parts that can be made of patches, and so on. Thus, to generate shapes in complex domains, it is important to define shape spaces over structural and geometric parameters, and capture hierarchical relationships between these parameters at different levels. Kalogerakis et al. [Kalogerakis et al. 2012a] (Figure 14) proposed a probabilistic model that represents variation and relationships of geometric descriptors and adjacency features for different part styles, as well as variation and relationships of part styles and repetitions for different shape styles. The method learns the model from a set of consistently segmented shapes. Part and shape styles are discovered based on latent variables that capture the underlying modes of shape variability. The method uses a search procedure to assemble new shapes from parts of the input shapes according to the learned probability distribution. Users can also set preferences for generating shapes from a particular shape style, with given part styles or specific parts. Instead of relying on pre-segmented shapes and high-level part descriptors to encode shape variability, Huang et al. [Huang et al. 2015a] propose a probabilistic model that jointly estimates shape segmentation, surface correspondences, and surface descriptors from an input shape dataset. A deep learning procedure was used to capture hierarchical relationships of corresponding surface point positions and parts as well as their existence in the input shapes. Their probabilistic model can be sampled to directly generate point-sampled surface geometry and shape structure.

Set evolution. Xu et al. [Xu et al. 2012] developed a method for generating shapes inspired by the theory of evolution in biology. The basic idea of set evolution is to define cross-over and mutation operators on shapes to perform part warping and part replacement. Starting from an initial generation of shapes with part correspondences and built-in structural information such as inter-part symmetries, these operators are applied to create a new generation of shapes. A selected subset from the generation is presented via a gallery to the user who provides feedback to the system by rating them. The ratings are used to define the fitness function for the evolution. Through the evolution, the set is personalized and populated with shapes that better fit to the user. At the same time, the system explicitly maintains the diversity of the population so as to prevent it from converging into an “elite” set.

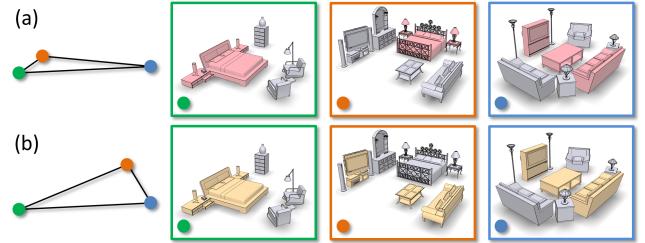


Figure 15: Scene comparisons may yield different similarity distances (left) depending on the focal points [Xu et al. 2014a].

Learned shape grammars. Talton et al. [Talton et al. 2012] leverage techniques from natural language processing to learn probabilistic generative grammars of shapes. The method takes as input a set of exemplar shapes represented with a scene graph specifying parent/child relationships and relative transformations between labeled shape components. They use Bayesian inference to learn a probabilistic formal grammar that can be used to synthesize novel shapes.

8 Scene analysis and synthesis

Analyzing and modeling indoor and outdoor environments has important applications in various domains. For example, in robotics it is desirable for an autonomous agent to understand the semantics of 3D environments to be able to interact with them. In urban planning and architecture, professionals build digital models of cities and buildings to validate and improve their designs. In computer graphics, artists create novel 3D scenes for movies and video games.

The fast growing number of 3D scenes in digital repositories provide new opportunities for data-driven scene analysis, editing, and synthesis. Emerging collections of 3D scenes pose novel research challenges that cannot be easily addressed with existing tools. In particular, representations created for analyzing collections of single models mostly focus on arrangement and relations between shape parts [Mitra et al. 2014], which usually exhibit less variations than objects in scenes. Capturing scene structure poses a greater challenge due to looser spatial relations and a more diverse mixture of functional substructures.

Inferring scene semantics is a long-standing problem in image understanding, with many methods developed for object recognition [Quattoni and Torralba 2009], classification [Swadzba and Wachsmuth 2010], layout and structure reasoning [Choi et al. 2013; Fouhey et al. 2013] with *a single image*. Previous work demonstrates that one can leverage collections of 3D models to facilitate scene understanding in images [Satkin et al. 2012]. In addition, the depth information in RGBD scans can be used to establish the link between 2D and 3D for model-driven scene understanding [Silberman et al. 2012]. The semantic annotations of images are not immediately useful for modeling and synthesizing 3D scenes, for which the geometric and structural priors have to be learned from 3D data.

In this section, we cover the data-driven techniques that leverage collections of 3D scenes for modeling, editing, and synthesizing novel scenes.

Context-based retrieval. To address the large variation in the geometry and arrangement of objects in scenes, Fisher et al. [Fisher and Hanrahan 2010; Fisher et al. 2011] propose to take advantage of local context. One of the key insights of their work is that collec-

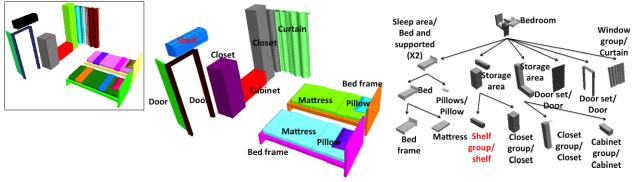


Figure 16: The algorithm processes raw scene graphs with possible over-segmentation (a) into consistent hierarchies capturing semantic and functional groups (b,c) [Liu et al. 2014].

tions of 3D scenes provide rich information about context in which objects appear. They show that capturing these contextual priors can help in scene retrieval and editing.

Their system takes an annotated collection of 3D scenes as input, where each object in a scene is classified. They represent each scene as a graph, where nodes represent objects and edges represent relations between objects, such as support and surface contact. In order to compare scenes, they define kernel functions for pairs of nodes measuring similarity in object geometry, and for pairs of edges, measuring similarity in relations of two pairs of objects. They further define a graph kernel to compare pairs of scenes. In particular, they compare all walks of fixed length originating at all pairs of objects in both scene graphs, which loosely captures similarities of all contexts in which objects appear [Fisher et al. 2011]. They show that this similarity metric can be used to retrieve scenes. By comparing only paths originated at a particular object, they can retrieve objects for interactive scene editing.

Focal points. Measuring the similarity of complex hybrid scenes such as studios composed of a bedroom, living room, and dining room poses a challenge to graph kernel techniques since they only measure global scene similarity. Thus, Xu et al. [2014a] advocate analyzing salient sub-scenes, which they call focal points, to compare hybrid scenes, i.e., scenes containing multiple salient sub-scenes. Figure 15 shows an example of comparing complex scenes, where the middle scene is a hybrid one encompassing two semantically salient sub-scenes, i.e., bed-nightstands and TV-table-sofa. The middle scene is closer to the left one when the bed and nightstands are focused on, and otherwise when the TV-table-sofa combo is the focal point. Therefore, scene comparison may yield different similarity distances depending on the focal points.

Formally, a focal point is defined as a representative substructure of a scene which can characterize a semantic scene category. That means the substructure should re-occur frequently only within that category. Therefore, focal point detection is naturally coupled with the identification of scene categories via scene clustering. This poses coupled problems of detecting focal points based on scene groups and grouping scenes based on focal points. These two problems are solved via interleaved optimization which alternates between focal point detection and focal-based scene clustering. The former is achieved by mining frequent substructures and the latter uses subspace clustering, where scene distances are defined in a focal-centric manner. Inspired by the work of Fisher et al. [Fisher et al. 2011], scene distances are computed using focal-centric graph kernels which are estimated from walks originating from representative focal points.

The detected focal points can be used to organize the scene collection and to support efficient exploration of the collection (see Section 9). Focal-based scene similarity can be used for novel applications such as multi-query scene retrieval, where one may issue queries consisting of multiple semantically related scenes and wish

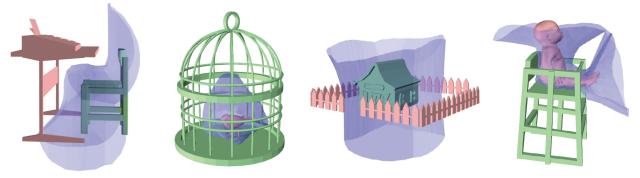


Figure 17: The interaction bisector surface (in blue) of several two-object scenes [Zhao et al. 2014].

to retrieve more scenes “of the same kind”.

Synthesis. Given an annotated scene collection, one can also synthesize new scenes that have a similar distribution of objects. The scene synthesis technique of Fisher et al. [2012] learns two probabilistic models from the training dataset: (1) object occurrence, indicating which objects should be placed in the scene, and (2) layout optimization, indicating where to place the objects. Next, it takes an example scene, and then synthesizes similar scenes using the learned priors. It replaces or adds new objects using context-based retrieval techniques, and then optimizes for object placement based on learned object-to-object spatial relations. Synthesizing example scenes might be a challenging task, thus Xu et al. [2013a] propose modeling 3D indoor scenes from 2D sketches, by leveraging a database of 3D scenes. Their system jointly optimizes for sketch-guided co-retrieval and co-placement of all objects.

Hierarchical scene annotation. All aforementioned applications take an annotated collection of 3D scenes as an input. Unfortunately, most scenes in public repositories are not annotated and thus require additional manual labeling [Fisher et al. 2012]. Liu et al. [2014] address the challenge of annotating novel scenes. The key observation of their work is that understanding hierarchical structure of a scene enables efficient encoding of functional scene substructures, which significantly simplifies detecting objects and representing their relationships. Thus, they propose a supervised learning approach to estimate a hierarchical structure for novel scenes. Given a collection of scene graphs with consistent hierarchies and labels, they train a probabilistic hierarchical grammar encoding the distributions of shapes, cardinalities, and spatial relationships between objects. Such a grammar can then be used to parse new scenes: find segmentations, object labels, and hierarchical organization of objects consistent with the annotated collection (see Figure 16).

Challenges and opportunities. The topic of 3D scene analysis is quite new and there are many open problems and research opportunities. *The first* problem is to efficiently characterize spatial relationships between objects and object groups. Most existing methods work with bounding box representation which are efficient to process, but not sufficiently informative to characterize object-to-object relationships. For example, one cannot reliably determine the object enclosure relationship based on a bounding box. Recently, He et al. [2014] propose to use biologically-inspired bisector surface to characterize the geometric interaction between adjacent objects and to index 3D scenes (Figure 17). The bisector surface can be extended into a geometric descriptor for contextual modeling of the functionality of a 3D object in a given scene [Hu et al. 2015]. *Second*, most existing techniques heavily rely on expert user supervision for scene understanding. Unfortunately, online repositories rarely have models with reliable object tags. Therefore there is a need for methods that can leverage scenes containing only partial and/or noisy annotations. *Finally*, the popularity of commodity RGBD cameras has significantly simplified the acquisition of in-

door scenes. This emerging scanning technique opens space for new applications such as online scene analysis [Zhang et al. 2014; Xu et al. 2015].

9 Exploration and organization

The rapidly growing quantity and variety of digital 3D models in large online collections have caused an emerging need to develop algorithms and techniques that effectively organize these large collections and allow users to interactively explore them. For example, an architect might furnish a digital building by searching through databases organized according to furniture types, regions of interest and design styles. Likewise, an industrial designer can explore shape variations among existing products when creating a new object. Most existing repositories only support text-based search, relying on user-entered tags and titles. This approach suffers from inaccurate and ambiguous tags, often entered in different languages. While it is possible to try using shape analysis to infer consistent tags as discussed in Section 3, it is difficult to convey stylistic and geometric variations using only text. An alternative approach can be to perform shape, sketch, or image based queries. However, to formulate such search queries the user needs to have a clear mental model of the shape that should be retrieved. Thus, some researchers focus on providing tools for *exploring* shape collections. Unlike search, exploration techniques do not assume a-priori knowledge of the repository content, and help the user to understand geometric, topological, and semantic variations within the collection.

Problem statement and method categorization. Data exploration and organization is a classical problem in data analysis and visualization [Paulovich et al. 2011]. Given a data collection, the research focuses on *grouping and relating data points, learning the data variations in the collection, and organizing the collection into a structured form*, to facilitate retrieval, browsing, summarization, and visualization of the data, based on efficient *interfaces or metaphors*.

The first step to organizing model collections is to devise appropriate metrics to relate different data points. Various similarity metrics have been proposed in the past to relate entire shapes as well as local regions on shapes. In particular, previous sections of this document cover algorithms for computing global shape similarities (Section 3), part-wise correspondences (Section 4), and point-wise correspondences (Section 5). In this section, we will focus on techniques that take advantage of these correlations to provide different interfaces for exploring and understanding geometric variability in collections of 3D shapes. We categorize the existing exploration approaches based on four aspects:

- **Metaphor:** a user interface for exploring shape variations. We will discuss five basic exploration interfaces, ones that use proxy shapes (templates), regions of interest, probability plots, query shapes, or continuous attributes.
- **Shape comparison:** techniques used to relate different shapes. We will discuss techniques that use global shape similarities, as well as part or point correspondences.
- **Variability:** shape variations captured by the system. Most methods we will discuss rely on geometric variability of shapes or parts. Some techniques also take advantage of topological variability; that is, variance in number of parts or how they are connected (or variance in numbers of objects and their arrangements in scenes).
- **Organizational form:** a method to group shapes. We will discuss methods that group similar shapes to facilitate explor-

Method	Meta.	Comp.	Var.	Org.
[Ovsjanikov et al. 2011]	temp.	simi.	geom.	n/a
[Kim et al. 2013a]	temp.	part	both	cluster
[Averkiou et al. 2014]	plot	part	both	cluster
[Kim et al. 2012a]	ROI	point	both	n/a
[Rustamov et al. 2013]	ROI	point	geom.	n/a
[Huang et al. 2014b]	ROI	point	both	cluster
[Xu et al. 2014a]	ROI	simi.	topo.	cluster
[Fish et al. 2014]	plot	part	geom.	cluster
[Huang et al. 2013b]	query	simi.	both	hierarchy

Table 4: A summary of several recent works over four aspects. *Metaphor*: templates, surface painted ROIs, probability distribution plots, or query shapes. *Shape Comparison*: shape similarity, part or point correspondence. *Variability*: geometry, topology or both. *Organization Form*: cluster or hierarchy.

ing intra-group similarities and inter-group variations, typically including clustering and hierarchical clustering.

Table 4 summarizes several representative works in terms of these aspects. In the remaining part of this section we list several recent techniques which are grouped based on the exploration metaphor.

Template-based exploration. Component-wise variability in position and scale of parts reveals useful information about a model collection. Several techniques use box-like templates to show variations among models of the same class. Ovsjanikov et al. [Ovsjanikov et al. 2011] describe a technique for learning these part-wise variations without solving the challenging problem of consistent segmentation. First, they use the segmentation of a single shape to construct the initial template. This is the only step that needs to be verified and potentially fixed by the user. The next goal is to automatically infer deformations of the template that would capture the most important geometric variations of other models in the collection. They hypothesize that all shapes can be projected on a low-dimensional manifold based on their global shape descriptors. Finally, they reveal the manifold structure by deforming a template to fit to the sample points. Directions for interesting variations are depicted by arrows on the template and the shapes that correspond to the current template configuration are presented to the user.

The descriptor-based approach described above assumes that all intra-class shapes share the same parts and that there exists a low-dimensional manifold that can be captured by deforming a single template. These assumptions do not hold for large and diverse collections of 3D models. To tackle this challenge, Kim et al. [Kim et al. 2013a] proposed an algorithm for learning several part-based templates capturing multi-modal variability in collections of shapes. They start with an initial template that includes a super-set of all parts that might occur in a dataset, and jointly learn part segmentations, point-to-point surface correspondence as well as a compact deformation model. The output is *a set of templates* that groups the input models into clusters, capturing their styles and variations.

ROI-based exploration. Not all interesting variations occur at the scale of parts: they can occur at sub-part scale, or span multiple sub-regions from multiple parts. In these cases the user may prefer to select an arbitrary region on a 3D model and look for more models sharing similar regions of interest. Such detailed and flexible queries require a finer understanding of correspondences between different shapes. Kim et al. [Kim et al. 2012a] propose fuzzy point correspondences to encode the inherent ambiguity in relating diverse shapes. Fuzzy point correspondences are represented



Figure 18: Shape exploration based on fuzzy correspondence. The user paints a region of interest (ROI) on a query shape (left column), and the method sorts models based on their similarity within the region (right).

by real values specified for all pairs of points, indicating how well the points correspond. They leverage transitivity in correspondence relationships to compute this representation from a sparse set of pairwise point correspondences. The interface proposed by Kim et al. allows users to paint regions of interest directly on a surface and then retrieve similar regions among other shapes, or even show geometric variations found in the selected region (see Figure 18).

One limitation of correspondence-based techniques is that they typically do not consider the entire collection when estimating shape differences. Rustamov et al. [Rustamov et al. 2013] focus on a fundamental intrinsic representation for shape differences. Starting with a functional map between two shapes, that is, a map that describes a change of functional basis, they derive a shape difference operator revealing detailed information about the location, type, and magnitude of distortions induced by a map. This makes shape difference a quantifiable object that can be co-analyzed within a context of the entire collection. They show that this deeper understanding of shape differences can help in exploration. For example, one can embed shapes in a low-dimensional space based on shape differences, or use shape difference to interpolate variations by showing “intermediate” shapes between two regions of interest. To extend these techniques to man-made objects, Huang et al. [Huang et al. 2014b] construct a consistent functional basis for shape collections exhibiting large geometric and topological variability. They show that the resulting consistent maps are capable of capturing discrete topological variability, such as variance in the number of bars on the back of a chair.

ROI-based scene exploration. Recent works on organizing and exploring 3D visual data mostly focus on object collections. Exploring 3D scenes poses additional challenges since scenes typically exhibit more structural variations. Unlike man-made objects that usually contain a handful of object parts, scenes can contain anywhere from ten to hundreds of objects. Not only this, but the objects themselves do not typically have a prescribed rigid arrangement with respect to each other. Thus, global scene similarity metrics, such as the graph kernel based one used in [Fisher et al. 2012] are limited to organizing datasets based on very high-level features, such as scene type. Xu et al. [Xu et al. 2014a] advocate that 3D scenes should be compared from the *perspective* of a particular focal point which is a representative substructure of a specific scene category. Focal points are detected through contextual analysis of a collection of scenes, resulting in a clustering of the scene collection where each cluster is characterized by its representative focal points (see Section 8). Consequently, the focal points extracted from a scene collection can be used to organize collection into an

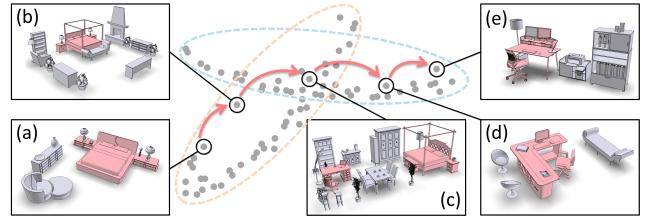


Figure 19: Focal-based scene clustering produces overlapping clusters, which is due to hybrid scenes possessing multiple focal points. An exploratory path, from (a) to (e), through the overlap, smoothly transit between the two scene clusters, representing bedroom and offices, respectively.

