Elements of Style: Learning Perceptual Shape Style Similarity

Zhaoliang Lun[1] Evangelos Kalogerakis[1] Alla Sheffer[2]

[1]University of Massachusetts Amherst [2]University of British Columbia

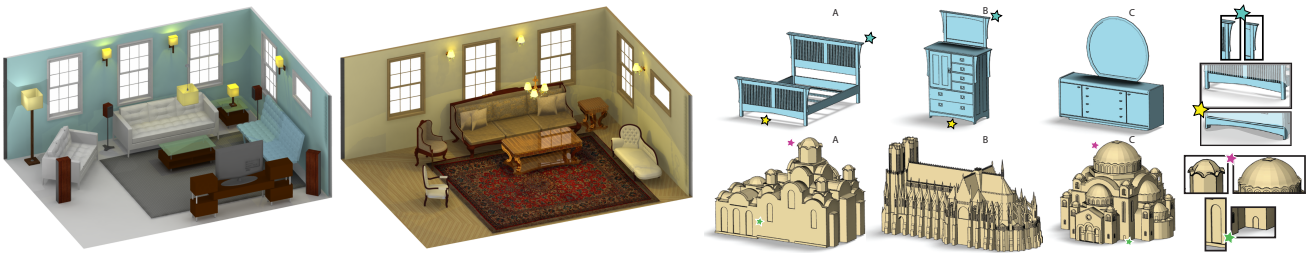

Figure 1: *(left) Changing the style of objects in a scene influences the sense of time and place. (right) Style similarity transcends structure: in the top row, the bed A is pronouncedly more similar, style-wise, to dresser B than C; in the bottom row, buildings A and C are stylistically more similar (insets highlight some stylistically similar elements).*

Abstract

The human perception of stylistic similarity transcends structure and function: for instance, a bed and a dresser may share a common style. An algorithmically computed style similarity measure that mimics human perception can benefit a range of computer graphics applications. Previous work in style analysis focused on shapes within the same class, and leveraged structural similarity between these shapes to facilitate analysis. In contrast, we introduce the first *structure-transcending* style similarity measure and validate it to be well aligned with human perception of stylistic similarity. Our measure is inspired by observations about style similarity in art history literature, which point to the presence of similarly shaped, salient, *geometric elements* as one of the key indicators of stylistic similarity. We translate these observations into an algorithmic measure by first quantifying the geometric properties that make humans perceive geometric elements as similarly shaped and salient in the context of style, then employing this quantification to detect pairs of matching style related elements on the analyzed models, and finally collating the element-level geometric similarity measurements into an object-level style measure consistent with human perception. To achieve this consistency we employ crowdsourcing to quantify the different components of our measure; we learn the relative perceptual importance of a range of elementary shape distances and other parameters used in our measurement from 50K responses to cross-structure style similarity queries provided by over 2500 participants.We train and validate our method on this dataset, showing it to successfully predict relative style similarity with near 90% accuracy based on 10-fold cross-validation.

CR Categories: I.3.5 [Computing Methodologies]: Computer Graphics—Computational Geometry and Object Modeling;

Keywords: style similarity, crowdsourcing, machine learning

1 Introduction

Human perception of style similarity transcends structure and function; we can meaningfully discuss style similarity between a cup and a coffee pot, a bed and a dresser, or a church and a castle. Style coordination across heterogeneous object arrangements greatly contributes to their overall aesthetics, and significantly improves the believability of virtual scenes (Figure 1, left). Thus, when designing both real and virtual environments, artists and designers put significant effort into generating style coordinated object arrangements at all scales - from putting together a place-setting at a table, through room furnishing, and all the way to design of building ensembles and cityscapes. This task requires users to navigate heterogeneous object databases based on style similarity. Such style-based database navigation can be significantly accelerated by the availability of a measure that can robustly evaluate style similarity between structurally different models and detect models which share a similar style despite large functional differences (Figure 1, right). While previous work focused on evaluating style similarity between objects with similar overall structure (Section 2), we introduce the first *structure-transcending* method for style similarity evaluation between 3D shapes, and validate that it is well aligned with human perception.

Our similarity measure is motivated by observations about human perception of style hinted at by art history and appraisal literature. Art history experts often classify objects as belonging to a particular geographic or temporal style by looking at salient *geometric elements* on the objects with recurring *visual motifs* [Nutting 1928; Blumenson 1995]. For instance, classical Byzantine churches are likely to have rounded domes and arches, while Gothic structures are dominated by steep gables and flying buttresses (Figure 1, bottom row A and B). While style extends beyond the search for motif level similarity, our work focuses on the role of common motifs in style evaluation. Our style similarity metric is therefore designed around the presence of pairs of similarly shaped, or matching, salient geometric elements on the evaluated models. The relative size of such elements, their number, and the percentage of the object's surface covered by them can vary dramatically (Figure 1, right), making detection of matching elements a challenging problem that is distinctly different from partial matching or co-segmentation. We detect matching elements on the analyzed objects using a combination of bottom up clustering and top down search. We then evaluate their prevalence, their saliency, and the degree of similarity between them to generate a single style similarity measurement.

When evaluating element shape similarity and salience, we are

faced with a range of plausible geometric metrics to consider, and require a principled way to determine the relevance and importance of each metric for evaluating object-level style similarity. Since style has no unified quantifiable, objective, definition, choices made by any single individual or small group could be subjective, and hence questionable. To minimize the impact of such subjective choices we learn the impact of the different metrics on human perception of style similarity from crowdsourced data collected via a large-scale Mechanical Turk study. Since no standard scale for style similarity exists, asking participants to assign an absolute style similarity score to pairs of shapes would result in uncalibrateable data. However, we observe, and validate through experimentation, that humans are largely consistent in evaluating *relative* style-similarity: when asked if an object A is more similar style-wise to object B or C, participants overwhelmingly pick the same answer. We leverage participant consistency in answering relative style similarity queries by collecting a dataset of informative crowdsourced responses to such queries and using them to learn an algorithmic style similarity measure. We start with a measure that combines a large range of elementary shape distances and saliency measures linked to style by either art history or computer graphics literature. We then learn the relative weights of the different shape distances and saliency metrics, as well as other measurement parameters, that maximally align our resulting style similarity measure with crowdsourced majority responses. We regularize the set of weights assigned to elementary distances and style features using an L^1 norm formulation that implicitly suppresses the weights of features that the learning framework deems less important.

We validate our approach in several ways. We first confirm our foundational hypothesis that human observers can reliably answer relative style similarity queries. The analysis of our study output shows that viewers are consistent when answering such queries: different participants provide identical answers to the same query 85% of the time on average. Participant behavior is also persistent: the same participant provides the same answer when faced with variants of the same query 86% of the time. These findings indicate that humans share a common perception of stylistic similarity, highlighting the importance of developing a style measure that would agree with human perception. We then validate our algorithmically computed style measure against participant responses using ten-fold cross-validation. Our method produces answers that are similar to the participant majority answers close to 90% of the time, achieving a similar consistency to the average participant. Finally, we show a range of applications of structure-transcending style similarity computation, including stylistic suggestions for scene assembly and style-based organization of shape collections.

Contributions. Our main contribution is a structure-transcending method for evaluating the stylistic similarity of 3D shapes. Our work bridges perception and computation by addressing a problem that till now had only been tackled in a qualitative, perceptual context. The proposed measure is well aligned with the human perception of style, is motivated by art history literature, and is learned from and validated against crowdsourced data. While previous methods exist to evaluate style similarity within a particular class of objects with similar overall structure, our method is the first to enable structure-transcending style similarity evaluation.

Our approach also stands out in its use of crowdsourced data. Rather than employing potentially subjective style metrics or definitions proposed by individual researchers, as had been the case through much of the literature so far, our method derives the relative importance of a range of potentially style related geometric features to the actual human perception of style, as reflected in the participant answers to our style similarity queries. A further benefit of our crowdsourcing framework is the collection of a large trove of participant responses to style similarity questions, which we intend to make public. Our collected data contains over 50,000 responses from more than 2,500 participants, constructed from a database of over a thousand models in seven diverse categories. It is our hope that this data will facilitate a range of further studies on human perception of style.

2 Background and Related Work

We believe that the most relevant literary source for understanding human perception of 3D object style is art history and appraisal literature. These texts discuss at length the geometric features of architectural structures, furniture and other artifacts associated with particular historic or geographic styles, e.g. [Nutting 1928; Blumenson 1995; Lewis 2008], and frequently refer to characteristic "elements of design" or "motifs" to describe a particular style. For example, [Blumenson 1995] states "starting from recognizing motifs you will soon recognize styles", "The purpose of this brief guide is to provide photographic illustrations of ... architectural details, elements, and forms to enable the user to ..... recognize styles and elements". The book proceeds to describe a range of architectural styles based on the choice of architectural elements they employ, e.g. mansards, towers, or porches, as well as the characteristic shape of different building parts such as roofs or windows. Nutting [1928] similarly catalogs European and American furniture styles based on the shape of different furniture elements such as feet, trims, or posts.

The style definitions employed in this literature are descriptive rather than constructive, motivating our search for a constructive style similarity measure. Our work builds upon previous methods for shape style analysis, as well as methods for learning style measures in other types of data, discussed below.

Style analysis for same class models. A range of methods provide strategies for evaluating fine-grained similarity between shapes with similar structure [Xu et al. 2010; Kalogerakis et al. 2012; Kim et al. 2013; Huang et al. 2013; van Kaick et al. 2013; Yumer and Kara 2014]. These methods rely on the shared structure to first extract either a dense correspondence, e.g. [Kim et al. 2013; Huang et al. 2013] or a segmentation of the two models into corresponding, compatible, parts (a co-segmentation), e.g. [Xu et al. 2010; Kalogerakis et al. 2012]. They then use these correspondences to evaluate fine-grained similarity measuring either point-wise or part-wise geometric differences with respect to pre-defined distance metrics whose relative weights are either hard-coded or learned from database distribution. For instance, Xu et al. [2010] co-segment models into roughly corresponding parts and define the style distance between shapes based on differences in scales and orientations of part bounding boxes. Kalogerakis et al. [2012] define object and part styles using dominant modes of a learned probability distribution across a database of models on a range of geometric descriptors. Kim et al. [2013] and Huang et al. [2013] classify shapes within the same class, e.g. chairs, as belonging to different fine-grained categories, e.g. office or rocking chairs. Kim et al. perform the categorization by first producing a set of probabilistic part-based templates and grouping the shapes based on the template they fit best. Huang et al. group the shapes based on partial and local similarity measured using a combination of spin images, distance, and deformation fields, with the importance of each term learned from database distribution. Yumer et al. [2014] use co-analysis of shapes within the same class to learn geometric and spatial constraints among the different parts of an object, and use this information for style transfer and other applications.