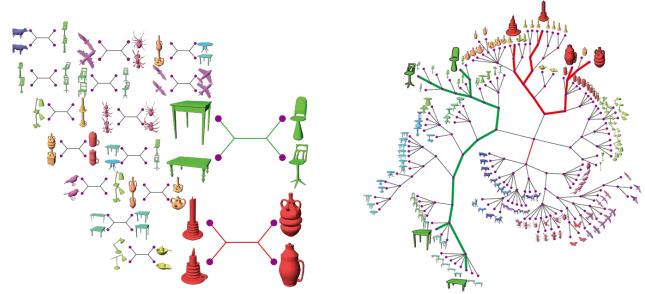


Figure 20: Given a set of heterogeneous shapes, a reliable qualitative similarity is derived from quartets composed of two pairs of objects (left). Aggregating such qualitative information from many quartets computed across the whole set leads to a categorization tree as a hierarchical organization of the input shape collection (right).

interlinked and well-connected cluster formation, which facilitates scene exploration. Figure 19 shows an illustration of such cluster-based organization and an exploratory path transiting between two scene clusters/categories.

Plot-based exploration. All aforementioned exploration techniques typically do not visualize the probabilistic nature of shape variations. Fish et al. [Fish et al. 2014] study the configurations of shape parts from a probabilistic perspective, trying to indicate which shape variations are more likely to occur. To learn the distributions of part arrangements, all shapes in the family are pre-segmented consistently. The resulting set of probability density functions (PDFs) characterizes the variability of relations and arrangements across different parts. A peak in a PDF curve represents that particular a configuration of the related parts frequently appeared among several shapes in the family. The multiple PDFs can be used as interfaces to interactively explore the shape family from various perspectives. Averkiou et al. [Averkiou et al. 2014] use part structure inferred by this method to produce a low-dimensional part-aware embedding of all models. The user can explore interesting variations in part arrangements simply by moving the mouse over the 2D embedding. In addition, their technique allows the synthesis of novel shapes by clicking on empty spaces in the embedded space. Upon clicking, the system would deform parts from neighboring shapes to synthesize a novel part arrangement.

Query-based exploration. For a heterogeneous shape collection encompassing diverse object classes, it is typically not possible to characterize part-structure and correspondences between all pairs of shapes. Even global shape similarity is not a very reliable feature in

this setting, which makes organizing and exploring heterogeneous collections especially difficult. To address this challenge, Huang et al. [Huang et al. 2013b] introduce qualitative analysis techniques from the field of bioinformatics. Instead of relying on quantitative distances, which may be ill-applied between dissimilar shapes, the method considers a more reliable *qualitative similarity* derived from *quartets* composed of two pairs of objects. The shapes that are paired in the quartet are close to each other and far from the shapes in the other pair, where distances are estimated from multiple shape descriptors. They aggregate this topological information from many quartets computed across the entire shape collection, and construct a hierarchical *categorization tree* (see Figure 20). Analogous to the phylogenetic trees of species, this categorization tree of a shape collection provides an overview of the shapes as well as their mutual distance and hierarchical relations. Based on such an organization, they also define the degree of separation chart for every shape in the collection and apply it for interactive shape exploration.

Attribute-based exploration. Yet another approach seeks to allow users to interactively explore shapes with continuously valued semantic attributes. Blanz and Vetter [Blanz and Vetter 1999] provide an interface to explore faces based on continuous facial attributes, such as “smile” or “frown”, built upon the face parametric model (Section 6). Similarly, Allen et al. [Allen et al. 2003] allow users to explore the range of human bodies with features such as height, weight, and age. Chaudhuri et al.’s [Chaudhuri et al. 2013] interface enables exploration of shape parts according to learned strengths of semantic attributes (Figure 5).

10 Discussion

There is no “magic recipe” for developing new data-driven and machine learning applications in geometry processing and computer graphics. Yet, there are some important considerations one needs to make in devising a data-driven method, including computational complexity, scalability, applicability issues, proper evaluation procedures and limitations. In this section, we briefly discuss these issues.

Computational complexity. As explained in Section 2, data-driven shape analysis and processing algorithms generally contain several stages (Figure 2). The complexity of each stage varies and largely depends on the number of input shapes, resolution of the input shape representation (number of faces, surface points, pixels or voxels), as well as the number and type of the used geometric features. The feature extraction stage is usually executed per each shape, thus, its time complexity often tends to be linear in the number of input shapes. Local geometric features, such as surface curvature or PCA-based descriptors, are usually computed within a small neighborhood around each vertex, face, or surface point, thus their extraction depends linearly on the number of these primitives in the input shape representation. Extracting geometric features that capture less local or global information about the shape, such as shape diameter, geodesic distance-based features or heat-kernel descriptors, is often computationally more intensive i.e., super-linear in the number of primitives.

During the learning and inference steps, data-driven methods usually solve an optimization problem, which involves minimizing or maximizing a function e.g., a data likelihood function. In general, optimization techniques inhabit a wide range of computational complexities. For example, if the optimization involves the least-squares solution of a linear system, as in the case of linear regression, the complexity is $O(N \cdot F^2)$, where N is the number of input

training examples and F is the dimensionality of the input feature vector. If optimization is performed through a steepest descent algorithm, the complexity is $O(N \cdot F)$ per parameter update step. However, the performance of iterative optimization algorithms depends on the number of steps, which in turn varies according to their convergence properties and the function they optimize. We refer the reader to [Nocedal and Wright 2006; Koller and Friedman 2009b; Solomon 2015] for an in-depth discussion on the computational complexity and convergence properties of various optimization and inference algorithms.

Scalability. Data-driven methods are inherently bound up with the input data. Rapid developments in capturing and modeling techniques have engendered the growth of 3D shape and scene repositories over recent years, which have in turn influenced the advancement of data-driven geometry processing. This is evidenced by the fact that the number of training shapes employed in data-driven geometry processing techniques has grown from a few tens [Kalogerakis et al. 2010; Sidi et al. 2011] to several thousands [Kim et al. 2013a; Huang et al. 2014b]. On the one hand, the increasing availability of 3D data can improve the accuracy and generalization of data-driven methods. On the other hand, issues of scalability arise. More data causes longer processing times, which in turn makes the debugging of such methods harder for developers. The scalability issues are further exacerbated by the complexity (i.e., high-dimensionality) of the 3D geometric feature representations. Potential workarounds include debugging the pipeline of these methods on smaller datasets before turning to larger ones, trying simpler learning techniques before switching to more complex ones, or making use of computing clusters for executing offline steps.

Scope of application. Not every problem in shape analysis and processing is well suited to be solved by a data-driven method. When the underlying rules, principles, and parameters can be manually and unambiguously specified in a problem, then non-data-driven methods should be considered for it. For example, deforming a shape with an elastic material and known physical parameters and forces can be addressed by a physics-based method rather than a data-driven one. In contrast, there are several problems in shape analysis and processing for which it is hard, or even impossible, to hand-design a set of rules and principles, or quantify them through manually specified parameters. This is often the case for problems that involve shape and scene recognition, high-level processing, structure parsing, co-analysis, reconstruction from noisy missing data, and modeling with high-level user input. Shape co-analysis (e.g., co-segmentation), in particular, requires estimating several possible geometric and semantic correlations among input shapes, which would be practically impossible to capture through hand-designed rules. Data-driven methods that automatically compute geometric, semantic and structural relationships in the input shapes are more appropriate for such co-analysis problems [Xu et al. 2010; Huang et al. 2011; Huang and Guibas 2013a]. Another example can be found in the problem of shape style analysis. Although humans have an innate sense of style similarity, it is hard to manually quantify geometric criteria for modeling the stylistic similarity of shapes. A style analysis algorithm whose parameters are learned through a data-driven method is much more well-suited to perform this quantification [Liu et al. 2015; Lun et al. 2015].

Evaluation. Correctly evaluating the predictive performance of data-driven methods should be another important consideration for researchers of such methods. A common pitfall is to evaluate the predictive performance of a data-driven method with the same dataset on which it was trained on (e.g., through supervised learning), or a dataset for which any parameters of the method were

manually tuned. The risk here is that the method might not be able to generalize to any other datasets beyond the ones used in training or hand-tuning. A method that simply memorizes the training dataset, or overfits a model to a particular dataset, will obviously perform well there. However, if its performance on other datasets is poor, the method is effectively useless. The best practice is to introduce training and test splits of the input datasets. The models and parameters should then be learned or tuned exclusively on the training portion, and evaluated exclusively on the testing portion. To insure fair-play, it is also necessary that different data-driven methods be compared using the same training and test splits.

Limitations. The data-driven approach to shape analysis and processing is bound by a few limitations that we summarize below. We also discuss potential workarounds to overcome some of these.

- **Generalization guarantees.** It is generally hard to provide any guarantees about the generalization performance of data-driven algorithms. In other words, when a data-driven algorithm makes use of a particular dataset for training, it is often impossible to predict how well it will generalize to other datasets beforehand. Although statistical error bounds can be provided under particular assumptions on data distributions, in particular within the context of the Probably Approximately Correct learning theory [Valiant 1984] or the Bayes decision theory [Fukunaga 1990], these assumptions often cannot be validated in practice.
- **Complexity and scalability.** As discussed above, data-driven methods are computationally intensive in general. The complexity of data-driven methods depends on the number of input training shapes. As a general rule of thumb, the accuracy of data-driven methods improves with more training data. On the other hand, this comes at a higher computational cost during training time.
- **Size of 3D shape datasets.** Despite their recent growth, the size of available 3D shape datasets remains much smaller than those used in computer vision and natural language processing (e.g., image and text datasets). As a result, overfitting remains a common issue with data-driven methods for 3D shape processing. Overfitting occurs when a learned model, or function, captures random error, noise, or patterns specific only to the input training dataset instead of the underlying, correct relationships in the data. It usually occurs when the learned model, or function, is excessively complex, e.g., having an extremely large number of parameters relative to the size of the training dataset. To mitigate this issue, regularization techniques can be used to favor simpler models and functions [Ng 2004; Domingos 2012]. Another promising approach is to use both 3D shapes and 2D images as input to data-driven methods, or in other words, to perform co-analysis of image and shape data. We discuss this issue as one important future research direction for further development in data-driven methods in the next section.
- **Data collection.** Data-driven techniques rely on the availability of data for the particular problem they attempt to solve. Gathering training data for several geometry processing tasks is often not an easy task, especially when human labor is involved to process or annotate geometric data. Although crowdsourcing Internet marketplaces, such as Amazon Mechanical Turk [Amazon 2009], can help gather training data efficiently, online questionnaires and user studies still require careful design, monetary compensation, and participant consistency checks.

- **From data to knowledge.** Data-driven methods put particular emphasis on discovering patterns and models that explain the input data and provide useful insights to the problem being solved. However, these learned patterns and models might not always be readily interpretable i.e., might not correspond to “easy-to-understand” rules. This is a common situation when one treats the internals of the data-driven method (e.g., the learning process) as a “black box” without first trying to understand their exact functionality in detail. In general, interpreting such patterns and models requires significant time and effort.

11 Conclusion and open problems

In this survey, we have so far discussed state-of-the-art on data-driven methods for 3D shape analysis and processing. We also presented the main concepts and methodologies used to develop such methods. We hope that this survey will act as a tutorial that will help researchers develop new data-driven algorithms related to shape analysis and processing. There are several exciting research directions that have not been sufficiently explored so far in our community that we discuss below:

Joint analysis of 2D and 3D data. Generating 3D content from images requires building mappings from 2D to 3D space. Unfortunately, the problem remains largely ill-posed. However, with the help vast quantities of 2D images available on the web, effective priors can be developed to map 2D visual elements or features to 3D shape and scene representations. Indeed, we have in fact seen recent attempts made in this very vein of thought with some success in [Su et al. 2014; Aubry et al. 2014; Li et al. 2015b; Huetting et al. 2015; Su et al. 2015b], which attempts depth estimation through joint analysis over 2D image collections and 3D model databases. We have also seen success of the joint analysis framework in the setting of texture-data with [Yumer et al. 2014], which attempts cosegmentation of textured 3D shapes.

Following this line, it would be interesting to jointly analyze and process multi-modal visual data, including depth scans and videos. The key challenge lies in the integration of heterogeneous information in a unified learning framework.

Better and scalable shape analysis techniques. Many data-driven applications rely on high-quality shape analysis results, particularly shape segmentations and correspondences. We believe it is important to further advance research in both these directions. This includes designing shape analysis techniques for specific data and/or making them scalable to very large datasets, especially recently emerging large-scale richly-annotated repositories [Su et al. 2015c].

From geometry to semantics and vice versa. Several data-driven methods have tried to map 2D and 3D geometric data to high-level concepts, such as shape categories, semantic attributes, or part labels. Gathering relevant training data is a key component in achieving this aim, a task which remains a non-trivial endeavor. Several recent promising works employ crowdsourcing to address this issue [Chen et al. 2009; Chaudhuri et al. 2013; Lun et al. 2015; Liu et al. 2015; Yumer et al. 2015]. Existing methods deal with cases where only a handful of different entities are predicted for input shapes or scenes. Scaling these methods to handle thousands of categories, part labels and other such entities, as well as attaining human-level performance, is an open problem. The opposite direction is also interesting and insufficiently explored: generating shapes and scenes based on high-level specifications such as shape

styles, attributes, or even natural language. Such approaches may even potentially be combined with further diverse inputs, such as sketches and interactive handles, in the shape-generating pipeline. WordsEye [Coyne and Sproat 2001] was an early attempt to bridge this gap, yet requires extensive manual mappings.

Understanding function from geometry. The geometry of a shape is strongly related to its functionality, including the shape's relationship to human activity. Thus, analyzing shapes and scenes requires some understanding of their function. The recent works by Laga et al. [Laga et al. 2013], Kim et al. [Kim et al. 2014] and Hu et al. [Hu et al. 2015] are important examples of data-driven approaches that take into account functional aspects of shapes in the process of their analysis. In addition, data-driven methods can guide the synthesis of shapes that can be manufactured or 3D printed based on given functional specifications; an example of such an attempt is reflected in the work of Schulz et al [Schulz et al. 2014].

Data-driven shape abstractions. It is relatively easy for humans to communicate the essence of shapes with a few lines, sketches, and abstract forms. Developing methods that can build such abstractions automatically has significant applications in shape and scene visualization, artistic rendering, and shape analysis. There are a few data-driven approaches to line drawing [Cole et al. 2008; Kalogerakis et al. 2009; Kalogerakis et al. 2012b], saliency analysis [Chen et al. 2012], surface abstraction [Yumer and Kara 2012a], and viewpoint preferences [Secord et al. 2011] related to this goal. Matching human performance in these tasks is still a largely open problem, while synthesizing and editing shapes using shape abstractions as input remains a significant challenge.

Feature learning. Several shape and scene processing tasks depend on engineering geometric features for points and shapes, as discussed in Section 3. In general, it seems that some features work well in particular settings, but can fail in others. A prevailing issue is that universal geometric descriptors - features that can serve as reliable mid or high level representations ubiquitously across all variety of shapes - do not yet exist.

Recent work in machine learning has demonstrated that powerful feature representations can be learned directly from raw input text and image data with deep architectures [Hinton et al. 2006; Krizhevsky et al. 2012; Zeiler and Fergus 2014]. These architectures are composed of multiple processing layers which learn representations of the input data at multiple levels of abstraction. These data-driven representations are optimized for processing-performance in complex interpretation tasks. Such feature learning for 3D shapes with deep architectures has recently been demonstrated in the context of shape classification [Wu et al. 2015; Su et al. 2015a; Xie et al. 2015; Huang et al. 2015a]. Learning features for performing other complex high-level shape analysis and processing tasks remains an open problem.

Biographical sketches

Kai Xu received his PhD in Computer Science at the National University of Defense Technology (NUDT). He is now a faculty member at NUDT and a visiting associate professor at SIAT. Between 2009 and 2010, he visited Simon Fraser University, supported by the Chinese government. His research interests include geometry processing and geometric modeling, especially methods that utilize large collections of 3D shapes. He is currently serving on the editorial board of Computer and Graphics. He has served on the program committees of SGP 2013–2015, PG 2013–2015 and

GMP.

Vladimir G. Kim is a Research Scientist at Adobe Research. His interests include geometry processing and analysis of shapes and collections of 3D models. He received his PhD in the Computer Science Department at Princeton University in 2013, and was a postdoctoral scholar at Stanford University 2013–2015. Vladimir is a recipient of the Siebel Scholarship and the NSERC Postgraduate Scholarship. He was also on the International Program Committee for SIGGRAPH 2015, SGP 2013–2015, and PG.

Qixing Huang is a research assistant professor at TTI Chicago. He earned his PhD from the Department of Computer Science at Stanford University in 2012. He obtained both his MS and BS degrees in Computer Science from Tsinghua University in 2005 and 2002, respectively. His research interests include data-driven geometry processing and co-analysis of shapes and collections of 3D models using convex optimization techniques. He received a Best Paper Award from SGP 2013 and the Most Cited Paper Award from the journal of Computer-Aided Geometric Design in 2011 and 2012. He has served on the program committees of SGP, PG and GMP.

Niloy Mitra REVISE!

Evangelos Kalogerakis is an assistant professor in computer science at the University of Massachusetts Amherst. His research deals with automated analysis and synthesis of 3D visual content, with particular emphasis on machine learning techniques that learn to perform these tasks by combining data, probabilistic models, and prior knowledge. He obtained his PhD from the University of Toronto in 2010 and BEng from the Technical University of Crete in 2005. He was a postdoctoral researcher at Stanford University from 2010 to 2012. He served on the program committees of EG 2014 and 2015, SGP 2012, 2014 and 2015. His research is supported by NSF and NVidia.

Acknowledgements

We thank Zimo Li for proofreading this survey and the anonymous reviewers for helpful suggestions. Kalogerakis gratefully acknowledges support from NSF (CHS-1422441). Kai Xu is supported by NSFC (61572507, 61202333 and 61532003).

References

- Trimble 3d warehouse.
- AIGER, D., MITRA, N. J., AND COHEN-OR, D. 2008. 4-points congruent sets for robust pairwise surface registration. In *ACM SIGGRAPH 2008 papers*, ACM, SIGGRAPH '08, 85:1–85:10.
- ALEXA, M., COHEN-OR, D., AND LEVIN, D. 2000. As-rigid-as-possible shape interpolation. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, SIGGRAPH '00, 157–164.
- ALLEN, B., CURLESS, B., AND POPOVIĆ, Z. 2003. The space of human body shapes: Reconstruction and parameterization from range scans. *ACM Trans. Graph.* 22, 3.
- ALLIEZ, P., COHEN-STEINER, D., DEVILLERS, O., LÉVY, B., AND DESBRUN, M. 2003. Anisotropic polygonal remeshing. In *ACM SIGGRAPH 2003 Papers*, SIGGRAPH '03, 485–493.

Work	Training data			Feature		Learning model/approach	Learning type	Learning outcome
	Rep.	Preproc.	Scale	Type	Sel.			
[Funkhouser et al. 2005]	Point	No	Thousands	Local	No	SVM classifier	Supervised	Object classifier
[Bronstein et al. 2011]	Mesh	No	Thousands	Local	No	Similarity Sensitive Hashing	Supervised	Distance metric
[Huang et al. 2013a]	Mesh	Pre-align.	Thousands	Local	No	Max-marginal distance learning	Semi-supervised	Distance metric
[Kalogerakis et al. 2010]	Mesh	No	Tens	Local	Yes	Jointboost classifier	Supervised	Face classifier
[van Kaick et al. 2011a]	Mesh	Yes	Tens	Local	Yes	Gentleboost classifier	Supervised	Face classifier
[Benhabiles et al. 2011]	Mesh	No	Tens	L.&G.	Yes	Adaboost classifier	Supervised	Boundary classifier
[Xie et al. 2014]	Mesh	No	Hundreds	Local	Yes	Feedforward neural networks	Supervised	Face/patch classifier
[Xu et al. 2014b]	Mesh	Pre-seg.	Tens	Local	No	Sparse model selection	Supervised	Segment similarity
[Lv et al. 2012]	Mesh	No	Tens	Local	Yes	Entropy regularization	Semi-supervised	Face classifier
[Wang et al. 2012]	Mesh	Pre-seg.	Hundreds	Local	No	Active learning	Semi-supervised	Segment classifier
[Wang et al. 2013b]	Image	Labeled parts	Hundreds	Local	No	2D shape matching	Supervised	2D shape similarity
[Hu et al. 2012a]	Mesh	Over-seg.	Tens	Local	Yes	Subspace clustering	Unsupervised	Patch similarity
[Sidi et al. 2011]	Mesh	Pre-seg.	Tens	Local	No	Spectral clustering	Unsupervised	Seg. simi./classifier
[Xu et al. 2010]	Mesh	Part	Tens	Struct.	No	Spectral clustering	Unsupervised	Part proportion simi.
[van Kaick et al. 2013]	Mesh	Part	Tens	Struct.	No	Multi-instance clustering	Unsupervised	Seg. hier. simi.
[Golovinskiy and Funkhouser 2009b]	Mesh	No	Tens	Global	No	Global shape alignment	Unsupervised	Face similarity
[Huang et al. 2011]	Mesh	Pre-seg.	Tens	Local	No	Joint part matching	Unsupervised	Segment similarity
[Huang et al. 2014b]	Mesh	Init. corr.	Tens	Global	No	Consistent func. map networks	Unsupervised	Segment similarity
[Kim et al. 2013a]	Mesh	Template	Thousands	Local	No	Shape alignment	Semi-supervised	Templates
[Mattausch et al. 2014]	Mesh	Over-seg.	Hundreds	Local	No	Density-based clustering	Unsupervised	Patch similarity
[Nguyen et al. 2011]	Mesh	Init. corr.	Tens	L.&G.	No	Inconsistent map detection	Unsupervised	Point similarity
[Huang et al. 2012b]	Mesh	Init. corr.	Tens	L.&G.	No	MRF joint matching	Unsupervised	Point similarity
[Kim et al. 2012a]	Mesh	Pre-align.	Tens	Global	No	Spectral matrix recovery	Unsupervised	Point similarity
[Huang and Guibas 2013a]	Mesh	Init. corr.	Tens	Global	No	Low-rank matrix recovery	Unsupervised	Point similarity
[Ovsjanikov et al. 2011]	Mesh	Part	Hundreds	Global	No	Manifold learning	Unsupervised	Parametric model
[Rustamov et al. 2013]	Mesh	Map	Tens	None	N/A	Functional map analysis	Unsupervised	Difference operator
[Fish et al. 2014]	Mesh	Labeled parts	Hundreds	Struct.	No	Kernel Density Estimation	Supervised	Prob. distributions
[Averkiou et al. 2014]	Mesh	[Kim et al. 2013a]	Thousands	Struct.	No	Manifold learning	Unsupervised	Parametric models
[Huang et al. 2013b]	Mesh	No	Hundreds	Global	No	Quartet analysis and clustering	Unsupervised	Distance measure
[Blanz and Vetter 1999]	Mesh	Pre-align.	Hundreds	Local	No	Principal Component Analysis	Unsupervised	Parametric model
[Allen et al. 2003]	Point	Pre-align.	Hundreds	Local	No	Principal Component Analysis	Unsupervised	Parametric model
[Hasler et al. 2009b]	Point	Pre-align.	Hundreds	Local	No	PCA & linear regression	Unsupervised	Parametric model
[Pauly et al. 2005a]	Mesh	Pre-align.	Hundreds	Global	No	Global shape alignment	Unsupervised	Shape similarity
[Nan et al. 2012]	Point	Labeled parts	Hundreds	Struct.	No	Random Forest Classifier	Supervised	Object classifier
[Shen et al. 2012]	Mesh	Labeled parts	Tens	Global	No	Part matching	Unsupervised	Part detector
[Kim et al. 2012b]	Point	Labeled parts	Tens	Local	No	Joint part fitting and matching	Unsupervised	Object detector
[Salas-Moreno et al. 2013]	Mesh	No	Tens	L.&G.	No	Shape matching	Unsupervised	Object detector
[Xu et al. 2011]	Mesh	Labeled parts	Tens	Struct.	No	Structural shape matching	Unsupervised	Part detector
[Aubry et al. 2014]	Mesh	Projected	Thousands	Visual	No	Linear Discriminant Analysis	Supervised	Object detector
[Su et al. 2014]	Mesh	Projected	Tens	Visual	No	Shape matching	Unsupervised	2D-3D correlation
[Chaudhuri and Koltun 2010]	Mesh	No	Thousands	Global	No	Shape matching	Unsupervised	Part detector
[Chaudhuri et al. 2011a]	Mesh	[Kalogerakis et al. 2010]	Hundreds	Local	No	Bayesian Network	Unsupervised	Part reasoning model
[Xie et al. 2013]	Mesh	Labeled parts	Tens	Struct.	No	Contextual part matching	Unsupervised	Part detector
[Kalogerakis et al. 2012a]	Mesh	[Kalogerakis et al. 2010]	Hundreds	L.&G.	No	Bayesian Network	Unsupervised	Shape reasoning model
[Xu et al. 2012]	Mesh	Part	Tens	Struct.	No	Part matching	Unsupervised	Part similarity
[Talton et al. 2012]	Mesh	Labeled parts	Tens	Struct.	No	Structured concept learning	Unsupervised	Probabilistic grammar
[Yumer and Kara 2012a]	Mesh	No	Tens	Global	No	Shape matching	Unsupervised	Shape abs. similarity
[Yumer and Kara 2014]	Mesh	Pre-seg.	Tens	Local	No	Segment matching	Unsupervised	Segment abs. simi.
[Chaudhuri et al. 2013]	Mesh	[Kalogerakis et al. 2010]	Hundreds	L.&G.	No	SVM ranking	Supervised	Ranking metric
[Fisher et al. 2011]	Scene	Labeled obj.	Tens	Struct.	No	Relevance feedback	Supervised	Contextual obj. simi.
[Fisher et al. 2012]	Scene	Labeled obj.	Hundreds	Struct.	No	Bayesian Network	Supervised	Mixture models
[Xu et al. 2013a]	Scene	Labeled obj.	Hundreds	Struct.	No	Frequent subgraph mining	Unsupervised	Frequent obj. groups
[Xu et al. 2014a]	Scene	Labeled obj.	Hundreds	Struct.	No	Weighted subgraph mining	Unsupervised	Distinct obj. groups
[Liu et al. 2014]	Scene	Labeled hier.	Tens	Struct.	No	Probabilistic learning	Supervised	Probabilistic grammar

Table 5: Comparison of various works on data-driven shape analysis and processing. For each work, we summarize over the criterion set defined for data-driven methods: training data (encompassing data representation, preprocessing and scale), feature (including feature type and whether feature selection is involved), learning model or approach, learning type (supervised, semi-supervised, and unsupervised), learning outcome (e.g., a classifier or a distance metric), as well as its typical application scenario. See the text for detailed explanation of the criteria. Some works employ another work as a pre-processing stage (e.g., [Chaudhuri et al. 2013] requires the labeled segmentation produced by [Kalogerakis et al. 2010]). There are four types of features including local geometric features (Local), global shape descriptors (Global), both local and global shape features (L.&G.), structural features (Struct.) as well as 2D visual features (Visual).

ALON, N., MOSHKOVITZ, D., AND SAFRA, S. 2006. Algorithmic construction of sets for k-restrictions. *ACM Trans. Algorithms* 2, 153–177.

AMAZON, 2009. Mechanical Turk. <http://www.mturk.com>.

AMENTA, N., BERN, M., AND KAMVYSELIS, M. 1998. A new voronoi-based surface reconstruction algorithm. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, ACM, SIGGRAPH ’98, 415–421.

AMIT, Y., FINK, M., SREBRO, N., AND ULLMAN, S. 2007. Uncovering shared structures in multiclass classification. 17–24.

ANGUELOV, D., SRINIVASAN, P., PANG, H.-C., KOLLER, D., THRUN, S., AND DAVIS, J. 2004. The correlated correspondence algorithm for unsupervised registration of nonrigid surfaces. *NIPS 17*, 33–40.

ANGUELOV, D., SRINIVASAN, P., KOLLER, D., THRUN, S., RODGERS, J., AND DAVIS, J. 2005. Scape: shape completion

- and animation of people. In *Proc. of SIGGRAPH*, 408–416.
- ANGUELOV, D., SRINIVASAN, P., KOLLER, D., THRUN, S., RODGERS, J., AND DAVIS, J. 2005. Scape: shape completion and animation of people. In *ACM SIGGRAPH 2005 Papers*, SIGGRAPH '05, 408–416.
- ANGUELOV, D., TASKAR, B., CHATALBASHEV, V., KOLLER, D., GUPTA, D., HEITZ, G., AND NG, A. 2005. Discriminative learning of markov random fields for segmentation of 3D scan data. In *Proc. CVPR*.
- ANKERST, M., KASTENMÜLLER, G., KRIESEL, H.-P., AND SEIDL, T. 1999. 3D shape histograms for similarity search and classification in spatial databases. In *SSD'99*, Springer, 207–226.
- ATTENE, M., KATZ, S., MORTARA, M., PATANE, G., SPAGNUOLO, M., AND TAL, A. 2006. Mesh segmentation - a comparative study. In *Proceedings of the IEEE International Conference on Shape Modeling and Applications 2006*, IEEE Computer Society, Washington, DC, USA, 7–.
- ATTENE, M., FALCIDENO, B., AND SPAGNUOLO, M. 2006. Hierarchical mesh segmentation based on fitting primitives. *The Visual Computer* 22, 3, 181–193.
- AUBRY, M., MATURANA, D., EFROS, A. A., RUSSELL, B. C., AND SIVIC, J. 2014. Seeing 3d chairs: Exemplar part-based 2d-3d alignment using a large dataset of cad models. In *Proc. CVPR*.
- AVERKIOU, M., KIM, V. G., ZHENG, Y., AND MITRA, N. J. 2014. ShapeSynth: Parameterizing Model Collections for Coupled Shape Exploration and Synthesis. *Computer Graphics Forum* 33, 2.
- AVERKIOU, M., KIM, V. G., AND MITRA, N. J. 2015. Autocorrelation descriptor for efficient co-alignment of 3d shape collections. *Computer Graphics Forum*.
- BAAK, A., MÜLLER, M., BHARAJ, G., SEIDEL, H.-P., AND THEOBALT, C. 2011. A data-driven approach for real-time full body pose reconstruction from a depth camera. In *Proc. ICCV*, 1092–1099.
- BAGHSNAH, M. S., AND SHOURAKI, S. B. 2009. Semi-supervised metric learning using pairwise constraints. IJCAI'09, 1217–1222.
- BANKO, M., AND BRILL, E. 2001. Mitigating the paucity-of-data problem: exploring the effect of training corpus size on classifier performance for natural language processing. In *Proc. Int. Conf. on Human Lang. Tech. Research*, 1–5.
- BARRETT, W. W., JOHNSON, C. R., AND LOEWY, R. 1996. *The Real Positive Definite Completion Problem: Cycle Completeness*. Amer Mathematical Society.
- BARUTCUOGLU, Z., AND DECORO, C. 2006. Hierarchical shape classification using bayesian aggregation. *Shape Modeling International*.
- BELONGIE, S., MALIK, J., AND PUZICHA, J. 2002. Shape Matching and Object Recognition Using Shape Contexts. *IEEE Trans. Pattern Anal. Mach. Intell.* 24, 4.
- BEN-CHEN, M., BUTSCHER, A., SOLOMON, J., AND GUIBAS, L. J. 2010. On discrete killing vector fields and patterns on surfaces. *Comput. Graph. Forum* 29, 5, 1701–1711.
- BENGIO, Y. 2009. Learning deep architectures for ai. *Foundations and Trends in Machine Learning* 2, 1, 1–127.
- BENHABILES, H., LAVOUÉ, G., VANDEBORRE, J.-P., AND DAOUDI, M. 2011. Learning boundary edges for 3d-mesh segmentation. *Computer Graphics Forum* 30, 8.
- BERNER, A., BOKELOH, M., WAND, M., SCHILLING, A., AND SEIDEL, H.-P. 2008. A graph-based approach to symmetry detection. In *Proceedings of the Eurographics / IEEE VGTC Workshop on Volume Graphics 2008, Los Angeles, California, USA, 2008*, 1–8.
- BESL, P. J., AND MCKAY, N. D. 1992. A method for registration of 3-d shapes. *IEEE Trans. Pattern Anal. Mach. Intell.* 14, 2.
- BESL, P. J., AND MCKAY, N. D. 1992. A method for registration of 3-d shapes. *IEEE Trans. Pat. Ana. & Mach. Int.* 14, 2, 239–256.
- BEZDEK, J. C., AND HATHAWAY, R. J. 2003. Convergence of alternating optimization. *Neural, Parallel Sci. Comput.* 11, 351–368.
- BIEDERMAN, I. 1987. Recognition-by-components: A theory of human image understanding. *Psychological Review* 94, 115–147.
- BISHOP, C. M. 2006. *Pattern Recognition and Machine Learning*. Springer.
- BLANZ, V., AND VETTER, T. 1999. A morphable model for the synthesis of 3D faces. In *Proc. of SIGGRAPH*, 187–194.
- BLUM, M., SPRINGENBERG, J. T., WULFING, J., AND RIEDMILLER, M. 2012. A learned feature descriptor for object recognition in RGB-D data. In *Proc. IEEE Int. Conf. on Rob. and Auto.*, 1298–1303.
- BO, L., REN, X., AND FOX, D. 2014. Learning hierarchical sparse features for RGB-(D) object recognition. *International Journal of Robotics Research*, to appear.
- BOGO, F., ROMERO, J., LOPER, M., AND BLACK, M. J. 2014. FAUST: Dataset and evaluation for 3D mesh registration. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 3794–3801.
- BOKELOH, M., BERNER, A., WAND, M., SEIDEL, H.-P., AND SCHILLING, A. 2009. Symmetry detection using line features. *Computer Graphics Forum* 28, 2, 697–706.
- BOKELOH, M., WAND, M., AND SEIDEL, H.-P. 2010. A connection between partial symmetry and inverse procedural modeling. In *ACM SIGGRAPH 2010 papers*, ACM, SIGGRAPH '10, 104:1–104:10.
- BOMMES, D., ZIMMER, H., AND KOBBELT, L. 2009. Mixed-integer quadrangulation. In *ACM SIGGRAPH 2009 papers*, ACM, SIGGRAPH '09, 77:1–77:10.
- BOSCAINI, D., MASCI, J., MELZI, S., BRONSTEIN, M. M., CASTELLANI, U., AND VANDERGHEYNST, P. 2015. Learning class-specific descriptors for deformable shapes using localized spectral convolutional networks. *Computer Graphics Forum* 34, 5.
- BOYD, S., AND VANDENBERGHE, L. 2004. *Convex Optimization*. Cambridge University Press.
- BOYD, S., PARikh, N., CHU, E., PELEATO, B., AND ECKSTEIN, J. 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.* 3, 1, 1–122.