In contrast to these methods, we measure style similarity between models with drastically different structures for which dense point or part correspondences do not exist (e.g. bed and dresser in Figure 1, right). Furthermore, while these methods operate on different fixed

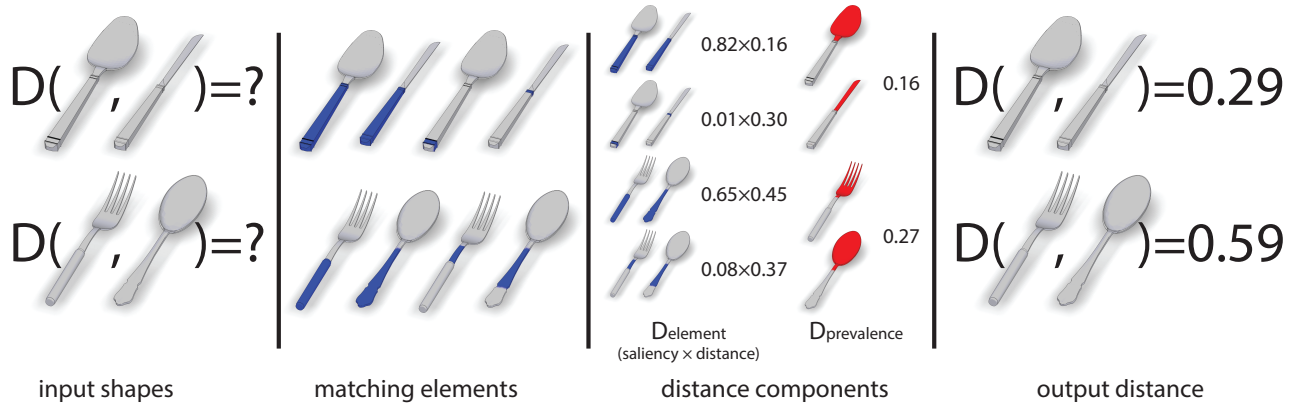

Figure 2: *To evaluate style similarity, we identify potentially matching elements which will then be used in a distance function that accounts for element similarity, element saliency and prevalence. The parameters of both steps are learned from crowdsourced perceptual similarity data.*

sets of distance functions which the authors believe to reflect style or other fine-grained object categories, we propose a style measure derived from and validated against a crowdsourced perceptual style similarity study.

Structure-transcending shape style analysis So far, little has been done when it comes to analyzing style across different structures. Li et al. [2013] highlight the difficulty in evaluating or defining structure-transcending styles for 3D objects. Thus, rather than considering style of 3D shapes, they focus on identifying styles on closed 2D curves. They segment the curves at curvature extrema, and evaluate style similarity between curves by comparing segment shapes and curvature histograms. Their conclusions highlight the need to perform a perception study to establish a style definition consistent with human intuition. Our work follows this perception-driven route while addressing the much more challenging question of analyzing style of structurally different 3D objects.

In 3D space, Ma et al. [2014] leverage analogy transformations between input objects with the same style but different structure to capture structural differences between them. They then use the extracted analogy transformations for style transfer. Their definition of style is both boolean - objects either have the same style or not - and narrow, in that they assume that objects with the same style have most of their surfaces related via patch-wise similarity transformations. We introduce and compute a continuous and flexible style similarity measure, that provides meaningful evaluation for pairs of objects with large dissimilar areas (e.g. Figure 1, right), and which correctly accounts for geometric element which share common shape characteristics, but are not scaled replicas of one another (e.g. the domes in Figure 7). More importantly, while Ma et al., employ their definition of style to process user inputs that are *a priori* assumed to conform with this definition, we address the actual evaluation of style similarity given an arbitrary pair of models.

Learning style in other domains. There is a significant effort to analyze style in other types of data, such as images, video and audio. Tenenbaum and Freeman [2000] discuss ways to separate content and style factors in speech, typography, and face images. Significant effort has been made in learning style parameters from exemplar images and video and transferring them to other instances [Hertzmann et al. 2001; Bonneel et al. 2013]. Researchers have addressed style analysis, recognition and retrieval in 2D images [Willats and Durand 2005; Hurtut et al. 2011], music [Aucouturier and Pachet 2002], and film [Bell and Koren 2007]. Doersch et al [2012] recognize the visual motifs, or style elements, that distinguish photos taken in different cities.

We differ from these works in the domain of application as well as in the use of crowdsourced data to both facilitate and validate our style similarity measure. Our work is closer to that of Garces et al. [2014], who employ crowdsourcing to learn a similarity measure for clip art styles. However, we target 3D shapes which require a very different style definition and a distinct measurement approach.

3 Overview

Our goal is to obtain a structure-transcending style similarity measure for man-made 3D shapes (Figure 2). While the notion of style extends beyond shape, we consider a purely geometry based measure; for most modeling applications properties such as texture can be easily changed once a shape is available. Moreover shape databases frequently contain only geometric information, making a measure which contains other properties less useful. Our framework for computing a style similarity measure consists of the key components outlined below.

Element Similarity. We first develop a method to measure element-level similarity in the context of style evaluation. Our measure is inspired by observations in art history literature about the types of geometric criteria that play a role in style identification (Section 4.1).

Matching Elements. We use this measurement method within a matching algorithm that detects similar geometric elements on the evaluated objects. We do not know the size or location of these elements *a priori*, and in contrast to existing co-segmentation frameworks, do not expect most of the object's surfaces to be covered by pairs of matching elements. For instance, the table and ceiling lamps in Figure 8 have similar shades, but the rest of their geometry bears little similarity to one another . Comparing all pairs of regions at all resolutions across the models would be prohibitively time consuming. We make the problem tractable by noting that since matching elements are expected to have similar shape characteristics, they should at least approximately map to one another using an affine transformation. We also observe that such elements have similar internal geometry and are frequently visually separable from the surrounding surface. We therefore first search for paired regions on the processed models that satisfy the approximate mapping requirement. We then group neighboring region pairs together based on geometric similarity, both within each pair of regions, and

between adjacent regions (Section 4.2).

Combined Style Measure. We seek a measure that reflects both the degree of similarity between the detected matching elements as well as the percentage of the surface area on both models covered with similar elements - the larger the matched area the more stylistically similar the objects should be. Our overall style similarity measure balances these two terms.

Man-made shapes often contain large functional surfaces - for instance buildings, independent of style, tend to have large areas of flat vertical walls, and dishes (cups, sugar-bows, or milk jugs) frequently have a similarly shaped cavity designed to contain food or drink. To obtain a reliable style similarity measure we seek to downplay the presence or absence of similar functional surfaces. We note that, in contrast to style related elements, which are designed to be noticeable, or salient, functional surfaces are typically more simple and nondescript. We therefore incorporate saliency into our style measure as discussed in Section 4.3.

Learning. In each of the three steps above we face multiple parameter choices, such as "how to weigh different elementary distances when evaluating element similarity?", "how to decide when elements are similar enough for matching purposes?", or "how to evaluate saliency in the context of style measurement?" As we aim to obtain parameter values that lead to a style measure consistent with human perception, we elect to learn these parameters by studying human responses to style similarity queries and algorithmically tuning the parameters to best mimic these responses. The learned parameters include (i) weights of elementary distances (geometric features) used for evaluating similarity between shape elements; (ii) parameters of the combined style measure, including weights of saliency metrics used to determine the relative distinctiveness of different surface areas from a style perspective; and (iii) parameters of the element matching algorithm. Our training step is based on participant responses to relative style similarity queries, which we describe next, and is designed to maximize the agreement between our measure and participant responses. In analyzing the responses, we took into account both the percentages of participants that selected each answer, and the reliability of individual participants (the percentage of queries in which each participant agreed with the majority response).

Study of Style Perception. Our study was designed to achieve two goals. We wanted to examine our hypothesis that human perception of style similarity between differently structured objects is consistent. We also aimed to use the study results to facilitate parameter learning for our style measurement algorithm. As already noted, asking participants to assign an absolute style similarity score to a pair of shapes is impractical, as no uniform style similarity scale exists. Our study therefore was designed around *relative* comparisons, with users asked to evaluate if an object A is more stylistically similar to object B or C, see Section 5. As summarized in Table 1 the study validates our hypothesis that humans are both consistent and persistent in answering such queries.

The selection of queries, detailed in Section 5, was motivated by the two goals above. To focus on structure-transcending style similarity, when the evaluated shapes could be directly classified into sub-categories based on structure (e.g. different pieces of furniture), B and C were selected to have similar structure, different from that of A. Style similarity allows for equivalence classes, or groups of equally (dis)similar objects. For instance, three pieces in a single set of cutlery are equally similar style-wise; while a pagoda, a Gothic cathedral and a Hindu temple are likely to be seen as equally dissimilar. Discovering such equivalence classes is interesting from a human perception perspective. However, for training a similarity measure relative similarity query responses are most informative, or discriminative, when providing an actual ranking, i.e. rating one pair of objects as more similar than another. Since our primary goal was to train our style similarity measure, we introduce similarity bias into our query generation, designing most of the query triplets so that two of the shapes are subjectively expected to be more stylistically similar. For details on the process and its impact on participant responses see Section 5. We pre-processed the raw participant input for training and algorithm validation, removing queries with non-discriminative majority responses and answers from participants deemed unreliable (see Section 5).