- BOYD, S., PARIKH, N., CHU, E., PELEATO, B., AND ECKSTEIN, J. 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.* 3, 1, 1–122.
- BRONSTEIN, M. M., AND BRONSTEIN, A. M. 2011. Shape recognition with spectral distances. *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)* 33, 5, 1065–1071.
- BRONSTEIN, M. M., BRONSTEIN, A. M., AND KIMMEL, R. 2006. Generalized multidimensional scaling: a framework for isometry-invariant partial surface matching. 1168–1172.
- BRONSTEIN, A., BRONSTEIN, M., AND KIMMEL, R. 2008. *Numerical Geometry of Non-Rigid Shapes*, 1 ed. Springer Publishing Company, Incorporated.
- BRONSTEIN, A. M., BRONSTEIN, M. M., OVSJANIKOV, M., AND GUIBAS, L. J. 2011. Shape google: geometric words and expressions for invariant shape retrieval. *ACM Trans. Graphics* 30, 1, 1–20.
- BRONSTEIN, M. M., GLASHOFF, K., AND LORING, T. A. 2014. Making Laplacians commute: multimodal spectral geometry using closest commuting operators. SIAM Journal on Imaging Sciences (SIIS), submitted.
- BROWN, B., AND RUSINKIEWICZ, S. 2007. Global non-rigid alignment of 3-D scans. *ACM Trans. Graph.* 26, 3.
- CAMPEN, M., ATTENE, M., AND KOBBELT, L. 2012. A Practical Guide to Polygon Mesh Repairing. In *Eurographics tutorials*.
- CANDÈS, E. J., AND PLAN, Y. 2010. Matrix completion with noise. *Proceedings of the IEEE* 98, 6, 925–936.
- CANDÈS, E. J., AND PLAN, Y. 2011. Tight oracle inequalities for low-rank matrix recovery from a minimal number of noisy random measurements. *IEEE Trans. Inf. Theor.* 57, 4, 2342–2359.
- CANDÈS, E. J., AND RECHT, B. 2009. Exact matrix completion via convex optimization. *Found. Comput. Math.* 9, 6, 717–772.
- CANDÈS, E. J., AND TAO, T. 2005. Decoding by linear programming. *Trans. Inf. Theor.* 51, 12, 4203–4215.
- CANDÈS, E. J., ROMBERG, J., AND TAO, T. 2006. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory* 52, 2, 489–509.
- CANDÈS, E. J., LI, X., MA, Y., AND WRIGHT, J. 2011. Robust principal component analysis? *J. ACM* 58, 3, 11:1–11:37.
- CAO, C., WENG, Y., LIN, S., AND ZHOU, K. 2013. 3d shape regression for real-time facial animation. *ACM Trans. Graph.* 32, 4, 41:1–41:10.
- CAO, C., HOU, Q., AND ZHOU, K. 2014. Displaced dynamic expression regression for real-time facial tracking and animation. *ACM Trans. Graph.* 33, 4, 43:1–43:10.
- CAZALS, F., AND POUGET, M. 2003. Estimating differential quantities using polynomial fitting of osculating jets. In *Proc. Symp. on Geom. Proc.*, 177–187.
- CEYLAN, D., MITRA, N. J., ZHENG, Y., AND PAULY, M. 2014. Coupled structure-from-motion and 3d symmetry detection for urban facades. *ACM Trans. Graph.* 33, 1, 2:1–2:15.
- CHANG, W., AND ZWICKER, M. 2008. Automatic registration for articulated shapes. *Comput. Graph. Forum* 27, 5, 1459–1468.
- CHANG, J. Y., RASKAR, R., AND AGRAWAL, A. K. 2009. In *Proc. CVPR*, 1706–1713.
- CHANG, W., LI, H., MITRA, N., PAULY, M., AND WAND, M. 2012. Dynamic Geometry Processing. In *Eurographics tutorials*.
- CHANG, W., LI, H., MITRA, N. J., PAULY, M., AND WAND, M. 2012. Dynamic geometry processing. In *Eurographics 2012: Tutorial Notes*.
- CHARPIAT, G. 2009. Learning shape metrics based on deformations and transport. In *NORDIA 2009*.
- CHAUDHURI, S., AND KOLTUN, V. 2010. Data-driven suggestions for creativity support in 3d modeling. *ACM Trans. Graph.* 29, 6, 183:1–183:10.
- CHAUDHURI, S., KALOGERAKIS, E., GUIBAS, L., AND KOLTUN, V. 2011. Probabilistic reasoning for assembly-based 3d modeling. *ACM Trans. Graph.* 30, 4, 35:1–35:10.
- CHAUDHURI, S., KALOGERAKIS, E., GUIBAS, L., AND KOLTUN, V. 2011. Probabilistic reasoning for assembly-based 3d modeling. *ACM Trans. Graph.* 30, 4, 35:1–35:10.
- CHAUDHURI, S., KALOGERAKIS, E., GIGUERE, S., , AND FUNKHOUSER, T. 2013. AttribIt: Content creation with semantic attributes. *UIST*.
- CHEKURI, C., KHANNA, S., NAOR, J. S., AND ZOSIN, L. 2001. Approximation algorithms for the metric labeling problem via a new linear programming formulation. In *Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, SODA ’01, 109–118.
- CHEN, Y., AND MEDIONI, G. 1992. Object modelling by registration of multiple range images. *Image Vision Comput.* 10, 145–155.
- CHEN, D.-Y., TIAN, X.-P., SHEN, Y.-T., AND OUHYOUNG, M. 2003. On visual similarity based 3D model retrieval. *Computer Graphics Forum* 22, 3, 223–232.
- CHEN, D.-Y., TIAN, X.-P., SHEN, Y.-T., AND OUHYOUNG, M. 2003. On visual similarity based 3D model retrieval. *Computer Graphics Forum* 22, 3, 223–232.
- CHEN, G., SONG, Y., WANG, F., AND ZHANG, C. 2008. Semi-supervised multi-label learning by solving a sylvester equation. In *SDM [DBL 2008]*, 410–419.
- CHEN, X., KANG, S. B., XU, Y.-Q., DORSEY, J., AND SHUM, H.-Y. 2008. Sketching reality: Realistic interpretation of architectural. *ACM Trans. Graph.* 27, 2, 11:1–11:21.
- CHEN, X., GOLOVINSKIY, A., AND FUNKHOUSER, T. 2009. A benchmark for 3D mesh segmentation. *ACM Trans. Graph.* 28, 3, 73:1–73:12.
- CHEN, X., SAPAROV, A., PANG, B., AND FUNKHOUSER, T. 2012. Schelling points on 3D surface meshes. *ACM Trans. Graph.*
- CHEN, C., WENG, Y., ZHOU, S., TONG, Y., AND ZHOU, K. 2014. Facewarehouse: a 3d facial expression database for visual computing. *IEEE Trans. Vis. & Comp. Graphics* 20, 3.
- CHEN, K., LAI, Y.-K., WU, Y.-X., MARTIN, R., AND HU, S.-M. 2014. Automatic semantic modeling of indoor scenes from low-quality rgbd data using contextual information. *ACM Trans. Graph.* 33, 6.

- CHEN, Y., GUIBAS, L. J., AND HUANG, Q.-X. 2014. Near-optimal joint object matching via convex relaxation. *CoRR abs/1402.1473*.
- CHO, T. S., AVIDAN, S., AND FREEMAN, W. T. 2010. The patch transform. *IEEE Trans. Pattern Anal. Mach. Intell.* 32, 8, 1489–1501.
- CHO, T. S., AVIDAN, S., AND FREEMAN, W. T. 2010. A probabilistic image jigsaw puzzle solver. In *Proc. CVPR*, 183–190.
- CHOI, W., CHAO, Y.-W., PANTOFARU, C., AND SAVARESE, S. 2013. Understanding indoor scenes using 3d geometric phrases. In *Proc. CVPR*.
- CHOURDHALY, S., TREVOR, A., CHRISTENSEN, H., AND DELLAERT, F. 2014. Slam with object discovery; modeling and mapping.
- CHUI, H., AND RANGARAJAN, A. 2003. A new point matching algorithm for non-rigid registration. *Comput. Vis. Image Underst.* 89, 2-3, 114–141.
- CHUNG, F. R. K., LU, L., AND VU, V. H. 2003. The spectra of random graphs with given expected degrees. *Internet Mathematics* 1, 3, 257–275.
- CLARENZ, U., RUMPF, M., AND TELEA, A. 2004. Robust feature detection and local classification for surfaces based on moment analysis. *TVCG* 10, 5, 516–524.
- COHEN-STEINER, D., ALLIEZ, P., AND DESBRUN, M. 2004. Variational shape approximation. In *ACM SIGGRAPH 2004 Papers*, ACM, SIGGRAPH ’04, 905–914.
- COIFMAN, R. R., LAFON, S., LEE, A. B., MAGGIONI, M., WARNER, F., AND ZUCKER, S. 2005. Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. In *PNAS*, 7426–7431.
- COLE, F., GOLOVINSKIY, A., LIMPAECHER, A., BARROS, H. S., FINKELSTEIN, A., FUNKHOUSER, T., AND RUSINKIEWICZ, S. 2008. Where do people draw lines? *ACM Trans. Graph.*
- CONSTALES, D. 1998. A closed formula for the moore-penrose generalized inverse of a complex matrix of given rank. *Acta Math. Hungar.* 80, 83–88.
- COOPER, M., FOOTE, J., GIRGENSOHN, A., AND WILCOX, L. 2005. Temporal event clustering for digital photo collections. *ACM Trans. on Multi. Comp., Comm. and App.* 1, 3, 269–288.
- COSTA, L. D. F. D., AND CESAR, JR., R. M. 2000. *Shape Analysis and Classification: Theory and Practice*, 1st ed. CRC Press, Inc., Boca Raton, FL, USA.
- COYNE, B., AND SPROAT, R. 2001. Wordseye: An automatic text-to-scene conversion system. In *Proc. of SIGGRAPH*.
- CRANDALL, D., OWENS, A., SNAVELY, N., AND HUTTENLOCHER, D. 2011. Discrete-continuous optimization for large-scale structure from motion. *CVPR ’11*, 3001–3008.
- CYR, C. M., AND KIMIA, B. B. 2004. A similarity-based aspect-graph approach to 3d object recognition. *Int. J. Comput. Vision* 57, 1, 5–22.
- DA FONTOURA COSTA, L., AND CESAR JR., R. M. 2009. *Shape Classification and Analysis: Theory and Practice*, 2nd ed. CRC Press, Inc., Boca Raton, FL, USA.
- DANTZIG, G. B. 1951. Maximization of a linear function of variables subject to linear inequalities. In *Activity Analysis of Production and Allocation*, Cowles Commission Monograph No. 13. John Wiley & Sons Inc., 339–347.
- DATTORRO, J. 2011. *Convex Optimization & Euclidean Distance Geometry*. Meboo Publishing USA.
2008. Proceedings of the SIAM International Conference on Data Mining, SDM 2008, April 24-26, 2008, Atlanta, Georgia, USA, SIAM.
2009. 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA, IEEE.
- DE GOES, F., GOLDENSTEIN, S., AND VELHO, L. 2008. A hierarchical segmentation of articulated bodies. *Comput. Graph. Forum* 27, 5, 1349–1356.
- DE OLIVEIRA, M. C. F., AND LEVKOWITZ, H. 2003. From visual data exploration to visual data mining: a survey. *IEEE Trans. Vis. & Comp. Graphics* 9, 3, 378–394.
- DECARLO, D., FINKELSTEIN, A., RUSINKIEWICZ, S., AND SANTELLA, A. 2003. Suggestive contours for conveying shape. In *ACM SIGGRAPH 2003 Papers*, ACM, New York, NY, USA, SIGGRAPH ’03, 848–855.
- DENG, J., DONG, W., SOCHER, R., LI, L.-J., LI, K., AND LI, F.-F. 2009. In *Proc. CVPR*, 248–255.
- DENG, J., KRAUSE, J., AND FEI-FEI, L. 2013. Fine-grained crowdsourcing for fine-grained recognition. In *CVPR’13*.
- DEY, T. K. 2007. *Curve and surface reconstruction: algorithms with mathematical analysis*, vol. 23. Cambridge University Press.
- DIEBEL, J., THRUN, S., AND BRUENING, M. 2006. A bayesian method for probable surface reconstruction and decimation. *ACM Trans. Graph.* 25, 1.
- DING, X., AND JIANG, T. 2010. Spectral distributions of adjacency and laplacian matrices of random graphs. *Annals of Applied Probability* 20(6), 2086–2117.
- DO CARMO, M. 1976. *Differential Geometry of Curves and Surfaces*. Prentice Hall.
- DOMINGOS, P. 2012. A few useful things to know about machine learning. *Communications of the ACM* 55, 10, 78–87.
- DONG, S., KIRCHER, S., AND GARLAND, M. 2005. Harmonic functions for quadrilateral remeshing of arbitrary manifolds. *Comput. Aided Geom. Des.* 22, 5, 392–423.
- DONG, S., BREMER, P.-T., GARLAND, M., PASCUCCI, V., AND HART, J. C. 2006. Spectral surface quadrangulation. In *ACM SIGGRAPH 2006 Papers*, ACM, New York, NY, USA, SIGGRAPH ’06, 1057–1066.
- DUDA, R. O., HART, P. E., AND STORK, D. G. 2000. *Pattern Classification (2Nd Edition)*. Wiley-Interscience.
- EITZ, M., RICHTER, R., BOUBEKEUR, T., HILDEBRAND, K., AND ALEXA, M. 2012. Sketch-based shape retrieval. *ACM Trans. Graph.* 31, 4, 5:1–5:10.
- EITZ, M., RICHTER, R., BOUBEKEUR, T., HILDEBRAND, K., AND ALEXA, M. 2012. Sketch-based shape retrieval. *ACM Trans. Graph.* 31, 4, 31:1–31:10.

- EYNARD, D., KOVNATSKY, A., BRONSTEIN, M. M., GLASHOFF, K., AND BRONSTEIN, A. M. Multimodal manifold analysis using simultaneous diagonalization of laplacians. submitted to SIAM Journal on Imaging Sciences.
- FAN, L., WANG, R., XU, L., DENG, J., AND LIU, L. 2013. Modeling by drawing with shadow guidance. *Computer Graphics Forum* 32, 7, 157–166.
- FEI-FEI, L., FERGUS, R., AND PERONA, P. 2006. One-shot learning of object categories. *IEEE Trans. Pat. Ana. & Mach. Int.* 28, 4.
- FERGUS, R., WEISS, Y., AND TORRALBA, A. 2009. Semi-supervised learning in gigantic image collections. In *NIPS*, 522–530.
- FIEDLER, M. 1973. Algebraic Connectivity of Graphs. *Czechoslovak Mathematical Journal* 23, 298–305.
- FISCHLER, M. A., AND BOLLES, R. C. 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* 24, 6, 381–395.
- FISH, N., AVERKIOU, M., VAN KAICK, O., SORKINE-HORNUNG, O., COHEN-OR, D., AND MITRA, N. J. 2014. Meta-representation of shape families. *ACM Trans. Graph.* 33, 4, 34:1–34:11.
- FISHER, M., AND HANRAHAN, P. 2010. Context-based search for 3d models. *ACM Trans. Graph.* 29, 6, 182:1–182:10.
- FISHER, M., SAVVA, M., AND HANRAHAN, P. 2011. Characterizing structural relationships in scenes using graph kernels. *ACM Trans. Graph.* 30, 4, 34:1–34:12.
- FISHER, M., RITCHIE, D., SAVVA, M., FUNKHOUSER, T., AND HANRAHAN, P. 2012. Example-based synthesis of 3d object arrangements. *ACM Trans. Graph.* 31, 6, 135:1–135:12.
- FLORIANI, LEILA DE; SPAGNUOLO, M. E. 2007. *Shape Analysis and Structuring*. Springer. <http://www.springer.com/west/home/math/analysis?SGWID=4-10044-22-173665611-detailsPage=ppmmedia—toc>.
- FOSSATI, A., GALL, J., GRABNER, H., REN, X., AND KONOLIGE, K. 2013. *Consumer Depth Cameras for Computer Vision, Chapter 12*. Springer.
- FOUHEY, D. F., GUPTA, A., AND HEBERT, M. 2013. Data-driven 3d primitives for single image understanding. In *Proc. ICCV*.
- FREIFELD, O., AND BLACK, M. J. 2012. Lie bodies: A manifold representation of 3d human shape. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part I, ECCV'12*, 1–14.
- FREUND, Y., AND SCHAPIRA, R. E. 1995. A decision-theoretic generalization of on-line learning and an application to boosting. Springer-Verlag, London, UK, UK, EuroCOLT '95, 23–37.
- FROME, A., HUBER, D., KOLLURI, R., BULOW, T., AND MALIK, J. 2004. Recognizing objects in range data using regional point descriptors. In *Proc. ECCV*. 224–237.
- FU, H., COHEN-OR, D., DROR, G., AND SHEFFER, A. Upright orientation of man-made objects. *SIGGRAPH '08*, 42:1–42:7.
- FUKUNAGA, K. 1990. *Introduction to Statistical Pattern Recognition*. Academic Press Professional, Inc.
- FUNKHOUSER, T., AND SHILANE, P. 2006. Partial matching of 3d shapes with priority-driven search. In *Proc. Symp. on Geom. Proc.*
- FUNKHOUSER, T., MIN, P., KAZHDAN, M., CHEN, J., HALDERMAN, A., DOBKIN, D., AND JACOBS, D. 2003. A search engine for 3D models. *ACM Trans. Graph.* 22, 1, 83–105.
- FUNKHOUSER, T., KAZHDAN, M., SHILANE, P., MIN, P., KIEFER, W., TAL, A., RUSINKIEWICZ, S., AND DOBKIN, D. 2004. Modeling by example. *ACM Trans. Graph.* 23, 3, 652–663.
- FUNKHOUSER, T. A., KAZHDAN, M. M., MIN, P., AND SHILANE, P. 2005. Shape-based retrieval and analysis of 3d models. *Commun. ACM* 48, 6, 58–64.
- GAL, R., SHAMIR, A., HASSNER, T., PAULY, M., AND COHEN-OR, D. 2007. Surface reconstruction using local shape priors. In *Proc. Symp. on Geom. Proc.*, SGP '07, 253–262.
- GAL, R., SHAMIR, A., HASSNER, T., PAULY, M., AND COHEN-OR, D. 2007. Surface reconstruction using local shape priors. In *Proc. Symp. on Geom. Proc.*, 253–262.
- GAL, R., SORKINE, O., MITRA, N. J., AND COHEN-OR, D. 2009. iwiresh: an analyze-and-edit approach to shape manipulation. *ACM Trans. Graph.*, 33:1–33:10.
- GAO, L., LAI, Y.-K., HUANG, Q., AND HU, S.-M. 2013. A data-driven approach to realistic shape morphing. *Computer Graphics Forum* 32, 2, 449–457.
- GELFAND, N., AND GUIBAS, L. J. 2004. Shape segmentation using local slippage analysis. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, ACM, SGP '04, 214–223.
- GELFAND, N., MITRA, N. J., GUIBAS, L. J., AND POTTMANN, H. 2005. Robust global registration. In *Proc. Symp. on Geom. Proc.*
- GIORGIO, D., BIASOTTI, S., AND PARABOSCHI, L. 2007. Shrec: shape retrieval contest: Watertight models track.
- GIORGIO, D., BIASOTTI, S., AND PARABOSCHI, L. 2007. Shape retrieval contest 2007: Watertight models track.
- GOLDBERG, D., MALON, C., AND BERN, M. 2004. A global approach to automatic solution of jigsaw puzzles. *Comput. Geom. Theory Appl.* 28, 165–174.
- GOLDMAN, D. B., CURLESS, B., HERTZMANN, A., AND SEITZ, S. M. 2005. Shape and spatially-varying brdfs from photometric stereo. In *Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1 - Volume 01*, ICCV '05, 341–348.
- GOLOVINSKIY, A., AND FUNKHOUSER, T. 2008. Randomized cuts for 3d mesh analysis. In *ACM SIGGRAPH Asia 2008 papers*, ACM, SIGGRAPH Asia '08, 145:1–145:12.
- GOLOVINSKIY, A., AND FUNKHOUSER, T. 2009. Consistent Segmentation of 3D Models. *Proc. SMI* 33, 3.
- GOLOVINSKIY, A., AND FUNKHOUSER, T. A. 2009. Consistent segmentation of 3d models. *Computers & Graphics* 33, 3, 262–269.
- GOLOVINSKIY, A., KIM, V. G., AND FUNKHOUSER, T. 2009. Shape-based Recognition of 3D Point Clouds in Urban Environments. In *Proc. ICCV*.