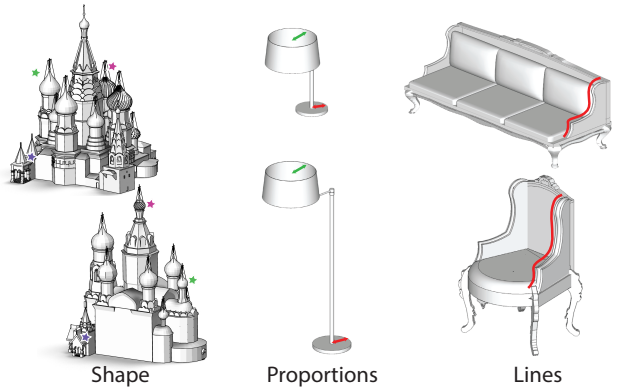

Figure 3: *Literature highlights three element-level style similarity criteria: intrinsic element geometry or shape, relative proportions or scale, and dominant curve or line shape.*

Lastly, we evaluated the reliability of the crowdsourced Mechanical Turk data via a pilot study whose participants were a combination of Mechanical Turk respondents and participants selected based on personal contacts, including domain experts, and concluded that Mechanical Turk users were sufficiently representative of the population at large; see Section 5 for details.

4 Measuring Style Similarity

Representation Man-made shapes in online databases are typically represented as partially connected meshes (polygon soups). To evaluate style similarity we densely resample these models, representing them as point clouds with normals (normal direction is set to point outward using point visibility). We assume the models to be upright oriented. Most models in online repositories, and essentially all the inputs we downloaded, satisfy this assumption. Misoriented models can be corrected using the method of Fu et al. [2008] or manually, if this method fails or is not available..

4.1 Geometric Similarity

Art-history literature [Nutting 1928; Blumenson 1995] and appraisal tutorials, e.g. [Connected Lines 2014], point to three separate geometric criteria that are useful when identifying a particular style and which are applicable across different structures: *shape*, *proportions*, and *lines* (Figure 3). This literature repeatedly stresses that objects with similar style are expected to have intrinsically similar, even if differently scaled, geometric elements - see the highlighted church domes in Figure 3, left or the skirts of the bed and dresser in Figure 1. It also indicated that relative and internal proportions of the elements play an important stylistic role - e.g. narrow vs square windows, sturdy or thin furniture legs, and so on (Figure 3, center). Finally, it emphasizes the importance of representative or noticeable surface curves in conveying style on the object's surface (Figure 3, right). Styles are often characterized by

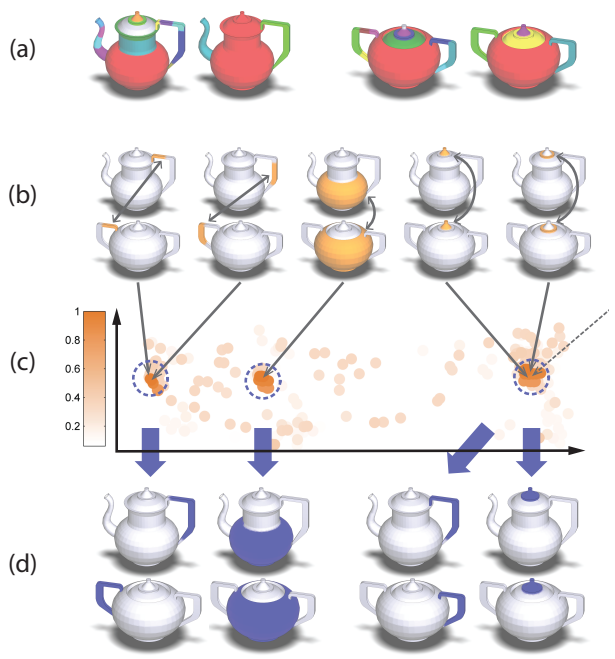

Figure 4: *Extracting matching elements (a through d): patch-segmentations (two levels visualized); example matches; transformation space (2D MDS projection) with clustering results; extracted elements.*

the use of straight versus curved, or clean versus ornate lines. While for some style comparisons all three criteria may come into play, it is the interaction or the relative weight of each criterion we seek to learn from training data.

We measure geometric similarity using elementary distances that relate to these criteria. When comparing intrinsic element geometry we employ both direct comparisons - measuring point-wise positional and normal distances computed after aligning the elements using an affine transformation, and indirect comparisons, measuring curvature distribution. We compare element proportions using their bounding box scales and shape diameter functions [Shapira et al. 2008]. To explicitly account for line similarity we detect and compare feature curves and representative silhouettes. All distances are normalized to the interval $[0, 1]$. We detail the exact distance metrics in the Appendix.

We represent the distance between two elements $\{p, p'\}$ as a weighted combination of the elementary distances, using learned distance weights w_i,

$$D(p, p') = \sum_{i=1}^{F} w_i d_i(p, p'). \quad (1)$$

4.2 Extracting Matching Elements

Given a pair of input models, we need to detect elements of one model that match, or are geometrically similar, to elements on the other and vice versa. These matching elements may not share the same exact geometry, but are expected to share similar geometric features, as measured by the geometric similarity measure above. Detecting matching elements is challenging since we do not *a priori* know the size, location, or number of such elements. While evaluating all possible pairs of patches across the two models may give us the best solution in this scenario, such computation is clearly too time consuming. We make the problem tractable by observing that geometric elements are typically self-similar: portions of the same element share similar geometry, and are frequently visually separable from the surrounding surface. Thus given a fine convex segmentation of the surface, it is reasonable to expect element boundaries to be a subset of segment boundaries. In addition, given the element similarity criteria discussed above, we can expect stylistically similar elements to *approximately* map to each other using an affine transformation. Following these observations, we first locate near-convex patches on the two models that approximately map to one another, we then locate dominant mapping transformations, and finally groups patches into elements by merging together adjacent patches that undergo a similar dominant mapping transformation. Our grouping aims to discard matched, yet dissimilar, patches and identify coherent geometric elements that share common geometric characteristics and which frequently stand apart from the surrounding surface. We satisfy these requirements by formulating grouping as a min-cut labeling problem where the distance from each patch to its matching patch is the unary term, and the geometric similarity between adjacent patches is the pairwise term. This bottom-up process is by necessity conservative and occasionally misses large weakly similar elements. We bootstrap the method to detect such large elements using a top-down search, as described below. The parameters used throughout the matching process are learned from our training data (Section 4.4). We now describe these steps in detail.

Patch Sampling. As a starting point for the matching process we sample each input model using a dense set of approximately convex patches, computed using the method of [Asafi et al. 2013] (Figure 4, a). Operating on patches, instead of individual points, significantly reduces the time complexity of our element computation. Patches also provide a more reliable starting point for matching since we can immediately evaluate match quality using our geometric distance measure, assisting further analysis. We generate patches at a number of scales by repeating the segmentation with different convexity thresholds (0.3, 0.5 and 0.7) and introduce additional larger patches by merging patches with similar shape diameter histograms [van Kaick et al. 2014]. Matching patches at different scales enables us to detect similarly shaped, but differently scaled, elements. Our method typically produces up to eighty patches per shape.

Transformation Clustering. For each patch computed in the previous step, we compute a transformation that approximately maps it to every patch on the other shape. The transformations are computed via an outlier-robust iterative closest point alignment [Besl and McKay 1992], where at each step, we compute an affine transformation aligning the two patches in a least-squares sense. To explicitly compute weakly similar yet dominant elements, we also perform outlier-robust ICP directly between the two models. The resulting transformation and the matched patches it detects are processed the same way as the located local maps. The initial alignment provides a common frame for computing elementary distances between the patches. To detect groups of adjacent patches that undergo similar transformations, we use a Hough transform based voting strategy [Ballard 1987; Mitra et al. 2006]. To imbue the transformation votes with geometric meaning, we represent each transformation as a point in a nine-dimensional space which consists of translation, rotation, and non-uniform scaling or reflection. These components of the transformation are computed via Singular Value Decomposition. Each point is assigned a confidence weight based on the shape distance between the transformed patch p and its image p':

$$\omega = \frac{A(p) + A(p')}{2} \exp\Big(- D(p, p')/s \Big), \quad (2)$$

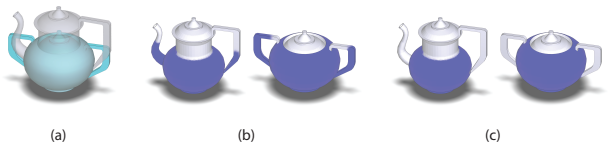

Figure 5: *Graph cut based element extraction (left to right): transformation $\mathcal{T}$ applied on input shapes; grouping using distance thresholding; grouping using graph cut.*

where $A(p)$ and $A(p')$ measure the percentage area of p and p' relative to their shape. The parameter s controls the confidence weight falloff as the distance increases, and is automatically estimated by using a grid search and selecting the value that maximizes the objective function (Equation 16) during model training (Section 4.4). To find the dominant transformations in the 9-dimensional voting space we perform mean-shift clustering. Following [Comaniciu 2003], we use the Epanechnikov kernel with adaptive bandwidth to estimate the density at each point in the voting space. We also follow Comaniciu's approach to remove low-confidence clusters representing infrequent transformations: after detecting initial cluster centers, we perturb them using small random vectors, re-execute mode detection and if any cluster centers change (indicating instability), the clusters are removed.

Each local maximum of density yields a cluster of voting transformations, and each cluster centroid corresponds to a dominant transformation that approximately maps a number of patches of one shape to the other. The transformation computation and subsequent clustering, visualized in Figure 4, are performed twice, from the first shape to the second and vice versa.