- GÖNEN, M., AND ALPAYDIN, E. 2011. Multiple kernel learning algorithms. *J. of Mac. Learn. Res.* 12, 2211–2268.
- GRANT, M., AND BOYD, S., 2011. CVX: Matlab software for disciplined convex programming. <http://www.stanford.edu/~boyd/cvx/>.
- GUO, X., LIN, J., XU, K., AND JIN, X. 2014. Creature grammar for creative modeling of 3d monsters. *Graphical Models (Special Issue of GMP)* 76, 5, 376–389.
- HALLDÓRSSON, M., AND RADHAKRISHNAN, J. 1994. Greed is good: approximating independent sets in sparse and bounded-degree graphs. STOC '94, 439–448.
- HASLER, N., STOLL, C., ROSENHAHN, B., THORMÄHLEN, T., AND SEIDEL, H.-P. 2009. Technical section: Estimating body shape of dressed humans. *Comput. Graph.* 33, 3, 211–216.
- HASLER, N., STOLL, C., SUNKEL, M., ROSENHAHN, B., AND SEIDEL, H.-P. 2009. A statistical model of human pose and body shape. *Comput. Graph. Forum* 28, 2, 337–346.
- HASLER, N., THORMÄHLEN, T., ROSENHAHN, B., AND SEIDEL, H.-P. 2010. Learning skeletons for shape and pose. In *Proceedings of the 2010 ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, I3D '10, 23–30.
- HAYS, J., LEORDEANU, M., EFROS, A. A., AND LIU, Y. 2006. Discovering texture regularity as a higher-order correspondence problem. In *Computer Vision - ECCV 2006, 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006, Proceedings, Part II*, 522–535.
- HELD, D., LEVINSON, J., THRUN, S., AND SAVARESE, S. 2014. Combining 3d shape, color, and motion for robust anytime tracking. In *RSS*.
- HILAGA, M., SHINAGAWA, Y., KOHMURA, T., AND KUNII, T. L. 2001. Topology matching for fully automatic similarity estimation of 3d shapes. In *Proc. of SIGGRAPH*, 203–212.
- HINTON, G. E., OSINDERO, S., AND TEH, Y.-W. 2006. A fast learning algorithm for deep belief nets. *Neural Computation* 18, 7, 1527–1554.
- HOCHBAUM, D. S., AND SINGH, V. 2009. An efficient algorithm for co-segmentation. In *IEEE 12th International Conference on Computer Vision, ICCV 2009, Kyoto, Japan, September 27 - October 4, 2009*, 269–276.
- HOFER, M., ODEHNAL, B., POTTMANN, H., STEINER, T., AND WALLNER, J. 2005. 3d shape recognition and reconstruction based on line element geometry. ICCV '05, 1532–1538.
- HOFFMAN, D. D., AND SINGH, M. 1997. Salience of visual parts. *Cognition* 63, 1, 29–78.
- HOFFMAN, D., RICHARDS, W., PENTL, A., RUBIN, J., AND SCHEUHAMMER, J. 1983. Parts of recognition. *Cognition* 18, 65–96.
- HOI, S. C., LIU, W., AND CHANG, S.-F. 2010. Semi-supervised distance metric learning for collaborative image retrieval and clustering. *ACM Trans. Multimedia Comput. Commun. Appl.* 6, 3, 18:1–18:26.
- HOIEM, D., EFROS, A. A., AND HEBERT, M. 2005. Automatic photo pop-up. *ACM Trans. Graph.* 24, 3, 577–584.
- HOLLAND, P. W., AND WELSCH, R. E. 1977. Robust regression using iteratively reweighted least-squares. *Communications in Statistics: Theory and Methods* A6, 813–827.
- HORN, B. K. P., HILDEN, H., AND NEGAHDARIPOUR, S. 1988. Closed-form solutions of absolute orientation using orthonormal matrices. *Journal of the Optical Society* 5, 1127–1135.
- HORN, B. K. P. 1984. Extended Gaussian Images. *Proceedings of the IEEE* 72, 12, 1671–1686.
- HORN, B. K. P. 1987. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A* 4, 4, 629–642.
- HU, R., FAN, L., AND LIU, L. 2012. Co-segmentation of 3d shapes via subspace clustering. *Computer Graphics Forum* 31, 5.
- HU, R., FAN, L., AND LIU, L. 2012. Co-Segmentation of 3D shapes via subspace clustering. *Computer Graphics Forum* 31, 5, 1703–1713.
- HU, R., FAN, L., AND LIU, L. 2012. Co-Segmentation of 3D shapes via subspace clustering. *Comput. Graph. Forum* 31, 5, 1703–1713.
- HU, R., ZHU, C., VAN KAICK, O., LIU, L., SHAMIR, A., AND ZHANG, H. 2015. Interaction context (icon): Towards a geometric functionality descriptor. *ACM Trans. Graph.* 34, 4, 83:1–83:12.
- HUANG, Q.-X., AND ANGUELOV, D. 2010. High quality pose estimation by aligning multiple scans to a latent map. In *IEEE International Conference on Robotics and Automation, ICRA 2010, Anchorage, Alaska, USA, 3-7 May 2010*, 1353–1360.
- HUANG, Q., AND GUIBAS, L. 2013. Consistent shape maps via semidefinite programming. *Computer Graphics Forum (SGP)* 32, 5, 177–186.
- HUANG, Q., AND GUIBAS, L. 2013. Consistent shape maps via semidefinite programming. *Computer Graphics Forum (SGP)* 32, 5, 177–186.
- HUANG, Q., FLÖRY, S., GELFAND, N., HOFER, M., AND POTTMANN, H. 2006. Reassembling fractured objects by geometric matching. *ACM Trans. Graph.* 25, 3, 569–578.
- HUANG, Q.-X., ADAMS, B., AND WAND, M. 2007. Bayesian surface reconstruction via iterative scan alignment to an optimized prototype. In *Proc. Symp. on Geom. Proc.*, SGP '07, 213–223.
- HUANG, J., ZHANG, M., MA, J., LIU, X., KOBELT, L., AND BAO, H. 2008. Spectral quadrangulation with orientation and alignment control. In *ACM SIGGRAPH Asia 2008 papers*, ACM, SIGGRAPH Asia '08, 147:1–147:9.
- HUANG, Q., ADAMS, B., WICKE, M., AND GUIBAS, L. J. 2008. Non-rigid registration under isometric deformations. In *Proc. Symp. on Geom. Proc.*, 1449–1457.
- HUANG, Q.-X., WICKE, M., ADAMS, B., AND GUIBAS, L. J. 2009. Shape decomposition using modal analysis. *Comput. Graph. Forum* 28, 2, 407–416.
- HUANG, Q., KOLTUN, V., AND GUIBAS, L. 2011. Joint shape segmentation using linear programming. *ACM Trans. Graph.* 30, 6, 125:1–125:12.
- HUANG, Q., GUIBAS, L., AND MITRA, N. 2012. Near-regular structure extraction using linear programming. *ACM Trans. Graph.*.
- HUANG, Q., ZHANG, G.-X., GAO, L., HU, S.-M., BUTSCHER, A., AND GUIBAS, L. 2012. An optimization approach for

- extracting and encoding consistent maps in a shape collection. *ACM Trans. Graph.* 31, 6, 167:1–167:11.
- HUANG, Q., SU, H., AND GUIBAS, L. 2013. Fine-grained semi-supervised labeling of large shape collections. *ACM Trans. Graph.* 32, 6, 190:1–190:10.
- HUANG, S.-S., SHAMIR, A., SHEN, C.-H., ZHANG, H., SHEFFER, A., HU, S.-M., AND COHEN-OR, D. 2013. Qualitative organization of collections of shapes via quartet analysis. *ACM Trans. Graph.* 32, 4, 71:1–71:10.
- HUANG, Q., CHEN, Y., AND GUIBAS, L. J., 2014. Scalable semidefinite relaxation for maximum a posterior estimation.
- HUANG, Q., WANG, F., AND GUIBAS, L. 2014. Functional map networks for analyzing and exploring large shape collections. *ACM Trans. Graph.* 33, 4.
- HUANG, H., KALOGERAKIS, E., AND MARLIN, B. 2015. Analysis and synthesis of 3d shape families via deep-learned generative models of surfaces. *Computer Graphics Forum* 34, 5.
- HUANG, Q., WANG, H., AND KOLTUN, V. 2015. Single-view reconstruction via joint analysis of image and shape collections. *ACM Trans. Graph.* 34, 87:1–87:10.
- HUBER, D. 2002. *Automatic Three-dimensional Modeling from Reality*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA.
- HUETING, M., OVSJANIKOV, M., AND MITRA, N. J. 2015. Crosslink: Joint understanding of image and 3d model collections through shape and camera pose variations. *ACM Trans. Graph.*
- IBM. 2011. Ibm ilog cplex optimization studio. <http://www-01.ibm.com/software/integration/optimization/cplex-optimization-studio/>.
- JAIN, A., THORMÄHLEN, T., RITSCHEL, T., AND SEIDEL, H.-P. 2012. Exploring shape variations by 3D-model decomposition and part-based recombination. *Computer Graphics Forum* 31, 2, 631–640.
- JAMES, D. L., AND TWIGG, C. D. 2005. Skinning mesh animations. *ACM Trans. Graph.* 24, 3, 399–407.
- JIAO, F., WANG, S., LEE, C.-H., GREINER, R., AND SCHUURMANS, D. 2006. Semi-supervised conditional random fields for improved sequence segmentation and labeling. In *21st International Conference on Computational Linguistics*.
- JOHNSON, A., AND HEBERT, M. 1999. Using Spin Images for Efficient Object Recognition in Cluttered 3D Scenes. *IEEE Trans. Pat. Ana. & Mach. Int.* 21, 5, 433–449.
- JOHNSON, A. E., AND HEBERT, M. 1999. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Trans. Pattern Anal. Mach. Intell.* 21, 5, 433–449.
- JOHNSON, A. 1997. *Spin-Images: A Representation for 3-D Surface Matching*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA.
- JOULIN, A., BACH, F. R., AND PONCE, J. 2010. Discriminative clustering for image co-segmentation. In *Proc. CVPR*, 1943–1950.
- KAHRAMAN, A., MORRIS, R. J., LASKOWSKI, R. A., AND THORNTON, J. M. 2007. Shape variation in protein binding pockets and their ligands. *Journal of Molecular Biology* 368, 1, 283–301.
- KALOGERAKIS, E., SIMARI, P., NOWROUZEZAHRAI, D., AND SINGH, K. 2007. Robust statistical estimation of curvature on discretized surfaces. In *Proc. Symp. on Geom. Proc.*
- KALOGERAKIS, E., NOWROUZEZAHRAI, D., SIMARI, P., MCCRAE, J., HERTZMANN, A., AND SINGH, K. 2009. Data-driven curvature for real-time line drawing of dynamic scenes. *ACM Trans. Graph.* 28, 1, 1–13.
- KALOGERAKIS, E., HERTZMANN, A., AND SINGH, K. 2010. Learning 3d mesh segmentation and labeling. *ACM Trans. Graph.* 29, 102:1–102:12.
- KALOGERAKIS, E., CHAUDHURI, S., KOLLER, D., AND KOLTUN, V. 2012. A probabilistic model for component-based shape synthesis. *ACM Trans. Graph.* 31, 4.
- KALOGERAKIS, E., NOWROUZEZAHRAI, D., BRESLAV, S., AND HERTZMANN, A. 2012. Learning Hatching for Pen-and-Ink Illustration of Surfaces. *ACM Trans. Graph.* 31, 1.
- KARP, R. 1972. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, R. Miller and J. Thatcher, Eds. Plenum Press, 85–103.
- KASS, M., WITKIN, A., AND TERZOPOULOS, D. 1988. Snakes: Active contour models. *INTERNATIONAL JOURNAL OF COMPUTER VISION* 1, 4, 321–331.
- KATZ, S., AND TAL, A. 2003. Hierarchical mesh decomposition using fuzzy clustering and cuts. *SIGGRAPH '03*, 954–961.
- KATZ, S., LEIFMAN, G., AND TAL, A. 2005. Mesh segmentation using feature point and core extraction. *The Visual Computer* 21, 8–10, 649–658.
- KAZHDAN, M., CHAZELLE, B., DOBKIN, D., FUNKHOUSER, T., AND RUSINKIEWICZ, S. 2003. A reflective symmetry descriptor for 3d models. *Algorithmica* 38, 1.
- KAZHDAN, M., FUNKHOUSER, T., AND RUSINKIEWICZ, S. 2003. Rotation invariant spherical harmonic representation of 3d shape descriptors. *Proc. Symp. on Geom. Proc.*
- KAZHDAN, M., FUNKHOUSER, T., AND RUSINKIEWICZ, S. 2003. Rotation invariant spherical harmonic representation of 3d shape descriptors. *SGP '03*, 156–164.
- KAZHDAN, M., FUNKHOUSER, T., AND RUSINKIEWICZ, S. 2004. Symmetry descriptors and 3d shape matching. *Proc. Symp. on Geom. Proc..*
- KAZHDAN, M., FUNKHOUSER, T., AND RUSINKIEWICZ, S. 2004. Shape matching and anisotropy. In *ACM SIGGRAPH 2004 Papers*, ACM, *SIGGRAPH '04*, 623–629.
- KAZHDAN, M., FUNKHOUSER, T., AND RUSINKIEWICZ, S. 2004. Symmetry descriptors and 3d shape matching. In *Proc. Symp. on Geom. Proc.*, ACM, *SGP '04*, 115–123.
- KAZHDAN, M., BOLITHO, M., AND HOPPE, H. 2006. Poisson surface reconstruction. In *Proc. Symp. on Geom. Proc.*, *SGP '06*, 61–70.
- KHOLGADE, N., SIMON, T., EFROS, A., AND SHEIKH, Y. 2014. 3d object manipulation in a single photograph using stock 3d models. *ACM Trans. Graph.* 33, 4, 127:1–127:12.
- KILIAN, M., MITRA, N. J., AND POTTMANN, H. 2007. Geometric modeling in shape space. In *ACM SIGGRAPH 2007 papers*, ACM, *SIGGRAPH '07*.

- KIM, V. G., LIPMAN, Y., CHEN, X., AND FUNKHOUSER, T. A. 2010. Möbius transformations for global intrinsic symmetry analysis. *Comput. Graph. Forum* 29, 5, 1689–1700.
- KIM, V. G., LIPMAN, Y., AND FUNKHOUSER, T. 2011. Blended intrinsic maps. *ACM Trans. Graph.* 30, 4, 79:1–79:12.
- KIM, V. G., LI, W., MITRA, N. J., DiVERDI, S., AND FUNKHOUSER, T. 2012. Exploring collections of 3D models using fuzzy correspondences. *ACM Trans. Graph.* 31, 4, 54:1–54:11.
- KIM, Y. M., MITRA, N. J., YAN, D.-M., AND GUIBAS, L. 2012. Acquiring 3d indoor environments with variability and repetition. *ACM Trans. Graph.* 31, 6, 138:1–138:11.
- KIM, V. G., LI, W., MITRA, N. J., CHAUDHURI, S., DiVERDI, S., AND FUNKHOUSER, T. 2013. Learning part-based templates from large collections of 3D shapes. *ACM Trans. Graph.* 32, 4, 70:1–70:12.
- KIM, V. G., LI, W., MITRA, N. J., CHAUDHURI, S., DiVERDI, S., AND FUNKHOUSER, T. 2013. Learning Part-based Templates from Large Collections of 3D Shapes. *Transactions on Graphics (Proc. of SIGGRAPH)* 32, 4.
- KIM, Y. M., MITRA, N. J., HUANG, Q.-X., AND GUIBAS, L. J. 2013. Guided real-time scanning of indoor objects. *Comput. Graph. Forum* 32, 7, 177–186.
- KIM, V. G., CHAUDHURI, S., GUIBAS, L., AND FUNKHOUSER, T. 2014. Shape2Pose: Human-Centric Shape Analysis. *ACM Trans. Graph.* 33, 4.
- KIN-CHUNG AU, O., TAI, C.-L., COHEN-OR, D., ZHENG, Y., AND FU, H. 2010. Electors voting for fast automatic shape correspondence. *Computer Graphics Forum* 29, 2, 645–654.
- KLEINBERG, J., AND TARDOS, E. 2002. Approximation algorithms for classification problems with pairwise relationships: metric labeling and markov random fields. *J. ACM* 49, 616–639.
- KOBBELT, L., CAMPAGNA, S., AND SEIDEL, H. P. 1998. A general framework for mesh decimation. Citeseer, 43–50.
- KOKKINOS, I., BRONSTEIN, M., LITMAN, R., AND BRONSTEIN, A. 2012. Intrinsic shape context descriptors for deformable shapes. In *Proc. CVPR*.
- KOLLER, D., AND FRIEDMAN, N. 2009. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press.
- KOLLER, D., AND FRIEDMAN, N. 2009. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press.
- KOLMOGOROV, V. 2006. Convergent tree-reweighted message passing for energy minimization. *IEEE Trans. Pattern Anal. Mach. Intell.* 28, 1568–1583.
- KOMODAKIS, N., PARAGIOS, N., AND TZIRITAS, G. 2007. Mrf optimization via dual decomposition: Message-passing revisited. *Computer Vision, IEEE International Conference on* 0, 1–8.
- KOVNATSKY, A., BRONSTEIN, M. M., BRONSTEIN, A. M., GLASHOFF, K., AND KIMMEL, R. 2013. Coupled quasi-harmonic bases. In *Eurographics'13*, 439–448.
- KRAEVOY, V., AND SHEFFER, A. 2004. Cross-parameterization and compatible remeshing of 3d models. In *ACM SIGGRAPH 2004 Papers*, ACM, SIGGRAPH '04, 861–869.
- KRAEVOY, V., SHEFFER, A., SHAMIR, A., AND COHEN-OR, D. 2008. Non-homogeneous resizing of complex models. In *ACM SIGGRAPH Asia 2008 papers*, ACM, SIGGRAPH Asia '08, 111:1–111:9.
- KRAEVOY, V., JULIUS, D., AND SHEFFER, A. 2007. Model composition from interchangeable components. In *Proc. of Pacific Graphics*, IEEE Computer Society, 129–138.
- KRISHNAN, S., LEE, P. Y., MOORE, J. B., AND VENKATASUBRAMANIAN, S. 2005. Global registration of multiple 3d point sets via optimization-on-a-manifold. *SGP '05*.
- KRIZHEVSKY, A., SUTSKEVER, I., AND HINTON, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *Proc. NIPS*.
- KUHN, H., AND TUCKER, A. 1951. Nonlinear programming. In *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, University of California Press, Berkeley, California, J. Neyman, Ed., 481–492.
- KUMAR, M. P., KOLMOGOROV, V., AND TORR, P. H. S. 2009. An analysis of convex relaxations for MAP estimation of discrete MRFs. *Journal of Machine Learning Research* 10, 71–106.
- LAFFERTY, J. D., MCCALLUM, A., AND PEREIRA, F. C. N. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 282–289.
- LAGA, H., MORTARA, M., AND SPAGNUOLO, M. 2013. Geometry and context for semantic correspondences and functionality recognition in man-made 3d shapes. *ACM Trans. Graph.* 32, 5.
- LAI, Y.-K., HU, S.-M., MARTIN, R. R., AND ROSIN, P. L. 2008. Fast mesh segmentation using random walks. In *Proceedings of the 2008 ACM symposium on Solid and physical modeling*, ACM, SPM '08, 183–191.
- LAI, K., BO, L., AND FOX, D. 2013. Unsupervised feature learning for 3d scene labeling. In *Proc. IEEE Int. Conf. on Rob. and Auto.*
- LEE, J., AND FUNKHOUSER, T. 2008. Sketch-based search and composition of 3d models. In *Proc. SBIM*.
- LEE, Y. J., ZITNICK, L., AND COHEN, M. 2011. Shadowdraw: Real-time user guidance for freehand drawing. *ACM Trans. Graph.* 30, 4, 27:1–27:9.
- LENSCH, H. P. A., KAUTZ, J., GOESELE, M., HEIDRICH, W., AND SEIDEL, H.-P. 2003. Image-based reconstruction of spatial appearance and geometric detail. *ACM Trans. Graph.* 22, 2, 234–257.
- LEORDEANU, M., AND HEBERT, M. 2005. A spectral technique for correspondence problems using pairwise constraints. *ICCV '05*, 1482–1489.
- LEORDEANU, M., AND HEBERT, M. 2006. Efficient map approximation for dense energy functions. 545–552.
- LEVIN, A., FERGUS, R., DURAND, F., AND FREEMAN, W. T. 2007. Image and depth from a conventional camera with a coded aperture. In *ACM SIGGRAPH 2007 papers*, ACM, SIGGRAPH '07.
- LEVY, B., AND ZHANG, H. 2011. Elements of geometry processing. In *Invited SIGGRAPH Asia Courses*.
- LI, M., LANGBEIN, F. C., AND MARTIN, R. R. 2006. Constructing regularity feature trees for solid models. In *Proc. Geometric Modeling and Processing; LNCS*, Springer, 267–286.