Element Extraction. We use the dominant transformations $\mathcal{T}$ found in the previous step, to compute matching elements, where each element is defined as a group of contiguous patches and the matching element is defined by the image of these patches under $\mathcal{T}$. A basic grouping strategy would be for each patch to compute the distance to its image under the transformation $\mathcal{T}$, and merge adjacent patches whose distances are deemed below some threshold into elements (Figure 5, b). However, as demonstrated in the figure, using a purely distance-based threshold ignores the expectation for elements to be distinct from the surrounding surface. To satisfy this criterion, our grouping algorithm takes into account both the distances between the individual patches and their images and the geometric similarity between patches and their neighbors. Our aim is to make similar inside/outside decisions for similar, contiguous patches. To achieve this effect we use a min-cut labeling formulation (Figure 5, c). The labeling assigns each patch p a binary label c_p which is set to 1 if the patch is added to the group and 0 otherwise. We compute the labels by minimizing the following objective function over all the patch label assignments $\mathbf{c}$ per shape:

$$E(\mathbf{c};\mathcal{T}) = \sum_p E_1(c_p;\mathcal{T}) + \sum_{p,q} \frac{1}{|\mathcal{N}(p)| + |\mathcal{N}(q)|} E_2(c_p, c_q;\mathcal{T}) \tag{3}$$

where p, q are adjacent patches, $\mathcal{N}(p)$, $\mathcal{N}(q)$ are the sets of all patches adjacent to p and q respectively. The unary term in this function assesses the distance between p and its image T_p under the transformation $\mathcal{T}$, and the pairwise term assesses how likely a pair of adjacent patches p, q is to belong to the same element. Specifically, the unary term expresses the negative logarithm of the following probability for an individual patch p:

$$P(c_p = 1;\mathcal{T}) = \exp(-D(p, T_p)/s) \tag{4}$$

thus:

$$E_1(c_p = 1;\mathcal{T}) = D(p, T_p)/s \tag{5}$$

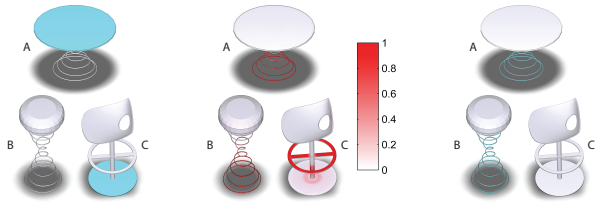

Figure 6: *Impact of saliency. Darker red regions indicate higher saliency values. (left) Ignoring saliency we would deem C stylistically closer to A than B, as the two share large similar area; taking saliency (center) into account increases the importance of the lower parts of the objects in the style similarity measure, reaching the opposite conclusion (right) which is consistent with the crowdsourced study consensus.*

and

$$E_1(c_p = 0;\mathcal{T}) = -\ln\Big[1 - \exp(-D(p, T_p)/s)\Big] \tag{6}$$

where T_p denotes all patches on the other shape that are closest to the patch p when it is transformed under the transformation $\mathcal{T}$. We use the same learned parameter s as in the clustering step.

The pairwise term expresses the negative logarithm of the probability for pairs of neighboring patches to have different binary labels based on the geometric distance between them:

$$E_2(c_p, c_q) = -[c_p \neq c_q] \ln\Big[1 - \exp\Big(-D(p, q)/s\Big)\Big] \tag{7}$$

To compute the distances between the patches, we apply a translation to align their centroids. A small distance indicates that the two patches are likely to belong to the same geometric element, and that the cost for assigning different labels to them should be high. In this case the pairwise term will encourage them to be either grouped into the element associated with T_p , or have both removed from the group depending on their unary terms. A tiny constant $\epsilon = 10^{-5}$ is added to the terms inside the above logarithms to prevent numerical issues. We compute the labeling using the standard min-cut framework [Greig et al. 1989] for each shape. Each computation yields a group of patches on one shape that approximately map to patches on the other shape under the transformation $\mathcal{T}$ and are internally similar. We perform labeling separately for the two transformation directions (from the first shape to the second and vice versa).

4.3 Combined Style Similarity Measure

Element-level Similarity Given a set $\mathcal{M}$ containing all the pairs of matching elements detected on the two input shapes, element-level similarity is computed as:

$$D_{element} = \sum_{\{p,p'\}\in\mathcal{M}} C(p, p')D(p, p') \tag{8}$$

where $D(p, p')$ is the distance between a pair of matching elements $\{p, p'\}$ on the two models, and $C(p, p')$ is the saliency of this pair of elements. As pointed out earlier, style elements are expected to be visually distinct, or salient, motivating the use of saliency to weigh the impact of individual element distances on the overall style similarity between shapes. We define saliency using a weighted combination of elementary saliency metrics suggested by recent literature [Chen et al. 2012; Leifman 2012; Shtrom et al. 2013] (see Appendix). The saliency of a pair of elements is defined as the average of their individual saliences $C(p, p') = .5[C(p) + C(p')]$, and element saliency is expressed as a weighted

sum of the saliences of its sample points. Specifically for the element p (and similarly for its matching element p'):

$$C(p) = \Big(\sum_{s \in p} \sigma\Big(\sum_{j=1}^{G} v_j x_{j,s} + v_0 \Big) / M(s) \Big) / C(S) \qquad (9)$$

where $\sigma(x) = 1/(1 + exp(-x))$ represents the sigmoid, or logistic, function, $x_{j,s}$ are the elementary saliency metrics measured at the sample point s, v_j is a learned weight per metric, and v_0 is a learned bias weight shifting the sigmoid along the input axis. The sigmoid transformation non-linearly combines the elementary saliency metrics and scales the resulting point saliences within the $[0, 1]$ range. Our experiments (Section 6) show that using this formulation to combine elementary saliency metrics is more predictive than using a simple linear combination. When elements overlap we aim to avoid counting the element distances in the overlapping region multiple times. Thus when integrating the saliency across each patch, we normalize the per-point values by the number $M(s)$ of matching elements the sample point s belongs to. We normalize the element saliency by the saliency integral across the entire input model S:

$$C(S) = \sum_{r \in S} \sigma\Big(\sum_{j=1}^{G} v_j x_{j,r} + v_0 \Big) \qquad (10)$$

Prevalence To estimate the prevalence of the matching elements we consider the percentage of the area not covered by these elements on both models. For identical input shapes, this percentage will be zero, and will increase to one if no matching elements are found. As with element similarity, we take saliency into account. If the uncovered area contains salient features, it would indicate a poorer stylistic match between shapes than if it is nondescript. We penalize unmatched areas z and z' on the two objects using the saliency integral across these areas, normalized by the saliency integral across the relevant shape

$$D_{prevalence} = .5[C(z) + C(z')] \cdot t \qquad (11)$$

where t is a learned penalty parameter.

Combined Distance Function The distance between two shapes is defined as the sum of the two terms above:

$$D = D_{elements} + D_{prevalence} \qquad (12)$$

We note that the distance between the two models is by definition symmetric. The impact of each term depends on the learned individual weights on the elementary distance and saliency metrics. We discuss the learned weights and other parameters in Section 6. We normalize the distance to $[0, 1]$ by dividing it by the penalty parameter t which defines the highest possible distance between two models when no matching elements are found. The combined function with the parameters learned as described in Section 4.4 achieved prediction accuracy of 89% on average.

4.4 Parameter Learning

The input to our parameter learning step is a set of user responses to relative similarity queries based on triplets of shapes $\{A, B, C\}$. For each query we have answers from multiple participants whether the pair of objects $\{B, A\}$ is more stylistically similar than the pair $\{C, A\}$ or vice versa. The output is a set of learned parameters (in total 99 parameters) for the distance function and the matching algorithm, which can then be used to compute style distances on other pairs of objects. We note that our problem setting is different from regression or classification, since our training data does not have the form of absolute, continuous or discrete, measurements of style. Instead, we use a probabilistic framework suited to handle relative comparisons for training. Since not all study participants are equally reliable in their answers, our training procedure weights each participant according to number of times they disagreed with the majority answer in each relative comparison.

Learning Distance Parameters For training, our model expresses the probability a participant rates $\{B, A\}$ as more similar than $\{C, A\}$, or more compactly $B_A \triangleright C_A$ as :

$$P(B_A \triangleright C_A) = \sigma\Big(D(C, A) - D(B, A) \Big) \qquad (13)$$

and similarly:

$$P(C_A \triangleright B_A) = \sigma\Big(D(B, A) - D(C, A) \Big) = 1 - P(B_A \triangleright C_A) \quad (14)$$

where $\sigma(x)$ is a sigmoid function that converts the shape distance differences into probabilities. This logistic-based probabilistic model follows [Burges et al. 2005], where it was used for learning model rankings in the context of information retrieval.

Our model contains a regularization term which can be seen as expressing a prior probability for the weights of elementary distances and saliency features to be small. Instead of standard L^2-norm we use L^1-norm regularization advocated by Tibshirani [1996]. The L^1-norm promotes sparsity by allowing some weights to dominate while pushing others toward zero. In addition, when the number of queries is smaller than the number of weights, the regularization encourages more zero weights, leading to a simpler model with better predictive performance. Our regularizer, or sparsity prior, is formulated as follows:

$$P(\mathbf{w}, \mathbf{v}, t) = \exp\Big(- \lambda_1 ||\mathbf{w}||_1 - \lambda_2 ||\mathbf{v}||_1 - \lambda_3 |t| \Big) \Big) \qquad (15)$$

where $\mathbf{w} = \{w_i\}_{i=1...F}$, $\mathbf{v} = \{v_j\}_{j=1...G}$. The regularization parameters $\lambda_1, \lambda_2, \lambda_3$ control the degree of sparsification of the model and are automatically estimated through 10-fold cross-validation on the training set. Given M training triplets, we learn the parameter values that maximize:

$$L(\mathbf{w}, \mathbf{v}, t) = \ln P(\mathbf{w}, \mathbf{v}, t) + \sum_{m=1}^{M} b[m] \cdot \ln P(B_A[m] \triangleright C_A[m]) + c[m] \cdot \ln P(C_A[m] \triangleright B_A[m]) \qquad (16)$$

where $b[m]$ and $c[m]$ represents our confidence that $B_A[m] \triangleright C_A[m]$ and $C_A[m] \triangleright B_A[m]$ respectively based on the user responses to the query m. The confidence per query is measured as follows. Each user is assigned a reliability weight that is equal to the percentage of times their answers agreed with the majority answer in the queries they were asked. The confidence $b[m]$ (and similarly $c[m]$) for a query m is measured as the sum of reliability weights for users that answered $B_A[m] \triangleright C_A[m]$ (or $C_A[m] \triangleright B_A[m]$) normalized by the total sum of reliability weights of the users who answered the query. We use bound constraints to enforce the parameters t and $\mathbf{w}$ to be positive.

Our objective function is continuously differentiable almost everywhere except when parameter values are equal to zero due to the use of the L^1-norm in our regularizer. There are several techniques that have been developed in the context of L^1-norm regularized optimization which are applicable to our problem (see [Schmidt et al. 2007; Andrew and Gao 2007] for related reviews and references). In our implementation, we experimented with two techniques, both based on Sequential Quadratic Programming (SQP) with numerical approximations for the Hessian according to the BFGS formula

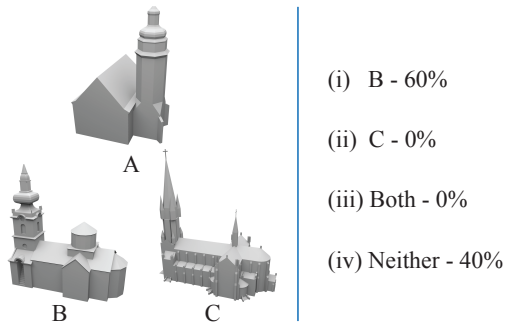

Figure 7: *(left) Study query layout. (right) response distribution for this query.*

[Nocedal and Wright 2006]. The first technique was SQP-GS [Curtis and Overton 2012], which uses gradient sampling to stabilize the optimization update steps, and has proven theoretical convergence in non-convex and non-smooth settings like ours. We also experimented with MATLAB's implementation of SQP. Even if convergence is not guaranteed with this implementation, in practice due to the use of inexact line search [Lewis and Overton 2013] and our bound constraints, the parameter updates do not run into non-differentiable values. As a result, this SQP implementation provided good optimum approximation, yielding almost identical output to SQP-GS. Our results in Section 6 are reported using this implementation which is more than an order of magnitude faster than SQP-GS. To initialize the optimization, the weights are set to small random values. Finally, we note that all elementary distances are normalized to $[0, 1]$ during training by dividing them by their $90th$ percentile value computed across all training pairs and then truncating all higher values to 1. The percentile is used instead of the maximum to discard any outlier values in the training data.

Learning Matching Parameters To learn the parameters of the overall distance function, we require the output of the matching step. However, our matching step requires an element-level distance measure to evaluate the shape differences between pairs of patches, creating a self-referential dependency between the two steps. To learn both sets of parameters we use an iterative procedure. We start with a naive distance measure generated by computing the average closest point-to-point patch distance after ICP and use this measure to detect an initial set of matching elements. We then update the parameters of the distance function by training our model with the procedure above. We repeat both steps, each time using the just learned, more reliable, distance function in transformation clustering and min-cut labeling for element matching, resulting in better matches. This iterative scheme has no convergence guarantees. However, in practice we found that style distances are improved after each iteration, when measured against human input as discusses in Section 6. In practice three iterations were sufficient for the method to converge to the results reported below. We refer the reader to the supplementary material for the values of the learned parameters for all our datasets. We also provide the source code of our implementation on our project web page.

5 Study of Style Perception

Our study tests the hypothesis that human observers are persistent and consistent in evaluating relative style similarity across structurally different objects, and provides data for training our algorithmic style similarity measure. We gathered most of our data using online questionnaires released through the Amazon Mechanical Turk (MTurk) service. We also conducted a pilot study, which we report on at the end of this section, to evaluate the reliability of MTurk versus curated participant sets.

Study Format. The queries used in our questionnaires were based on triplets of models, laid out as visualized in Figure 7, left. Subjects were asked the question "Which of the two shapes on the bottom (B or C) is more similar, style-wise, to the shape on the top (A)?" and were required to select one of the following answers: "(i) B, (ii) C, (iii) can't tell - Both B and C, (iv) can't tell - Neither B nor C".

The models used for the study were organized into seven structurally diverse categories: buildings, furniture, lamps, coffee sets, architectural columns (pillars), cutlery and dishes. This choice of categories was motivated by the online availability of diversely styled objects in these categories, and in particular the existence of coordinated scenes with multiple structurally different but similar style objects. These coordinated scenes were used to bootstrap our query generation with similarity bias, as described next. The model and study statistics are summarized in Table 1. The complete set of queries is provided as a supplementary material. To focus on structure-transcending style similarity, for categories with clear fine-grained structural sub-categories (furniture, lamps, coffee-sets and cutlery) B and C were selected to have similar structure, different from that of A (e.g two dressers and a bed, Figure 1). In categories with no clear structural sub-classes, the triplets were assembled based solely on similarity bias.

Our experiments show that humans frequently classify triplets of shapes as belonging to the same style-equivalence classes. Assembling query triplets at random results in a large number of queries (over 60% in our experiments) for which humans cannot provide a clear ranking and respond with "can't tell - Neither B nor C". While this finding is interesting from a perception perspective, such responses do little to assist our style learning algorithm. Moreover having a large percentage of such non-discriminative queries in a questionnaire causes participants to lose interest, as they no longer feel the need to focus to get a correct answer. Since our primary goal was to train our style similarity measure, we assembled most of the queries with the goal of obtaining *discriminative* responses, where participants clearly rank the degree of similarity between shapes, by introducing subjective bias. Specifically, while we generated 120 queries at random to validate the observation above, we designed the rest of the query triplets to have some *a priori* subjective similarity bias between the pairs (A, B) and (A, C). Specifically, roughly half of the study queries were constructed such that A and one of B or C were selected from a single database scene, or arrangement, (e.g. tableware set), and the remaining object was drawn from a different arrangement. We expected such queries to provide fine-grained style training and help identify what differentiates a coordinated arrangement from a random pairing. To learn coarser-level style similarity, the remaining queries were constructed such that one pair was classified by the authors as being in the same geographic or temporal style, while the third shape was subjectively classified as belonging to a different style. As expected the majority (60%) of participant responses to random queries were non-discriminative, "can't tell - Neither B nor C". For the two types of subjectively biased queries, the percentages of non-discriminative responses were 7% and 21% respectively. As expected, the subjective ranking used to generate the queries was frequently consistent with participant majority response. A tempting alternative to our study would be to use these subjective choices as-is to train the algorithm. However, this solution would, as the study shows, be inconsistent with the plurality response a significant fraction (13.7%) of the time, and would lack the additional accuracy boost we obtain by associating each plurality response with the size, or confidence, of this plurality.

Category	# Shapes	# Total Queries	# (%) Q. (iii) plurality	# (%) Q. (iv) plurality	# (%) (i) & (ii) plurality	# (%) Q. majority	**% persistence** all/reliable	**% consistency** all/reliable	**% consistency** majority	**% consistency** (i) vs (ii)
building	238	1000	0 (0.0%)	149 (14.9%)	798 (79.8%)	731 (73.1%)	73.8% / 76.7%	76.8% / 79.0%	86.6%	91.3%
furniture	278	1250	0 (0.0%)	134 (10.7%)	1088 (87.0%)	1065 (85.2%)	89.5% / 90.8%	86.0% / 87.2%	91.2%	97.4%
lamp	186	1250	1 (0.1%)	103 (8.2%)	1121 (89.7%)	1100 (88.0%)	92.5% / 93.4%	89.8% / 90.6%	94.4%	97.8%
column	74	800	0 (0.0%)	25 (3.1%)	760 (95.0%)	743 (92.9%)	86.4% / 88.8%	87.3% / 88.9%	91.7%	96.8%
coffee set	76	270	0 (0.0%)	32 (11.9%)	233 (86.3%)	224 (83.0%)	82.3% / 84.2%	83.5% / 85.0%	90.3%	94.5%
cutlery	74	200	3 (1.5%)	10 (5.0%)	184 (92.0%)	183 (91.5%)	88.1% / 89.8%	89.4% / 91.4%	93.7%	97.7%
dish	91	200	3 (1.5%)	18 (9.0%)	170 (85.0%)	162 (81.0%)	83.7% / 86.1%	78.7% / 81.6%	88.1%	92.6%
Total	1017	4970	7 (0.1%)	471 (9.5%)	4354 (87.6%)	4208 (84.7%)	85.7% / 87.7%	85.0% / 86.5%	91.3%	95.8%

Table 1: *Study statistics per category. Left to right: category, number of models, number of queries, number and percent of queries with plurality "Both B and C" response, number and percent of queries with plurality "Neither B nor C" response, number and percent of queries with plurality discriminative response, number and percent of queries with a* majority *discriminative response (majority formed by more than* 50% *participants), participant persistence across all participants and across reliable participants only, participant consistency across all participants and across reliable participants only, consistency for queries with a majority response, consistency for queries considering only discriminative responses.*

Questionnaire and Participant Information. Each questionnaire released via the Mechanical Turk contained 25 unique queries. Each question was repeated twice, with B and C flipped, to measure participant persistence. To collect a diverse set of answers per query and avoid any individual bias, we allowed each participant to complete *only one* questionnaire per category. Participants were rewarded $\$0.50$ for each questionnaire completion. Full participant statistics are reported in the Appendix.

Query Response Processing. Any large-scale study faces the risk of attracting unreliable respondents. For algorithm training and validation we detected and discarded outlier responses using a two stage filter. Participants who gave two different answers to more than 6 out of the 25 unique queries in the questionnaire, or took less than 3 minutes to complete it, were classified as unreliable and all their answers were discarded. For all other participants, we ignored non-persistent answers, where a participant answered the same question differently. The full filtering statistics are listed in the Appendix. To form a statistically significant majority we gathered answers to each query by 10 different, reliable users. For learning purposes we only used queries with a majority discriminative ((i) B or (ii) C) response. While the answer "(iii) Both B and C" could potentially be used in a learning procedure, the percentage of queries with such plurality answers is negligible (0.1%) and does not justify the extra effort required to incorporate them into the training algorithm. The number and percentage of discarded and remaining queries are listed in Table 1 columns four through six. The number and percentage of queries with discriminative majority responses used for learning are listed in Table 1 column seven.

Hypothesis Validation. We hypothesize that participants' consistency and persistence in this study can be considered as a measure of human performance for comparing the style similarity of shapes. We can measure this consistency as the percentage of times that MTurk participants' answers agree with the plurality answer per query, i.e., the percentage, or size, of the plurality. We note that this definition of plurality size takes into account all four answers. This value is shown in Table 1, ninth column. By aggregating the answers across all queries, the average plurality size is 85%. If we consider only discriminative responses the size of the plurality increases to 95.8% (last column). The fact that on average 8.5 out of 10 users agree on the response for a query confirms that human observers are consistent in evaluating relative style: if we formulate the null hypothesis that users provide a response at random given the four options per query, the probability of getting the same response from 8 or more out of 10 participants would be extremely small (p-value 0.0004 according to a binomial test), which provides strong evidence against this null hypothesis. Participant persistence is 85.7% on average, which using the same binomial test, similarly provides strong evidence against the null hypothesis.