- LI, H., SUMNER, R. W., AND PAULY, M. 2008. Global correspondence optimization for non-rigid registration of depth scans. *Computer Graphics Forum* 27, 5, 1421–1430.
- LI, H., WEISE, T., AND PAULY, M. 2010. Example-based facial rigging. *ACM Trans. Graph.* 29, 4, 32:1–32:6.
- LI, Y., WU, X., CHRYSATHOU, Y., SHARF, A., COHEN-OR, D., AND MITRA, N. J. 2011. Globfit: consistently fitting primitives by discovering global relations. In *ACM SIGGRAPH 2011 papers*, ACM, SIGGRAPH ’11, 52:1–52:12.
- LI, B., GODIL, A., AONO, M., BAI, X., FURUYA, T., LI, L., LOPEZ-SASTRE, R., JOHAN, H., OHBUCHI, R., REDONDO-CABRERA, C., TATSUMA, A., YANAGIMACHI, T., AND ZHANG, S. 2012. Shrec’12 track: Generic 3d shape retrieval. *Eurographics Workshop on 3D Object Retrieval*.
- LI, H., YU, J., YE, Y., AND BREGLER, C. 2013. Realtime facial animation with on-the-fly correctives. *ACM Trans. Graph.* 32, 4, 42:1–42:10.
- LI, B., LU, Y., LI, C., GODIL, A., SCHRECK, T., AONO, M., CHEN, Q., CHOWDHURY, N. K., FANG, B., FURUYA, T., JOHAN, H., KOSAKA, R., KOYANAGI, H., OHBUCHI, R., AND TATSUMA, A. 2014. Shrec’14 track: Large scale comprehensive 3d shape retrieval. *Eurographics Workshop on 3D Object Retrieval*, 131–140.
- LI, Y., DAI, A., GUIBAS, L., AND NIESSNER, M. 2015. Database-assisted object retrieval for real-time 3d reconstruction. In *Computer Graphics Forum*, vol. 34.
- LI, Y., SU, H., QI, C. R., FISH, N., COHEN-OR, D., AND GUIBAS, L. 2015. Joint embeddings of shapes and images via cnn image purification. *ACM Trans. Graph.*.
- LI, S. Z. 2009. *Markov Random Field Modeling in Image Analysis*, 3rd ed. Springer Publishing Company, Incorporated.
- LIN, H.-Y. S., LIAO, H.-Y. M., AND LIN, J.-C. 2007. Visual salience-guided mesh decomposition. *IEEE Transactions on Multimedia* 9, 1, 46–57.
- LIN, Z., CHEN, M., AND MA, Y. 2011. The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices.
- LIPMAN, Y., AND FUNKHOUSER, T. 2009. Möbius voting for surface correspondence. *ACM Trans. Graph.* 28, 3.
- LIPMAN, Y., SORKINE, O., LEVIN, D., AND COHEN-OR, D. 2005. Linear rotation-invariant coordinates for meshes. In *ACM SIGGRAPH 2005 Papers*, ACM, SIGGRAPH ’05, 479–487.
- LIPP, M., WONKA, P., AND WIMMER, M. 2008. Interactive visual editing of grammars for procedural architecture. In *ACM SIGGRAPH 2008 papers*, ACM, SIGGRAPH ’08, 102:1–102:10.
- LITMAN, R., BRONSTEIN, A. M., BRONSTEIN, M. M., AND CASTELLANI, U. 2014. Supervised learning of bag-of-features shape descriptors using sparse coding. *SGP*.
- LIU, R., AND 0002, H. Z. 2007. Mesh segmentation via spectral embedding and contour analysis. *Comput. Graph. Forum* 26, 3, 385–394.
- LIU, Y., LIN, W.-C., AND HAYS, J. 2004. Near-regular texture analysis and manipulation. In *ACM SIGGRAPH 2004 Papers*, ACM, New York, NY, USA, SIGGRAPH ’04, 368–376.
- LIU, Y., POTTMANN, H., WALLNER, J., YANG, Y.-L., AND WANG, W. 2006. Geometric modeling with conical meshes and developable surfaces. In *ACM SIGGRAPH 2006 Papers*, ACM, New York, NY, USA, SIGGRAPH ’06, 681–689.
- LIU, Y., JIN, R., AND YANG, L. 2006. Semi-supervised multi-label learning by constrained non-negative matrix factorization. *AAAI’06*, 421–426.
- LIU, S., MARTIN, R. R., LANGBEIN, F. C., AND ROSIN, P. L. 2007. Segmenting periodic reliefs on triangle meshes. In *IMA Conference on the Mathematics of Surfaces*, 290–306.
- LIU, W., WANG, J., AND CHANG, S.-F. 2012. Robust and scalable graph-based semisupervised learning. *Proceedings of the IEEE* 100, 9, 2624–2638.
- LIU, T., CHAUDHURI, S., KIM, V., HUANG, Q.-X., MITRA, N. J., AND FUNKHOUSER, T. 2014. Creating consistent scene graphs using a probabilistic grammar. *ACM Trans. Graph.* 33, 6, to appear.
- LIU, T., HERTZMANN, A., LI, W., AND FUNKHOUSER, T. 2015. Style compatibility for 3d furniture models. *ACM Trans. Graph.* 34, 4, 85:1–85:9.
- LJUPČO TODOROVSKI, S. D. 2006. Integrating knowledge-driven and data-driven approaches to modeling. *Ecological Modelling* 194, 1–3, 3–13.
- LLOYD, S. P. 1982. Least squares quantization in pcm. *IEEE Transactions on Information Theory* 28, 129–137.
- LOEFF, N., FARHADI, A., ENDRES, I., AND FORSYTH, D. 2009. Unlabeled data improves word prediction. In *ICCV’09*, 956–962.
- LOPER, M. M., MAHMOOD, N., AND BLACK, M. J. 2014. MoSh: Motion and shape capture from sparse markers. *ACM Trans. Graph.* 33, 6, 220:1–220:13.
- LUENBERGER, D. G., AND YE, Y. 2008. *Linear and Nonlinear Programming, Third Edition*. Springer.
- LUN, Z., KALOGERAKIS, E., AND SHEFFER, A. 2015. Elements of style: Learning perceptual shape style similarity. *ACM Trans. Graph.* 34, 4, 84:1–84:14.
- LV, J., CHEN, X., HUANG, J., AND BAO, H. 2012. Semi-supervised mesh segmentation and labeling. *Comp. Graph. Forum* 31, 7–2.
- MA, C., HUANG, H., SHEFFER, A., KALOGERAKIS, E., AND WANG, R. 2014. Analogy-driven 3d style transfer. *Computer Graphics Forum* 33, 2, 175–184.
- MANGAN, A. P., AND WHITAKER, R. T. 1999. Partitioning 3d surface meshes using watershed segmentation. *IEEE Trans. Vis. Comput. Graph.* 5, 308–321.
- MANNING, C. D., RAGHAVAN, P., AND SCHÜTZE, H. 2008. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.
- MARANDE, W., AND BURGER, G. 2007. Mitochondrial dna as a genomic jigsaw puzzle. *Science* 318, 415.
- MARKOVSKY, I. 2012. *Low Rank Approximation: Algorithms, Implementation, Applications*. Springer.
- MARTINET, A., SOLER, C., HOLZSCHUCH, N., AND SILLION, F. X. 2006. Accurate detection of symmetries in 3d shapes. *ACM Trans. Graph.* 25, 439–464.
- MARTINET, A. 2007. *Structuring 3D Geometry based on Symmetry and Instancing Information*. These, Institut National Polytechnique de Grenoble - INPG.

- MATTAUSCH, O., PANZOZO, D., MURA, C., SORKINE-HORNUNG, O., AND PAJAROLA, R. 2014. Object detection and classification from large-scale cluttered indoor scans. *Computer Graphics Forum* 33, 2.
- MCCLURE, S. G., AND D.E. 1987. Statistical methods for tomographic image reconstruction. In *Proceedings of the 46th Session of the International Statistical Institute, Bulletin of the ISI*, 52.
- MÉMOLI, F., AND SAPIRO, G. 2005. A theoretical and computational framework for isometry invariant recognition of point cloud data. *Foundations of Computational Mathematics* 5, 3, 313–347.
- MERRELL, P., SCHKUFZA, E., LI, Z., AGRAWALA, M., AND KOLTUN, V. 2011. Interactive furniture layout using interior design guidelines. *ACM Trans. Graph.* 30, 4, 87:1–87:9.
- MERRELL, P. 2007. Example-based model synthesis. In *Proc. I3D*, 105–112.
- MILLER, G. A. 1995. Wordnet: A lexical database for english. *Communications of the ACM* 38, 39–41.
- MIN, P., HALDERMAN, A., KAZHDAN, M., AND FUNKHOUSER, T. 2003. Early experiences with a 3D model search engine. In *Web3D Symposium*.
- MIN, P., KAZHDAN, M., AND FUNKHOUSER, T. 2004. A comparison of text and shape matching for retrieval of online 3D models. In *European Conference on Digital Libraries*.
- MITCHELL, J. S. B., MOUNT, D. M., AND PAPADIMITRIOU, C. H. 1987. The discrete geodesic problem. *SIAM J. Comput.* 16, 4.
- MITRA, N. J., GUIBAS, L. J., AND PAULY, M. 2006. Partial and approximate symmetry detection for 3d geometry. *SIGGRAPH '06*, 560–568.
- MITRA, N. J., BRONSTEIN, A. M., AND BRONSTEIN, M. M. 2010. Intrinsic regularity detection in 3d geometry. In *Computer Vision - ECCV 2010, 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part III*, 398–410.
- MITRA, N., WAND, M., ZHANG, H., COHEN-OR, D., KIM, V., AND HUANG, Q.-X. 2014. Structure-aware shape processing. *SIGGRAPH Course*.
- MOREELS, P., AND PERONA, P. 2007. Evaluation of features detectors and descriptors based on 3d objects. *Int. J. Comput. Vision* 73, 3, 263–284.
- MÜLLER, P., WONKA, P., HAEGLER, S., ULMER, A., AND VAN GOOL, L. 2006. Procedural modeling of buildings. *ACM Trans. Graph.*, 614–623.
- MÜLLER, P., ZENG, G., WONKA, P., AND VAN GOOL, L. 2007. Image-based procedural modeling of facades. In *ACM SIGGRAPH 2007 papers*, ACM, New York, NY, USA, SIGGRAPH '07.
- MURRAY, S. O., KERSTEN, D., OLSHAUSEN, B. A., SCHRATER, P., AND WOODS, D. L. 2002. Shape perception reduces activity in human primary visual cortex. *Proc. Natl. Acad. Sci.* 99, 23, 15164–15169.
- NAN, L., XIE, K., AND SHARF, A. 2012. A search-classify approach for cluttered indoor scene understanding. *ACM Trans. Graph.* 31, 6, 137:1–137:10.
- NATTERER, J., BURGER, N., AND MÜLLER, A. 2002. The roof structure “Expodach” at theworld exhibition Hannover. In *Proc. 5th Intl. Conf. Space Structures*, 185–193.
- NG, A. Y., JORDAN, M. I., AND WEISS, Y. 2001. On spectral clustering: Analysis and an algorithm. In *ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS*, MIT Press, vol. 14, 849–856.
- NG, A. Y. 2004. Feature selection, l1 vs. l2 regularization, and rotational invariance. In *ICML*.
- NGUYEN, G. P., AND WORRING, M. 2008. Interactive access to large image collections using similarity-based visualization. *J. Visual Lang. Comp.* 19, 2, 203–224.
- NGUYEN, A., BEN-CHEN, M., WELNICKA, K., YE, Y., AND GUIBAS, L. 2011. An optimization approach to improving collections of shape maps. *Computer Graphics Forum* 30, 5, 1481–1491.
- NOCEDAL, J., AND WRIGHT, S. J. 2006. *Numerical Optimization*. Springer.
- NOVOTNI, M., AND KLEIN, R. 2003. 3d zernike descriptors for content based shape retrieval. *solid modeling*.
- OHBUCHI, R., AND FURUYA, T. 2010. Distance metric learning and feature combination for shape-based 3d model retrieval. In *Proceedings of the ACM Workshop on 3D Object Retrieval*, ACM, New York, NY, USA, 3DOR '10, 63–68.
- OSADA, R., FUNKHOUSER, T., CHAZELLE, B., AND DOBKIN, D. 2002. Shape distributions. *ACM Trans. Graph.* 21, 807–832.
- OVSJANIKOV, M., SUN, J., AND GUIBAS, L. J. 2008. Global intrinsic symmetries of shapes. *Comput. Graph. Forum* 27, 5, 1341–1348.
- OVSJANIKOV, M., MÉRIGOT, Q., MÉMOLI, F., AND GUIBAS, L. J. 2010. One point isometric matching with the heat kernel. *Computer Graphics Forum* 29, 5, 1555–1564.
- OVSJANIKOV, M., LI, W., GUIBAS, L., AND MITRA, N. J. 2011. Exploration of continuous variability in collections of 3d shapes. *ACM Trans. Graph.* 30, 4, 33:1–33:10.
- OVSJANIKOV, M., BEN-CHEN, M., SOLOMON, J., BUTSCHER, A., AND GUIBAS, L. 2012. Functional maps: A flexible representation of maps between shapes. *ACM Trans. Graph.* 31, 4, 30:1–30:11.
- OVSJANIKOV, M., BEN-CHEN, M., SOLOMON, J., BUTSCHER, A., AND GUIBAS, L. 2012. Functional maps: A flexible representation of maps between shapes. *ACM Trans. Graph.* 31, 4.
- OVSJANIKOV, M., BEN-CHEN, M., CHAZAL, F., AND GUIBAS, L. J. 2013. Analysis and visualization of maps between shapes. *Comput. Graph. Forum* 32, 6, 135–145.
- PACHAURI, D., KONDOR, R., AND SINGH, V. 2013. Solving the multi-way matching problem by permutation synchronization. In *Proc. NIPS*, 1860–1868.
- PALMER, S. E. 1999. *Vision Science: Photons to Phenomenology*. MIT Press.
- PAN, S. J., AND YANG, Q. 2010. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.* 22, 10, 1345–1359.
- PARK, M., BROCKLEHURST, K., COLLINS, R. T., AND LIU, Y. 2009. Deformed lattice detection in real-world images using