Representative Sample. An interesting question to ask is how the consistency of MTurk participants compares to a curated participant set, and how their perception of style compares to that of experts. To answer these questions, prior to conducting our large-scale study, we performed a pilot study which included a mix of participants: 55 unique participants found through the MTurk service, 32 casual participants located based on personal contacts, and 5 Arts Ph.D. students. The last group can be considered as experts for our task. This study had 250 queries. Across the casual user group the average plurality size was 93.5% based on all answers and 99.3% excluding the non-discriminative answers. Within the expert user group, the pluralities were very similar - 95.3% and 99.5% correspondingly. For MTurk participants in this smaller study, the pluralities were 88.6% and 98.5% correspondingly, slightly smaller and similar to the ones in our large study. We conclude that the overall consistency across the different user groups is similar, and the consistency rate we observe among MTurk participants serves as a plausible estimate of such consistency in the general population. When comparing the majority responses across all three groups taking only discriminative responses into account, the percentage of times that casual users or MTurk participants disagree with the plurality answer provided by experts were negligible, 0.6%, and 1.2% respectively. In other words when participants were able to provide a ranking these rankings were essentially identical. This observation indicates that our learning method, which relies only in discriminative majority answers of MTurk participants, is likely to be consistent with expert perception of style. We observed a larger difference in the percentage of time one group of participants chose the non-discriminative 'Neither B nor C' (iv) response while the other one chose a discriminative one. Overall, MTurk participants agree with expert plurality 83% of the time, while casual participants agree with expert plurality 87% of the time. Such discrepancy is to be expected, as experts may look for different style cues beyond those noticeable by laymen. Since the difference remains small and is limited to non-descriptive answers, we believe that the MTurk participant responses can be relied on to derive an accurate picture of human perception of relative style similarity and to train a robust style similarity measure.

6 Algorithm Validation

We validate our style similarity measure by performing ten-fold cross-validation on queries with a majority discriminative response among the study participants. Queries tested during validation excluded all pairs of models present in the training queries. The percentages of queries on which our algorithm agrees with the majority response are reported in Table 2, second column. Across all categories our method agrees with the majority response 89.1% of

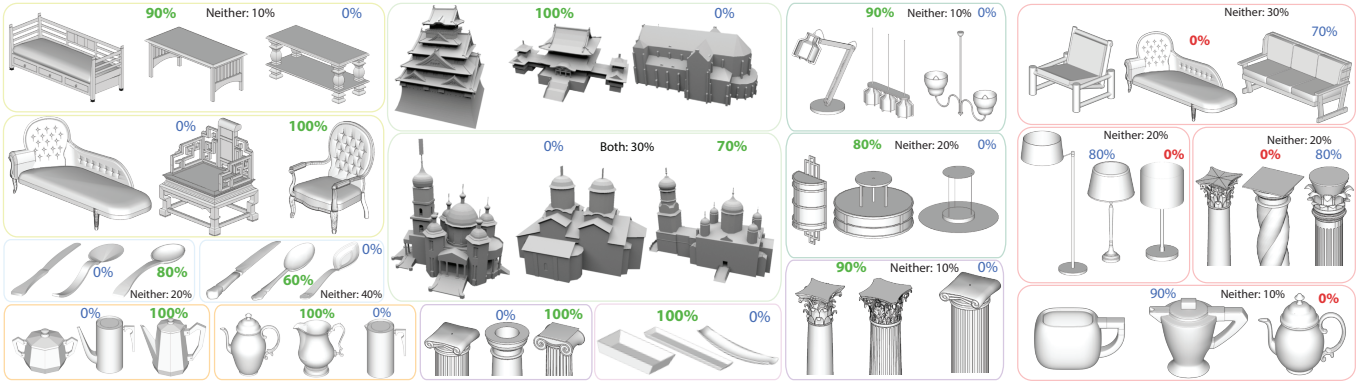

Figure 8: *(left) Examples of the 89% of queries where our method agrees with study majority response(shared answer in green), numbers show percentages of participants who selected each answer (not listed answers received zero votes). (right) Some representative failure cases (majority response blue, ours red).*

Category	**all terms**	no prevalence term	linear saliency	no saliency term	LFD
building	**81.4%**	77.4%	79.3%	79.9%	70.7%
furniture	**91.4%**	87.9%	90.8%	90.6%	74.6%
lamp	**95.0%**	86.1%	94.5%	94.7%	61.6%
column	**90.2%**	87.2%	87.8%	87.9%	55.5%
coffee set	**90.6%**	87.1%	89.3%	86.2%	62.1%
cutlery	**85.8%**	72.7%	82.5%	81.4%	61.2%
dish	**89.5%**	86.4%	87.0%	87.0%	88.9%
average	**89.1%**	83.5%	87.3%	86.8%	66.6%
mixed	**86.6%**	82.1%	86.3%	85.9%	67.8%

Table 2: *Prediction accuracy, Left to right: category, our prediction accuracy, prediction accuracy with alternate formulations. Rows one to seven show results where training and validation were done per category, bottom row shows results where both were done on the entire database.*

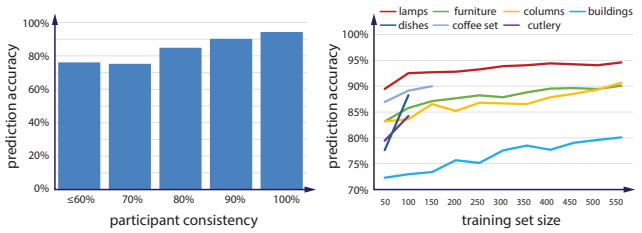

Figure 9: *(left) Prediction accuracy as a function of participant consistency; (right) prediction accuracy as a function of size of training data (for coffee sets, cutlery, and dishes the graphs terminate earlier due to the smaller number of available queries, see Table 1)*

the time. This number is comparable to the agreement level between the individual reliable participants for these queries (91.3%). We report our method's results for each query in the supplementary material. Table 2, rows one through seven, report the predictive accuracy for the scenario where the algorithm was trained and validated separately against each model category. Performing these two tasks on all the categories at once, the average accuracy slightly drops to 86.6% since, as one would expect, the importance of different measure components may vary across different object categories. As expected, the performance improves if we evaluate the method only on queries with higher participant consistency, see Figure 9, left. Prediction accuracy is measured by performing the same ten-fold cross-validation procedure and is averaged over our seven categories. When evaluated on queries with 100% participant consistency, the prediction accuracy of our method raises from 89.1% to 94.3%. Figure 9, right demonstrates the impact of decreasing the sizes of training datasets on our algorithm's performance. We note that even with just 50 queries, the performance is comparable to human consistency for two of the datasets shown (lamps and furniture).

Algorithmic Choices. We evaluated a number of alternative approaches for style measurement, summarized in Table 2, columns three to five. We evaluated the impact of dropping the prevalence term, using linear vs sigmoid saliency models, and ignoring saliency altogether. As expected each change led to drop in prediction accuracy, with the omission of prevalence leading to the largest drop. While one could expect an even larger drop without the prevalence term, such a drop is prevented by our use of approximate element matching: we purposefully classify elements as approximately matching even if the distance between them is significant. This choice assists style similarity evaluation when the input shapes do not share identical style elements. In the last column of Table 2, we provide comparisons against using the popular LightField shape descriptor alone (LFD) as a distance measure between shapes [Chen et al. 2003]. Our learning method has significantly higher average prediction accuracy compared to using LFD for style similarity. We also experimented with using a popular co-segmentation technique [Huang et al. 2011] in place of our element matching procedure. Instead of using our detected matching elements we used the set of matching parts returned by this co-segmentation technique, while keeping all other steps of our algorithm unchanged including the same distance measure formulation, input geometric features, and parameter learning. Even for the relatively simple class of coffee sets, the prediction accuracy measured on the same cross-validation sets dropped by 14 points to 73%, confirming that standard co-segmentation approaches are not well suited for evaluating style similarity. As explained in Section 4.4, our algorithm employs an iterative scheme that alternates between matching elements using our distance function and then updating its parameters. After the initial iteration, the prediction accuracy of our algorithm averaged over our seven datasets is 87.9%. At the second iteration, the accuracy increases to 89.0%, and at the third iteration the accuracy converges to 89.1% as reported in Table 2.

Elementary Distances. Figure 10 (left) shows the relative importance of each elementary distance as reflected by its learned weight in the element similarity term (normalized by the sum of all elementary distance weights), averaged over all our seven cat-

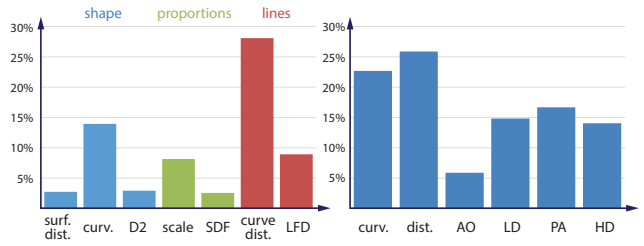

Figure 10: *(left) Similarity weights, features from left to right are: surface distance, curvature difference, shape distribution D2 difference, scale difference, shape diameter difference, curve distance, light field descriptor difference, (right) saliency weights, features from left to right are: curvature metrics, location metrics, ambient occlusion, and three levels of shape distinctness from [Shtrom et al. 2013].*

egories. We observe that the distances between feature curves are contributing the most to our style measure. Figure 10 (right) similarly shows the relative importance of each saliency feature as reflected by the absolute scale of its learned weight in the saliency model (normalized by the sum of all absolute saliency weights). We note that the relative importance for saliency features is less direct since we employ a non-linear sigmoid-based model for measuring saliency. Curvature- and location-based features appeared to have higher contribution. We include all the learned weights for each elementary distance and saliency feature for each dataset in the supplementary material.

Complexity and Runtimes. Our distance computation has two main time-consuming steps: computation of per sample point geometric features used for elementary distances and saliency metrics, and element matching. Feature computation takes 40 seconds on average for a pair of shapes. Element matching consists of patch sampling, transformation clustering and element extraction steps which take 115, 85 and 35 seconds respectively for a shape pair on average. In total, evaluating the distance function takes about 4.5 minutes. We note that several parts of our algorithm could be implemented much more efficiently e.g., patch segmentation is implemented on a single thread. Regarding computational complexity, the distance function evaluation has quadratic complexity in the number of patches in shapes and linear in the number of point samples. We note that the number of patches is relatively low, ranging from 20 to 80 at most.