- mean-shift belief propagation. *IEEE Trans. Pattern Anal. Mach. Intell.* 31, 1804–1816.
- PAULOVICH, F., ELER, D., POCO, J., BOTHA, C., MINGHIM, R., AND NONATO, L. 2011. Piecewise laplacian-based projection for interactive data exploration and organization. *Computer Graphics Forum* 30, 3, 1091–1100.
- PAULY, M., KEISER, R., AND GROSS, M. 2003. Multi-scale feature extraction on point-sampled surfaces. *Comput. Graph. Forum (Proc. Eurographics)* 22, 3, 281–290.
- PAULY, M., MITRA, N. J., GIESEN, J., GROSS, M., AND GUIBAS, L. J. 2005. Example-based 3d scan completion. In *Proc. Symp. on Geom. Proc.*, SGP ’05.
- PAULY, M., MITRA, N. J., GIESEN, J., GROSS, M., AND GUIBAS, L. J. 2005. Example-based 3d scan completion. In *Proc. Symp. on Geom. Proc.*, SGP ’05.
- PAULY, M., MITRA, N. J., WALLNER, J., POTTMANN, H., AND GUIBAS, L. J. 2008. Discovering structural regularity in 3d geometry. In *ACM SIGGRAPH 2008 papers*, ACM, New York, NY, USA, SIGGRAPH ’08, 43:1–43:11.
- PEKELNY, YURI, GOTSMAN, AND CRAIG. 2008. Articulated object reconstruction and markerless motion capture from depth video. *Computer Graphics Forum* 27, 2, 399–408.
- PIRAZZI, C., AND WEINAND, Y. 2006. Geodesic lines on free-form surfaces:optimized grids for timber rib shells. In *Proc. World Conference on Timber Engineering*. 7pp.
- PODOLAK, J., SHILANE, P., GOLOVINSKIY, A., RUSINKIEWICZ, S., , AND FUNKHOUSER, T. 2006. A planar-reflective symmetry transform for 3d shapes. *ACM Trans. Graph.* 25, 3.
- PODOLAK, J., SHILANE, P., GOLOVINSKIY, A., RUSINKIEWICZ, S., AND FUNKHOUSER, T. 2006. A planar-reflective symmetry transform for 3d shapes. In *ACM SIGGRAPH 2006 Papers*, ACM, New York, NY, USA, SIGGRAPH ’06, 549–559.
- PONS-MOLL, G., ROMERO, J., MAHMOOD, N., AND BLACK, M. 2015. Dyna: A model of dynamic human shape in motion. *ACM Trans. Graph.* 34, 4, to appear.
- POTTMANN, H., LIU, Y., WALLNER, J., BOBENKO, A., AND WANG, W. 2007. Geometry of multi-layer freeform structures for architecture. In *ACM SIGGRAPH 2007 papers*, ACM, New York, NY, USA, SIGGRAPH ’07.
- POTTMANN, H., SCHIFTNER, A., BO, P., SCHMIEDHOFER, H., WANG, W., BALDASSINI, N., AND WALLNER, J. 2008. Freeform surfaces from single curved panels. In *ACM SIGGRAPH 2008 papers*, ACM, New York, NY, USA, SIGGRAPH ’08, 76:1–76:10.
- POTTMANN, H., WALLNER, J., HUANG, Q.-X., AND YANG, Y.-L. 2009. Integral invariants for robust geometry processing. *Comput. Aided Geom. Des.* 26, 37–60.
- POTTMANN, H., HUANG, Q., DENG, B., SCHIFTNER, A., KILIÁN, M., GUIBAS, L., AND WALLNER, J. 2010. Geodesic patterns. In *ACM SIGGRAPH 2010 papers*, ACM, New York, NY, USA, SIGGRAPH ’10, 43:1–43:10.
- QUATTTONI, A., AND TORRALBA, A. 2009. Recognizing indoor scenes. In *Proc. CVPR*, 413–420.
- RAND, W. M. 1971. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association* 66, 336, 846–850.
- RAVIKUMAR, P., AGARWAL, A., AND WAINWRIGHT, M. J. 2010. Message-passing for graph-structured linear programs: Proximal methods and rounding schemes. *J. Mach. Learn. Res.* 11, 1043–1080.
- RAY, N., LI, W. C., LÉVY, B., SHEFFER, A., AND ALLIEZ, P. 2006. Periodic global parameterization. *ACM Trans. Graph.* 25, 1460–1485.
- REN, X., AND MALIK, J. 2003. Learning a classification model for segmentation. In *Proceedings of the Ninth IEEE International Conference on Computer Vision - Volume 2*, IEEE Computer Society, Washington, DC, USA, ICCV ’03, 10–17.
- REN, X., AND RAMANAN, D. 2013. Histograms of sparse codes for object detection. In *Proc. CVPR*, 3246–3253.
- RESENDE, M. G. C., RAMAKRISHNAN, K. G., AND DREZNER, Z. 1995. Computing lower bounds for the quadratic assignment problem with an interior point algorithm for linear programming. *Operations Research* 43, 781–791.
- RIFKIN, R., AND KLAUTAU, A. 2004. In defense of one-vs-all classification. *J. Mach. Learn. Res.* 5, 101–141.
- ITCHIE, D. W., AND KEMP, G. J. L. 1999. Protein docking using spherical polar fourier correlations. *Proteins* 39, 178–194.
- ROBERTS, R., SINHA, S. N., SZELISKI, R., AND STEEDLY, D. 2011. Structure from motion for scenes with large duplicate structures. In *Proc. CVPR*, 3137–3144.
- ROTHER, C., MINKA, T., BLAKE, A., AND KOLMOGOROV, V. 2006. Cosegmentation of image pairs by histogram matching - incorporating a global constraint into mrfs. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1*, IEEE Computer Society, Washington, DC, USA, 993–1000.
- RUBNER, Y., TOMASI, C., AND GUIBAS, L. J. 2000. The earth mover’s distance as a metric for image retrieval. *Int. J. Comput. Vision* 40, 2, 99–121.
- RUSSELL, B., EFROS, A., SIVIC, J., FREEMAN, W., AND ZISSERMAN, A. 2009. Segmenting scenes by matching image composites. *Advances in Neural Information Processing Systems*, 1–9.
- RUSTAMOV, R. M., OVSJANIKOV, M., AZENCOT, O., BEN-CHEN, M., CHAZAL, F., AND GUIBAS, L. 2013. Map-based exploration of intrinsic shape differences and variability. *ACM Trans. Graph.* 32, 4, 72:1–72:12.
- SALAS-MORENO, R. F., NEWCOMBE, R. A., STRASDAT, H., KELLY, P. H. J., AND DAVISON, A. J. 2013. SLAM++: simultaneous localisation and mapping at the level of objects. In *Proc. CVPR*, IEEE, 1352–1359.
- SATKIN, S., LIN, J., AND HEBERT, M. 2012. Data-driven scene understanding from 3d models. In *Proc. BMVC*, 128:1–128:11.
- SAUPE, D., AND VRANIC, D. V. 2001. 3d model retrieval with spherical harmonics and moments. In *Proceedings of the 23rd DAGM-Symposium on Pattern Recognition*, 392–397.
- SAXENA, A., SUN, M., AND NG, A. Y. 2009. Make3d: Learning 3d scene structure from a single still image. *IEEE Trans. Pattern Anal. Mach. Intell.* 31, 5, 824–840.
- SCHAFFER, S., AND YUKSEL, C. 2007. Example-based skeleton extraction. In *Proc. Symp. on Geom. Proc.*, 153–162.

- SCHAEFER, S., AND YUKSEL, C. 2007. Example-based skeleton extraction. In *Proc. Symp. on Geom. Proc.*, 153–162.
- SCHAFFALITZKY, F., AND ZISSERMAN, A. 1999. Geometric grouping of repeated elements within images. In *Shape, Contour and Grouping in Computer Vision*, Springer-Verlag, London, UK, 165–181.
- SCHLESINGER, D., AND FLACH, B. 2006. Transforming an arbitrary minsum problem into a binary one. *Dresden University of Technology Tech Rep*, Technical Report TUD-FI06-01, 1–16.
- SCHNEIDMAN-DUHOVNY, D., INBAR, Y., NUSSINOV, R., AND WOLFSON, H. J. 2005. Geometry-based flexible and symmetric protein docking. *Proteins* 60, 2, 224–231.
- SCHREINER, J., ASIRVATHAM, A., PRAUN, E., AND HOPPE, H. 2004. Inter-surface mapping. In *ACM SIGGRAPH 2004 Papers*, ACM, New York, NY, USA, SIGGRAPH ’04, 870–877.
- SCHRIJVER, A. 1986. *Theory of linear and integer programming*. John Wiley & Sons, Inc., New York, NY, USA.
- SCHULZ, A., SHAMIR, A., LEVIN, D. I. W., SITTHI-AMORN, P., AND MATUSIK, W. 2014. Design and fabrication by example. *ACM Trans. Graph.* 33, 4.
- SECORD, A., LU, J., FINKELSTEIN, A., SINGH, M., AND NEALEN, A. 2011. Perceptual models of viewpoint preference. *ACM Trans. Graph.* 30, 5.
- SEDERBERG, T. W., AND PARRY, S. R. 1986. Free-form deformation of solid geometric models. *SIGGRAPH ’86*, 151–160.
- SHAKHNAROVICH, G., VIOLA, P., AND DARRELL, T. 2003. Fast pose estimation with parameter-sensitive hashing. In *Proceedings of the Ninth IEEE International Conference on Computer Vision - Volume 2*, IEEE Computer Society, Washington, DC, USA, ICCV ’03, 750–.
- SHAMIR, A. 2008. A survey on mesh segmentation techniques. *Comput. Graph. Forum* 27, 6, 1539–1556.
- SHAO, T., XU, W., ZHOU, K., WANG, J., LI, D., AND GUO, B. 2012. An interactive approach to semantic modeling of indoor scenes with an rgbd camera. *ACM Trans. Graph.* 31, 6, 136:1–136:11.
- SHAPIRA, L., SHAMIR, A., AND COHEN-OR, D. 2008. Consistent mesh partitioning and skeletonisation using the shape diameter function. *Vis. Comput.* 24, 249–259.
- SHAPIRA, L., SHALOM, S., SHAMIR, A., ZHANG, R. H., AND COHEN-OR, D. 2010. Contextual Part Analogies in 3D Objects. *International Journal of Computer Vision* 89, 2, 309–315.
- SHARF, A., LEWINER, T., SHKLARSKI, G., TOLEDO, S., AND COHEN-OR, D. 2007. Interactive topology-aware surface reconstruction. In *ACM SIGGRAPH 2007 papers*, ACM, SIGGRAPH ’07.
- SHEN, C.-H., FU, H., CHEN, K., AND HU, S.-M. 2012. Structure recovery by part assembly. *ACM Trans. Graph.* 31, 6, 180:1–180:11.
- SHI, J., AND MALIK, J. 2000. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* 22, 888–905.
- SHILANE, P., AND FUNKHOUSER, T. 2007. Distinctive regions of 3d surfaces. *ACM Trans. Graph.* 26, 2.
- SHILANE, P., MIN, P., KAZHDAN, M., AND FUNKHOUSER, T. 2004. The princeton shape benchmark. In *Proc. SMI*, IEEE Computer Society, Washington, DC, USA, 167–178.
- SHILANE, P., MIN, P., KAZHDAN, M., AND FUNKHOUSER, T. 2004. The Princeton shape benchmark. In *Shape Modeling International*.
- SHLAFMAN, S., TAL, A., AND KATZ, S. 2002. Metamorphosis of polyhedral surfaces using decomposition. *Comput. Graph. Forum* 21, 3.
- SHOTTON, J., WINN, J., ROTHER, C., AND CRIMINISI, A. 2009. Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *Int. J. Comput. Vision* 81, 2–23.
- SHOTTON, J., FITZGIBBON, A., COOK, M., SHARP, T., FINOCCHIO, M., MOORE, R., KIPMAN, A., AND BLAKE, A. 2011. Real-time human pose recognition in parts from single depth images. In *Proc. CVPR*.
- SIDI, O., VAN KAICK, O., KLEIMAN, Y., ZHANG, H., AND COHEN-OR, D. 2011. Unsupervised co-segmentation of a set of shapes via descriptor-space spectral clustering. *ACM Trans. Graph.* 30, 6, 126:1–126:10.
- SILBERMAN, N., KOHLI, P., HOIEM, D., AND FERGUS, R. 2012. Indoor segmentation and support inference from rgbd images. In *Proc. ECCV*.
- SIMARI, P., KALOGERAKIS, E., AND SINGH, K. 2006. Folding meshes: hierarchical mesh segmentation based on planar symmetry. In *Proc. Symp. on Geom. Proc.*, SGP ’06, 111–119.
- SIMARI, P. D., NOWROUZEZHABRAI, D., KALOGERAKIS, E., AND SINGH, K. 2009. Multi-objective shape segmentation and labeling. *Computer Graphics Forum* 28, 5, 1415–1425.
- SIMS, K. 1991. Artificial evolution for computer graphics. In *Proc. of SIGGRAPH*, 319–328.
- SINGER, A., AND WU, H.-T. 2011. Vector diffusion maps and the connection laplacian.
- SNAVELY, N., SEITZ, S. M., AND SZELISKI, R. 2006. Photo tourism: Exploring image collections in 3d. *ACM Trans. Graph.* 25, 3, 835–846.
- SNAVELY, N., SEITZ, S. M., AND SZELISKI, R. 2006. Photo tourism: Exploring photo collections in 3d. *ACM Trans. Graph.* 25, 3, 835–846.
- SOCHER, R., HUVAL, B., BHAT, B., MANNING, C. D., AND NG, A. Y. 2012. Convolutional-recursive deep learning for 3d object classification. In *Proc. NIPS*, 665–673.
- SOLomon, J., BEN-CHEN, M., BUTSCHER, A., AND GUIBAS, L. J. 2011. Discovery of intrinsic primitives on triangle meshes. *Comput. Graph. Forum* 30, 2, 365–374.
- SOLomon, J., NGUYEN, A., BUTSCHER, A., BEN-CHEN, M., AND GUIBAS, L. 2012. Soft maps between surfaces. In *Proc. SGP 2012*.
- SOLomon, J., NGUYEN, A., BUTSCHER, A., BEN-CHEN, M., AND GUIBAS, L. 2012. Soft maps between surfaces. *Computer Graphics Forum* 31, 5, 1617–1626.
- SOLomon, J. 2015. *Numerical algorithms*. AK Peters/CRC Press.
- SONG, S., AND XIAO, J. 2014. Sliding shapes for 3D object detection in depth images. In *Proc. ECCV*.
- SONTAG, D., AND JAAKKOLA, T. 2009. Tree block coordinate descent for MAP in graphical models. *Journal of Machine Learning Research - Proceedings Track 5*, 544–551.

- SORKINE, O., AND ALEXA, M. 2007. As-rigid-as-possible surface modeling. In *Proc. Symp. on Geom. Proc.*, 109–116.
- SORKINE, O., COHEN-OR, D., LIPMAN, Y., ALEXA, M., RÖSSL, C., AND SEIDEL, H.-P. 2004. Laplacian surface editing. In *Proc. Symp. on Geom. Proc.*, ACM, New York, NY, USA, SGP '04, 175–184.
- SPUYBROEK, L. 2004. *NOX: Machining Architecture*. Thames & Hudson.
- STAVA, O., BENES, B., MECH, R., ALIGA, D., AND KRISTOF, P. 2010. Inverse procedural modeling by automatic generation of L-systems. *Computer Graphics Forum* 29, 2, 665–674.
- SU, H., HUANG, Q.-X., MITRA, N. J., LI, Y., AND GUIBAS, L. J. 2014. Estimating image depth using shape collections. *ACM Trans. Graph.* 33, 4, 37:1–37:11.
- SU, H., MAJI, S., KALOGERAKIS, E., AND LEARNED-MILLER, E. 2015. Multi-view convolutional neural networks for 3d shape recognition. In *Proc. ICCV*.
- SU, H., QI, C. R., LI, Y., AND GUIBAS, L. J. 2015. Render for cnn: Viewpoint estimation in images using cnns trained with rendered 3d model views. In *The IEEE International Conference on Computer Vision (ICCV)*.
- SU, H., SAVVA, M., YI, L., CHANG, A. X., SONG, S., YU, F., LI, Z., XIAO, J., HUANG, Q., SAVARESE, S., FUNKHOUSER, T., HANRAHAN, P., AND GUIBAS, L. J., 2015. ShapeNet: An information-rich 3d model repository. <http://shapenet.cs.stanford.edu>.
- SU, H., NIESSNER, M., DAI, A., YAN, M., AND GUIBAS, L. 2016. Volumetric and multi-view cnns for object classification on 3d data. *CVPR*.
- SUMNER, R. W., AND POPOVIĆ, J. 2004. Deformation transfer for triangle meshes. *ACM Trans. Graph.* 23, 3, 399–405.
- SUMNER, R. W., SCHMID, J., AND PAULY, M. 2007. Embedded deformation for shape manipulation. In *ACM SIGGRAPH 2007 papers*, ACM, New York, NY, USA, SIGGRAPH '07.
- SUN, J., OVSJANIKOV, M., AND GUIBAS, L. 2009. A concise and provably informative multi-scale signature based on heat diffusion. In *SGP*, 1383–1392.
- SUN, J., OVSJANIKOV, M., AND GUIBAS, L. 2009. A concise and provably informative multi-scale signature based on heat diffusion. In *Computer Graphics Forum*, 1383–1392.
- SUNDAR, H., SILVER, D., N., G., AND DICKENSON, S. 2004. Skeleton based shape matching and retrieval. *shape modeling international*, 130–139.
- SUNG, M., KIM, V. G., ANGST, R., AND GUIBAS, L. 2015. Data-driven structural priors for shape completion. *ACM Trans. Graph.*
- SURAZHSKY, V., SURAZHSKY, T., KIRSANOV, D., GORTLER, S. J., AND HOPPE, H. 2005. Fast exact and approximate geodesics on meshes. In *ACM SIGGRAPH 2005 Papers*, ACM, New York, NY, USA, SIGGRAPH '05, 553–560.
- SUTTON, C., AND MCCALLUM, A. 2005. Piecewise training of undirected models. In *Proc. of UAI*.
- SWADZBA, A., AND WACHSMUTH, S. 2010. Indoor scene classification using combined 3D and gist features. In *Proc. ACCV*, 201–215.
- SZEGEDY, C., LIU, W., JIA, Y., SERMANET, P., REED, S., ANGUELOV, D., ERHAN, D., VANHOUCKE, V., AND RABINOVICH, A. 2015. Going deeper with convolutions. In *Proc. CVPR*.
- SZELISKI, R., ZABIH, R., SCHARSTEIN, D., VEKSLER, O., KOLMOGOROV, V., AGARWALA, A., TAPPEN, M., AND ROTHER, C. 2008. A comparative study of energy minimization methods for Markov random fields with smoothness-based priors. *IEEE Trans. Pattern Anal. Mach. Intell.* 30, 6, 1068–1080.
- TALTON, J. O., GIBSON, D., YANG, L., HANRAHAN, P., AND KOLTUN, V. 2009. Exploratory modeling with collaborative design spaces. *ACM Trans. Graph.* 28, 5.
- TALTON, J. O., GIBSON, D., YANG, L., HANRAHAN, P., AND KOLTUN, V. 2009. Exploratory modeling with collaborative design spaces. *ACM Trans. Graph.* 28, 5, 167:1–167:10.
- TALTON, J. O., LOU, Y., LESSER, S., DUKE, J., MĚCH, R., AND KOLTUN, V. 2011. Metropolis procedural modeling. *ACM Trans. Graph.* 30, 11:1–11:14.
- TALTON, J., YANG, L., KUMAR, R., LIM, M., GOODMAN, N., AND MĚCH, R. 2012. Learning design patterns with bayesian grammar induction. 63–74.
- TANGELDER, J. W. H., AND VELTKAMP, R. C. 2008. A survey of content based 3d shape retrieval methods. *Multimedia Tools and Applications* 39, 3, 441–471.
- TANGELDER, J. W., AND VELTKAMP, R. C. 2008. A survey of content based 3d shape retrieval methods. *Multimedia Tools and Applications* 39, 3, 441–471.
- THOMPSON, D. W. 1945. *On Growth and Form*. Cambridge University Press.
- TONG, Y., ALLIEZ, P., COHEN-STEINER, D., AND DESBRUN, M. 2006. Designing quadrangulations with discrete harmonic forms. In *Proc. Symp. on Geom. Proc.*, SGP '06, 201–210.
- TORRALBA, A., MURPHY, K. P., AND FREEMAN, W. T. 2004. Sharing features: efficient boosting procedures for multiclass object detection. In *Proceedings of the 2004 IEEE computer society conference on Computer vision and pattern recognition*, IEEE Computer Society, Washington, DC, USA, CVPR'04, 762–769.
- TORRALBA, A., MURPHY, K. P., AND FREEMAN, W. T. 2004. Sharing features: efficient boosting procedures for multiclass object detection. *CVPR'04*, 762–769.
- TORRALBA, A., MURPHY, K. P., AND FREEMAN, W. T. 2007. Sharing Visual Features for Multiclass and Multiview Object Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* 29, 5.
- TORRALBA, A., FERGUS, R., AND FREEMAN, W. T. 2008. 80 million tiny images: a large database for non-parametric object and scene recognition. *IEEE Trans. Pat. Ana. & Mach. Int.* 30, 11, 1958–1970.
- TRIMBLE, 2014. Trimble 3D warehouse. <http://sketchup.google.com/3Dwarehouse/>.
- TROPP, J. A. 2012. User-friendly tail bounds for sums of random matrices. *Found. Comput. Math.* 12, 4, 389–434.
- TSANG, S., BALAKRISHNAN, R., SINGH, K., AND RANJAN, A. 2004. A suggestive interface for image guided 3d sketching. In *CHI*, 591–598.