Optimizing the objective function for learning the parameters of our distance function requires 30 seconds per 100 training queries. The complexity of the parameter learning stage is linear in the number of triplets. Learning requires evaluating distance functions for all shapes pairs in the training queries. For our largest dataset, the learning stage requires about 50 hours. We note that the learning stage is an offline procedure; once the measure is learned, applying it for a shape pair requires only a few minutes, as discussed above. All running times are reported on an Intel E5-2697 v2 processor.

7 Applications

We describe three novel applications of our learned style similarity measure. We first discuss how to use this measure to organize a shape collection to allow users to visually explore groups of shapes based on style. Second, we train classifiers that infer style-related tags, or labels, for shapes to facilitate style keyword based search. Third, we discuss how to use our measure to suggest stylistically compatible shapes to designers during scene composition.

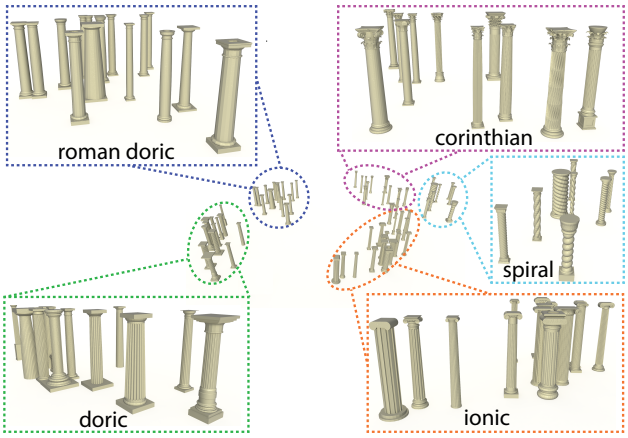

Figure 11: *Embedding the column dataset in a 2D space based on learned pairwise distances yields distinct Gaussian-like clusters that correspond to known architectural orders visualized by colored boxes.*

Organizing shape collections. Our similarity measure can be used for organizing shape collections based on style by using multi-dimensional space embedding. First, we compute the style distances between all shapes that belong to the triplets generated using the methodology described in Section 5. Then we construct a graph whose nodes represent shapes and whose edge connect shapes when their computed pairwise distance is available and is below 0.5. The edge lengths are set to be the pairwise distances between these shapes. We embed the graph in 2D using the Isomap technique [Tenenbaum et al. 2000], preserving the geodesic distances on the graph as much as possible. The 2D space can be directly visualized, Figure 11 shows the resulting embedding for columns. The shapes can be further clustered with unsupervised learning techniques to display stylistically similar groups of shapes to the users. In Figure 11, we use a Gaussian mixture model to perform clustering. Interestingly, the discovered groups are largely correlated to architectural orders commonly used by art historians to describe column styles.

Style-based shape tagging. Given a set of shapes with style labels provided by an expert, we can train a classifier that infers, or propagates, labels to the rest of the shapes in a collection. To build such a classifier, we first associate each shape with a feature vector. We employ the Isomap technique to place each shape in a high-dimensional space that can be used to reliably perform classification. The embedding coordinates per shape are concatenated into a feature vector, which is provided as input to the classifier for training and evaluation. For training, we labeled shapes in three collections, where style can be described with commonly agreed terms. For example for columns, we used the architectural orders, reported in Figure 11. We labeled buildings broadly according to their geographic-temporal style into 'Gothic', 'Byzantine', 'Russian', 'Baroque', and 'Asian'. For coffee sets, we used a coarse labeling of 'antique', 'modern', and 'art-deco'. We experimented with various classifiers, such as k-nearest neighbors, logistic regression, naive Bayes, Fisher's linear discriminant and Support Vector Machines. While training these classifiers, we performed hold-out validation on the training sets to choose the dimensionality D of the embedding for each collection: we start with $D = 1$, then we proceed with $D = 2, 3$, and so on. We stop once the hold-out validation error increases more than 10% with respect to the best previous value of D, or when $D = 20$. On average, the single nearest neighbor classifier was able to predict style labels with highest

accuracy. Through ten-fold cross-validation, the labeling accuracy in the test sets was 95.6% for columns, 86.6% for buildings, and 94.1% for coffee sets. We demonstrate labeling results visually in the accompanying video.

Style-based suggestions for scene modeling. Finally, our learned measure can be used to help designers during interactive scene composition by providing stylistic suggestions of shapes. The input to this application is a collection of shapes and a scene being modeled. The application compiles an ordered list of shapes from a collection according to their style distance to the shapes in the scene, or selected shapes of interest (query shapes) specified by the designer. One approach to compiling such an ordered list would be to compute the distances of the query shapes to all collection shapes using our learned measure; we could then order the shapes in the collection according to their distance to the query shapes. However, computing all such distances is computationally expensive. Thus, if the query shapes are not part of the collection, we propose a greedy procedure where, for each query shape, we compute its nearest shape neighbor in the collection. We do this by performing a nearest neighbor search in the space of features we use for element matching (see Appendix), but evaluated across the entire shape, sidestepping explicit element detection. In this manner, the shape that is most structurally and geometrically similar to the query shape is found first. To generate the ordered list, we use the geodesic distances from that shape to all other shapes in the database through a precomputed graph, which is constructed using the process described for organizing shape collections. We demonstrate the application of stylistic suggestions for furniture in the accompanying video.

8 Discussion

We have described the first algorithm for computing a structure-transcending style similarity measure between objects. As demonstrated, our measure is well aligned with human perception of style, owing to our novel use of parameter learning from crowdsourced style similarity queries. Since understanding style is fundamentally important for analysis of man-made objects, our method directly benefits a range of applications such as the ones described in the paper.

We see many exciting directions for future work. While we put significant effort into exploring geometric features and elementary distances relevant for visual motif and consequently style analysis, it remains an open question if the features discussed in our paper are sufficient to compare the style of shapes. In particular, for large structures, such as buildings, the overall arrangement of parts and elements is likely to play some role in style parsing. Instead of designing features and distances from scratch, it could be interesting to explore if these can be learned directly from raw shape data. Deep learning architectures could be used for this purpose, as well as for learning more advanced models of style similarity. Our work focuses on similarity within broad object categories, such as between pieces of furniture, or buildings, where stylistic commonalities are most obvious; it may be interesting to consider cross-category style evaluation between objects, e.g. evaluating style similarity between buildings and furniture. The first step for such a task would be to evaluate how consistent humans are at this task. In parallel to our work, another method was introduced to evaluate style similarity for furniture [Liu et al. 2015] based on co-segmented shapes. We speculate that combining feature-based joint segmentation [Huang et al. 2011] or template fitting methods [Kim et al. 2013] with our alignment-based element matching technique could further improve our style similarity measure for some classes. Lastly, evaluating the stylistic similarity between objects using our learned measure takes a considerable amount of time, which could be improved through a faster implementation and more efficient element matching techniques.

9 Acknowledgments

Kalogerakis gratefully acknowledges support from NSF (CHS-1422441). Sheffer gratefully acknowledges support from NSERC and GRAND NCE. We thank Nicholas Vining and Mikhail Bessmeltsev for their valuable help in editing the paper and the video. We thank the anonymous reviewers for their comments.

References

ANDREW, G., AND GAO, J. 2007. Scalable training of l1-regularized log-linear models. In *International Conference on Machine Learning*.

ASAFI, S., GOREN, A., AND COHEN-OR, D. 2013. Weak convex decomposition by lines-of-sight. In *Proc. SGP*.

AUCOUTURIER, J., AND PACHET, F. 2002. Music similarity measures: Whats the use. In *SMIR*.

BALLARD, D. H. 1987. Readings in computer vision: Issues, problems, principles, and paradigms. ch. Generalizing the Hough Transform to Detect Arbitrary Shapes.

BELL, R. M., AND KOREN, Y. 2007. Lessons from the netflix prize challenge. *SIGKDD Explor. Newsl. 9*, 2.

BESL, P. J., AND MCKAY, N. D. 1992. A method for registration of 3-d shapes. *IEEE Trans. Pattern Anal. Mach. Intell. 14*, 2.

BLUMENSON, J. J. G. 1995. *Identifying American Architecture: A Pictorial Guide to Styles and Terms, 1600-1945*.

BONNEEL, N., SUNKAVALLI, K., PARIS, S., AND PFISTER, H. 2013. Example-based video color grading. *ACM Trans. on Graph. 32*, 4.

BURGES, C., SHAKED, T., RENSHAW, E., LAZIER, A., DEEDS, M., HAMILTON, N., AND HULLENDER, G. 2005. Learning to rank using gradient descent. In *Proc. ICML*.

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.

CHEN, X., SAPAROV, A., PANG, B., AND FUNKHOUSER, T. 2012. Schelling points on 3d surface meshes. *ACM Trans. Graph. 31*, 4.

COMANICIU, D. 2003. An algorithm for data-driven bandwidth selection. *IEEE Trans. Pattern Anal. Mach. Intell. 25*, 2.

CONNECTED LINES, 2014. Period furniture style guide. $http://www.connectedlines.com/styleguide/index.htm$.

CURTIS, F. E., AND OVERTON, M. L. 2012. A Sequential Quadratic Programming Algorithm for Nonconvex, Nonsmooth Constrained Optimization. *SIAM Journal on Optimization 22*, 2.

DOERSCH, C., SINGH, S., GUPTA, A., SIVIC, J., AND EFROS, A. A. 2012. What makes Paris look like Paris? *ACM Trans. Graph. 31*, 4.

FU, H., COHEN-OR, D., DROR, G., AND SHEFFER, A. 2008. Upright orientation of man-made objects. *ACM Trans. Graph. 27*, 3.

GARCES, E., AGARWALA, A., GUTIERREZ, D., AND HERTZMANN, A. 2014. A similarity measure for illustration style. *ACM Trans. Graph. 33*, 4.

GREIG, D. M., PORTEOUS, B. T., AND SEHEULT, A. H. 1989. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society 51*, 2.