- TSOLI, A., MAHMOOD, N., AND BLACK, M. J. 2014. Breathing life into shape: Capturing, modeling and animating 3d human breathing. *ACM Trans. Graph.* 33, 4, 52:1–52:11.
- TSOUMAKAS, G., AND KATAKIS, I. 2007. Multi-label classification: An overview. *Int J Data Ware. and Mining* 2007, 1–13.
- TUYTELAARS, T., TURINA, A., AND VAN GOOL, L. 2003. Non-combinatorial detection of regular repetitions under perspective skew. *IEEE Trans. Pattern Anal. Mach. Intell.* 25, 418–432.
- UMETANI, N., IGARASHI, T., AND MITRA, N. J. 2012. Guided exploration of physically valid shapes for furniture design. *ACM Trans. Graph.* 31, 4.
- VALIANT, L. G. 1984. A theory of the learnable. *Commun. ACM* 27, 11.
- VAN KAICK, O., TAGLIASACCHI, A., SIDI, O., ZHANG, H., COHEN-OR, D., WOLF, L., AND HAMARNEH, G. 2011. Prior knowledge for part correspondence. *Computer Graphics Forum* 30, 2, 553–562.
- VAN KAICK, O., TAGLIASACCHI, A., SIDI, O., ZHANG, H., COHEN-OR, D., WOLF, L., AND HAMARNEH, G. 2011. Prior knowledge for shape correspondence. *Computer Graphics Forum* 30, 2, 553–562.
- VAN KAICK, O., ZHANG, H., HAMARNEH, G., AND COHEN-OR, D. 2011. A survey on shape correspondence. *Comput. Graph. Forum* 30, 6, 1681–1707.
- VAN KAICK, O., XU, K., ZHANG, H., WANG, Y., SUN, S., SHAMIR, A., AND COHEN-OR, D. 2013. Co-hierarchical analysis of shape structures. *ACM Trans. Graph.* 32, 4, 69:1–10.
- VICENTE, S., KOLMOGOROV, V., AND ROTHER, C. 2010. Cosegmentation revisited: models and optimization. In *Proceedings of the 11th European conference on Computer vision: Part II*, Springer-Verlag, Berlin, Heidelberg, ECCV’10, 465–479.
- WAINWRIGHT, M. J., JAAKKOLA, T., AND WILLSKY, A. S. 2005. MAP estimation via agreement on trees: message-passing and linear programming. *IEEE Transactions on Information Theory* 51, 11, 3697–3717.
- WAND, M., JENKE, P., HUANG, Q., BOKELOH, M., GUIBAS, L., AND SCHILLING, A. 2007. Reconstruction of deforming geometry from time-varying point clouds. In *Proc. Symp. on Geom. Proc.*, 49–58.
- WAND, M., ADAMS, B., OVSJANIKOV, M., BERNER, A., BOKELOH, M., JENKE, P., GUIBAS, L., SEIDEL, H.-P., AND SCHILLING, A. 2009. Efficient reconstruction of nonrigid shape and motion from real-time 3d scanner data. *ACM Trans. Graph.* 28, 2, 15:1–15:15.
- WANG, R. Y., AND POPOVIĆ, J. 2009. Real-time hand-tracking with a color glove. *ACM Trans. Graph.*, 63:1–63:8.
- WANG, L., AND SINGER, A. 2013. Exact and stable recovery of rotations for robust synchronization. *Information and Inference* 2, 2, 145–193.
- WANG, H., WEXLER, Y., OFEK, E., AND HOPPE, H. 2008. Factoring repeated content within and among images. In *ACM Trans. Graph.*, 14:1–14:10.
- WANG, Y., ASAIFI, S., VAN KAICK, O., ZHANG, H., COHEN-OR, D., AND CHEN, B. 2012. Active co-analysis of a set of shapes. *ACM Trans. Graph.* 31, 6, 165:1–165:10.
- WANG, F., HUANG, Q., AND GUIBAS, L. 2013. Image co-segmentation via consistent functional maps. In *Proc. ICCV*.
- WANG, Y., GONG, M., WANG, T., COHEN-OR, D., ZHANG, H., AND CHEN, B. 2013. Projective analysis for 3d shape segmentation. *ACM Trans. Graph.* 32, 6, 192:1–192:12.
- WANG, F., HUANG, Q., OVSJANIKOV, M., AND GUIBAS, L. 2014. Unsupervised multi-class joint image segmentation. In *Proc. CVPR*.
- WANG, F., HUANG, Q., OVSJANIKOV, M., AND GUIBAS, L. 2014. Unsupervised multi-class joint image segmentation. In *Proc. CVPR*.
- WATANABE, K., AND BELYAEV, A. G. 2001. Detection of salient curvature features on polygonal surfaces. *Comput. Graph. Forum (Proc. Eurographics)* 20, 3.
- WEI, L.-Y., LEFEBVRE, S., KWATRA, V., AND TURK, G. 2009. State of the art in example-based texture synthesis. In *Eurographics State of the Art Report*, 93–117.
- WEI, L., HUANG, Q., CEYLAN, D., VOUGA, E., AND LI, H. 2016. Dense human body correspondence using convolutional network. *CVPR*.
- WEINBERGER, K. Q., AND SAUL, L. K. 2009. Distance metric learning for large margin nearest neighbor classification. *J. Mach. Learn. Res.* 10, 207–244.
- WEISE, T., LI, H., VAN GOOL, L., AND PAULY, M. 2009. Face/off: Live facial puppetry. In *Proc. of Symp. on Comp. Anim.*, 7–16.
- WEISE, T., BOUAZIZ, S., LI, H., AND PAULY, M. 2011. Realtime performance-based facial animation. *ACM Trans. Graph.* 30, 4, 77:1–77:10.
- WEN, Z., GOLDFARB, D., AND YIN, W. 2010. Alternating direction augmented Lagrangian methods for semidefinite programming. *Math. Prog. Comput.* 2, 3-4, 203–230.
- WERNER, T. 2007. A linear programming approach to max-sum problem: A review. *IEEE Trans. Pattern Anal. Mach. Intell.* 29, 1165–1179.
- WESSEL, R., BLÜMEL, I., AND KLEIN, R. 2009. A 3d shape benchmark for retrieval and automatic classification of architectural data. In *Eurographics 2009 Workshop on 3D Object Retrieval*, 53–56.
- WILEY, D. F., AMENTA, N., ALCANTARA, D. A., GHOSH, D., KIL, Y. J., DELSON, E., HARCOURT-SMITH, W., ROHLF, F. J., JOHN, K. S., AND HAMANN, B. 2005. Evolutionary morphing. In *Proceedings of IEEE Visualization 2005*.
- WINN, J., AND JOJIC, N. 2005. Locus: Learning object classes with unsupervised segmentation. In *Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV’05) Volume 1 - Volume 01*, IEEE Computer Society, Washington, DC, USA, 756–763.
- WONKA, P., WIMMER, M., SILLION, F., AND RIBARSKY, W. 2003. Instant architecture. In *ACM SIGGRAPH 2003 Papers*, ACM, New York, NY, USA, SIGGRAPH ’03, 669–677.
- WU, J., AND KOBBELT, L. 2005. Structure recovery via hybrid variational surface approximation. *Comput. Graph. Forum* 24, 3, 277–284.

- WU, T.-P., TANG, C.-K., BROWN, M. S., AND SHUM, H.-Y. 2007. Shapepalettes: Interactive normal transfer via sketching. *ACM Trans. Graph.* 26, 3.
- WU, T.-P., SUN, J., TANG, C.-K., AND SHUM, H.-Y. 2008. Interactive normal reconstruction from a single image. In *ACM SIGGRAPH Asia 2008 Papers*, SIGGRAPH Asia '08, 119:1–119:9.
- WU, H., WANG, Y.-S., FENG, K.-C., WONG, T.-T., LEE, T.-Y., AND HENG, P.-A. 2010. Resizing by symmetry-summarization. In *ACM SIGGRAPH Asia 2010 papers*, ACM, New York, NY, USA, SIGGRAPH ASIA '10, 159:1–159:10.
- WU, Z., SONG, S., KHOSLA, A., YU, F., ZHANG, L., TANG, X., AND XIAO, J. 2015. 3d shapenets: A deep representation for volumetric shapes. In *Proc. CVPR*.
- XIANG, Y., MOTTAGHI, R., AND SAVARESE, S. 2014. Beyond pascal: A benchmark for 3d object detection in the wild. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*.
- XIANG, Y., SONG, C., MOTTAGHI, R., AND SAVARESE, S. 2014. Monocular multiview object tracking with 3d aspect parts. In *European Conference on Computer Vision (ECCV)*.
- XIE, X., XU, K., MITRA, N. J., COHEN-OR, D., GONG, W., SU, Q., AND CHEN, B. 2013. Sketch-to-design: Context-based part assembly. *Comp. Graph. Forum* 32, 8, 233–245.
- XIE, Z., XU, K., LIU, L., AND XIONG, Y. 2014. 3d shape segmentation and labeling via extreme learning machine. *Computer Graphics Forum* 33, 5.
- XIE, Z., XU, K., SHAN, W., LIU, L., XIONG, Y., AND HUANG, H. 2015. Projective feature learning for 3d shapes with multi-view depth images. to appear.
- XIN, S.-Q., AND WANG, G.-J. 2009. Improving chen and han's algorithm on the discrete geodesic problem. *ACM Trans. Graph.* 28, 104:1–104:8.
- XU, K., ZHANG, H., TAGLIASACCHI, A., LIU, L., LI, G., MENG, M., AND XIONG, Y. 2009. Partial intrinsic reflectional symmetry of 3d shapes. In *ACM SIGGRAPH Asia 2009 papers*, ACM, New York, NY, USA, SIGGRAPH Asia '09, 138:1–138:10.
- XU, W., WANG, J., YIN, K., ZHOU, K., VAN DE PANNE, M., CHEN, F., AND GUO, B. 2009. Joint-aware manipulation of deformable models. In *ACM SIGGRAPH 2009 papers*, ACM, New York, NY, USA, SIGGRAPH '09, 35:1–35:9.
- XU, K., LI, H., ZHANG, H., COHEN-OR, D., XIONG, Y., AND CHENG, Z.-Q. 2010. Style-content separation by anisotropic part scales. *ACM Trans. Graph.* 29, 5, 184:1–184:10.
- XU, K., ZHENG, H., ZHANG, H., COHEN-OR, D., LIU, L., AND XIONG, Y. 2011. Photo-inspired model-driven 3d object modeling. *ACM Trans. Graph.* 30, 4, 80:1–80:10.
- XU, K., ZHANG, H., COHEN-OR, D., AND CHEN, B. 2012. Fit and diverse: Set evolution for inspiring 3d shape galleries. *ACM Trans. Graph.* 31, 4, 57:1–57:10.
- XU, K., CHEN, K., FU, H., SUN, W.-L., AND HU, S.-M. 2013. Sketch2scene: Sketch-based co-retrieval and co-placement of 3d models. *ACM Trans. Graph.* 32, 4, 123:1–123:12.
- XU, K., CHEN, K., FU, H., SUN, W.-L., AND HU, S.-M. 2013. Sketch2scene: Sketch-based co-retrieval and co-placement of 3d models. *ACM Trans. Graph.* 32, 4, 123:1–123:12.
- XU, K., MA, R., ZHANG, H., ZHU, C., SHAMIR, A., COHEN-OR, D., AND HUANG, H. 2014. Organizing heterogeneous scene collections through contextual focal points. *ACM Trans. Graph.* 33, 4, 35:1–12.
- XU, W., SHI, Z., XU, M., ZHOU, K., WANG, J., ZHOU, B., WANG, J., AND YUAN, Z. 2014. Transductive 3d shape segmentation using sparse reconstruction. *Comp. Graph. F.* 33, 5.
- XU, K., HUANG, H., SHI, Y., LI, H., LONG, P., CAICHEN, J., SUN, W., AND CHEN, B. 2015. Autoscanning for coupled scene reconstruction and proactive object analysis. *ACM Trans. Graph.* 34, 6, 177:1–177:14.
- XU, K., KIM, V., Q. HUANG, N. M., AND KALOGERAKIS, E., 2016. Wikipage: Data-driven shape analysis and processing. <http://wp.cs.umass.edu/datadrivenshape/>.
- YAN, D.-M., LIU, Y., AND WANG, W. 2006. Quadric surface extraction by variational shape approximation. In *Proc. GMP*, vol. 4077, 73–86.
- YANG, L., AND JIN, R. 2006. Distance metric learning: A comprehensive survey.
- YAO, B., KHOSLA, A., AND FEI-FEI, L. 2011. Combining randomization and discrimination for fine-grained image categorization. In *Proc. CVPR*, 1577–1584.
- YAO, B., KHOSLA, A., AND FEI-FEI, L. 2011. Combining randomization and discrimination for fine-grained image categorization. In *Proc. CVPR*.
- YE, Y. 1991. An $O(n^3l)$ potential reduction algorithm for linear programming. *Math. Program.* 50, 239–258.
- YI, L., KIM, V. G., CEYLAN, D., SHEN, I.-C., YAN, M., SU, H., LU, C., HUANG, Q., SHEFFER, A., AND GUIBAS, L. 2016. A scalable active framework for region annotation in 3d shape collections. *SIGGRAPH Asia*.
- YU, K., AND NG, A. 2010. Feature learning for image classification. In *ECCV tutorials*.
- YUAN, M., AND LIN, Y. 2006. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society, Series B* 68, 49–67.
- YUAN, M., AND LIN, Y. 2006. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society, Series B* 68, 49–67.
- YUMER, M. E., AND KARA, L. B. 2012. Co-abstraction of shape collections. *ACM Trans. Graph.* 31, 6, 166:1–166:11.
- YUMER, M. E., AND KARA, L. B. 2012. Co-abstraction of shape collections. *ACM Trans. Graph.* 31.
- YUMER, M., AND KARA, L. 2014. Co-constrained handles for deformation in shape collections. *ACM Trans. Graph.* 32, 6, to appear.
- YUMER, M. E., AND MITRA, N. J. 2016. Learning semantic deformation flows with 3d convolutional networks. In *European Conference on Computer Vision (ECCV 2016)*, Springer, –.
- YUMER, M. E., CHUN, W., AND MAKADIA, A. 2014. Co-segmentation of textured 3d shapes with sparse annotations. In *Proc. CVPR*.
- YUMER, M. E., CHAUDHURI, S., HODGINS, J. K., AND KARA, L. B. 2015. Semantic shape editing using deformation handles. *ACM Trans. Graph.* 34.

- ZACH, C., KLOPSCHITZ, M., AND POLLEFEYS, M. 2010. Disambiguating visual relations using loop constraints. In *Proc. CVPR*.
- ZAHARESCU, A., BOYER, E., VARANASI, K., AND HORAUD, R. 2009. Surface feature detection and description with applications to mesh matching. In *Proc. CVPR [DBL 2009]*, 373–380.
- ZEILER, M. D., AND FERGUS, R. 2014. Visualizing and understanding convolutional neural networks. In *Proc. ECCV*.
- ZHA, Z.-J., MEI, T., WANG, J., WANG, Z., AND HUA, X.-S. 2009. Graph-based semi-supervised learning with multiple labels. *J. Vis. Commun. Image Represent.* 20, 2, 97–103.
- ZHANG, R., TSAI, P.-S., CRYER, J. E., AND SHAH, M. 1999. Shape from shading: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* 21, 8, 690–706.
- ZHANG, E., MISCHAIKOW, K., AND TURK, G. 2005. Feature-based Surface Parameterization and Texture Mapping. *ACM Trans. Graph.* 24, 1.
- ZHANG, H., SHEFFER, A., COHEN-OR, D., ZHOU, Q., VAN KAICK, O., AND TAGLIASACCHI, A. 2008. Deformation-driven shape correspondence. *Comput. Graph. Forum* 27, 5, 1431–1439.
- ZHANG, Y., XU, W., TONG, Y., AND ZHOU, K. 2014. Online structure analysis for real-time indoor scene reconstruction. *ACM Trans. Graph..*
- ZHAO, X., WANG, H., AND KOMURA, T. 2014. Indexing 3d scenes using the interaction bisector surface. *ACM Trans. Graph.* 33, 3, 22:1–22:15.
- ZHOU, K., HUANG, X., WANG, X., TONG, Y., DESBRUN, M., GUO, B., AND SHUM, H.-Y. 2006. Mesh quilting for geometric texture synthesis. In *ACM SIGGRAPH 2006 Papers*, ACM, New York, NY, USA, SIGGRAPH ’06, 690–697.
- ZHOU, S., FU, H., LIU, L., COHEN-OR, D., AND HAN, X. 2010. Parametric reshaping of human bodies in images. *ACM Trans. Graph.* 29, 4, 126:1–126:10.
- ZHU, L., ZHOU, Z., AND HU, D. 2008. Globally consistent reconstruction of ripped-up documents. *IEEE Trans. Pattern Anal. Mach. Intell.* 30, 1, 1–13.
- ZHU, X. 2006. Semi-supervised learning literature survey. *Computer Sciences TR 1530, University of Wisconsin Madison.*
- ZIA, M. Z., STARK, M., SCHIELE, B., AND SCHINDLER, K. 2013. Detailed 3d representations for object recognition and modeling. *IEEE Trans. Pat. Ana. & Mach. Int.* 35, 11, 2608–2623.