HERTZMANN, A., JACOBS, C. E., OLIVER, N., CURLESS, B., AND SALESIN, D. H. 2001. Image analogies. In *SIGGRAPH*.

HUANG, Q., KOLTUN, V., AND GUIBAS, L. 2011. Joint shape segmentation with linear programming. *ACM Trans. Graph. 30*, 6.

HUANG, Q.-X., SU, H., AND GUIBAS, L. 2013. Fine-grained semi-supervised labeling of large shape collections. *ACM Trans. Graph. 32*, 6.

HURTUT, T., GOUSSEAU, Y., CHERIET, F., AND SCHMITT, F. 2011. Artistic line-drawings retrieval based on the pictorial content. *J. Comput. Cult. Herit. 4*, 1.

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.

KALOGERAKIS, E., HERTZMANN, A., AND SINGH, K. 2010. Learning 3D mesh segmentation and labeling. *ACM Trans. Graphics 29*, 4.

KALOGERAKIS, E., CHAUDHURI, S., KOLLER, D., AND KOLTUN, V. 2012. A probabilistic model for component-based shape synthesis. *ACM Trans. Graph. 31*, 4.

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.

LEIFMAN, G. 2012. Surface regions of interest for viewpoint selection. In *CVPR*.

LEWIS, A. S., AND OVERTON, M. L. 2013. Nonsmooth optimization via quasi-newton methods. *Math. Program. 141*, 1-2.

LEWIS, M. 2008. *Architectura: elements of architectural style*. Barrons Educational Series.

LI, H., ZHANG, H., WANG, Y., CAO, J., SHAMIR, A., AND COHEN-OR, D. 2013. Curve style analysis in a set of shapes. *Computer Graphics Forum 32*, 6.

LIU, T., HERTZMANN, A., LI, W., AND FUNKHOUSER, T. 2015. Style compatibility for 3d furniture models. *ACM Trans. Graphics, to appear 34*, 4.

MA, C., HUANG, H., SHEFFER, A., KALOGERAKIS, E., AND WANG, R. 2014. Analogy-driven 3D style transfer. *Computer Graphics Forum 33*, 2.

MITRA, N. J., GUIBAS, L. J., AND PAULY, M. 2006. Partial and approximate symmetry detection for 3d geometry. *ACM Trans. Graph. 25*, 3.

NOCEDAL, J., AND WRIGHT, S. J. 2006. *Numerical Optimization*.

NUTTING, W. 1928. *Furniture Treasury*.

OSADA, R., FUNKHOUSER, T., CHAZELLE, B., AND DOBKIN, D. 2002. Shape distributions. *ACM Trans. Graph. 21*, 4.

SCHMIDT, M., FUNG, G., AND ROSALES, R. 2007. Fast optimization methods for l1 regularization: A comparative study and two new approaches. In *Proc. ECML*.

SHAPIRA, L., SHAMIR, A., AND COHEN-OR, D. 2008. Consistent mesh partitioning and skeletonisation using the shape diameter function. *The Visual Computer 24*, 4.

SHTROM, E., LEIFMAN, G., AND TAL, A. 2013. Saliency detection in large point sets. In *Proc. ICCV*.

TENENBAUM, J. B., AND FREEMAN, W. T. 2000. Separating style and content with bilinear models. *Neural Comput. 12*, 6.

TENENBAUM, J., SILVA, V., AND LANGFORD, J. 2000. A global geometric framework for nonlinear dimensionality reduction. *Science 290*, 5500.

TIBSHIRANI, R. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society 58*.

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. on Graphics 32*, 4.

VAN KAICK, O., FISH, N., KLEIMAN, Y., ASAFI, S., AND COHEN-OR, D. 2014. Shape segmentation by approximate convexity analysis. *ACM Trans. on Graph. 34*, 1.

WILLATS, J., AND DURAND, F. 2005. Defining pictorial style: lessons from linguistics and computer graphics. *Axiomathes 15*.

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*, 6.

YUMER, M., AND KARA, L. 2014. Co-constrained handles for deformation in shape collections. *ACM Trans. Graph. 32*, 6.

Appendix

Shape and part features

Elementary distance. We describe here the elementary distances we used for measuring geometric similarity between elements (Section 4.1). In total, we used 77 elementary distances. To compute them, first we uniformly sample the surfaces of the shapes with $20K$ points, so that the distances are invariant to mesh artifacts. Then we compute the elements' surface distance, by aligning them with ICP and including the average closest point-to-point distance and average distance between their normals as our 2 first elementary distances. Then we compute the curvature tensors for each point on the element surface and extract 13 feature values (min/max curvature by value, min/max curvature by magnitude, mean curvature, Gaussian curvature, the absolute value of the aforementioned six features, as well as the mean magnitude of the two principal curvatures). We compute histograms of those 13 curvature features with $16, 32, 64, 128$ bins for each element. We also compute histograms of the elements' shape diameter [Shapira et al. 2008] with $16, 32, 64, 128$ bins, and the D2 shape distribution histograms [Osada et al. 2002] with $16, 32, 64, 128$ bins. On each of those curvature, shape diameter and D2 histograms, we measure the Earth Mover's Distances (4×15 elementary distances). We then extract the following feature curves on the elements: boundaries, ridges and valleys lines. Using the same element alignment we got from ICP, we compute the average closest curve point-to-point and normal distances for each of the three types of feature curves separately and for the whole set of curves (4×2 elementary distances). We extract the silhouette of the aligned shapes under different viewpoints [Chen et al. 2003], compute their Zernike moments, Fourier coefficients, eccentricity, circularity and estimate their euclidean distance for each of them (4 elementary distances). Finally, we use the axis-aligned bounding box scales of the aligned shape features and we measure their absolute differences along three axes (3 elementary distances).

Collection	# Total users	# Reliable users	# Rejected users	# Male	# Female	# Unknown gender	# age 18-35	# age 36-50	# age >50	# Unknown age
building	583	522	61	276	303	4	375	137	68	3
furniture	662	610	52	323	336	3	427	170	63	2
lamp	659	601	58	333	322	4	423	162	71	3
column	439	383	56	200	236	3	291	98	49	1
coffee set	144	129	15	65	79	0	99	25	20	0
cutlery	108	95	13	56	51	1	68	26	14	0
dish	121	99	22	59	62	0	81	32	8	0
Total	2716	2439	277	1312	1389	15	1764	650	293	9
Total unique	1277	1198	175	605	666	6	812	308	156	1

Table 3: *Participant statistics.*

Category	# Total Queries	# (%) Q. (iii) plurality	# (%) Q. (iv) plurality	# (%) (i) & (ii) plurality	# (%) Q. majority	**% persistence** all/reliable	**% consistency** all/reliable	**% consistency** majority	**% consistency** (i) vs (ii)
furniture fine	1000	0 (0.0%)	21 (2.1%)	963 (96.3%)	953 (95.3%)	91.7% / 92.9%	89.5% / 90.6%	92.4%	98.3%
furniture coarse	200	0 (0.0%)	78 (39.0%)	111 (55.5%)	100 (50.0%)	81.5% / 83.0%	71.2% / 72.8%	81.3%	93.7%
furniture random	50	0 (0.0%)	35 (70.0%)	14 (28.0%)	12 (24.0%)	78.1% / 79.6%	74.2% / 76.2%	80.8%	94.4%
lamp fine	1000	1 (0.1%)	9 (0.9%)	984 (98.4%)	976 (97.6%)	94.7% / 95.6%	94.8% / 95.3%	96.3%	99.1%
lamp coarse	200	0 (0.0%)	56 (28.0%)	128 (64.0%)	115 (57.5%)	82.8% / 83.6%	69.6% / 71.5%	79.4%	92.8%
lamp random	50	0 (0.0%)	38 (76.0%)	9 (18.0%)	9 (18.0%)	87.2% / 87.7%	72.6% / 73.2%	72.2%	92.5%
coffee set fine	200	0 (0.0%)	7 (3.5%)	192 (96.0%)	188 (94.0%)	86.5% / 88.3%	88.6% / 90.1%	92.7%	96.2%
coffee set coarse	50	0 (0.0%)	16 (32.0%)	31 (62.0%)	27 (54.0%)	70.0% / 72.0%	68.9% / 70.6%	78.9%	89.3%
coffee set random	20	0 (0.0%)	9 (45.0%)	10 (50.0%)	9 (45.0%)	70.8% / 74.0%	68.6% / 70.0%	74.4%	91.2%
cutlery fine	200	3 (1.5%)	10 (5.0%)	184 (92.0%)	183 (91.5%)	88.1% / 89.8%	89.4% / 91.4%	93.7%	97.7%
other coarse	2000	3 (0.2%)	192 (9.6%)	1728 (86.4%)	1636 (81.8%)	79.7% / 82.5%	81.2% / 83.2%	89.1%	93.6%
all fine	2400	4 (0.2%)	47 (2.0%)	2323 (96.8%)	2300 (95.8%)	92.2% / 93.4%	91.6% / 92.6%	94.2%	98.4%
all coarse	2450	3 (0.1%)	342 (14.0%)	1998 (81.6%)	1878 (76.7%)	79.9% / 82.4%	79.2% / 81.2%	87.9%	93.5%
all random	120	0 (0.0%)	82 (68.3%)	33 (27.5%)	30 (25.0%)	80.6% / 82.1%	72.6% / 73.9%	76.3%	93.1%
Total	4970	7 (0.1%)	471 (9.5%)	4354 (87.6%)	4208 (84.7%)	85.7% / 87.7%	85.0% / 86.5%	91.3%	95.8%

Table 4: *Study and algorithm statistics based on query formation strategy. Other coarse includes buildings, dishes, and columns. The last column measures agreement when non-discriminative answers are excluded.*

Features for elementary saliency. We describe here the geometric features that were used in Section 4 for measuring element saliency (Section 4.3, see Equation 9) and the prevalence of matching elements (see Equation 11). In total, we gathered 20 geometric features in our element saliency measure. All geometric features are computed on the sample points of the elements' surface. First we used the height of the sample point and its horizontal distance to shape center. The metrics of height and horizontal distance are relative to the bounding box size of the shape (2 saliency features). We also compute the geodesic distance from each point to all other points and use the average geodesic distance as feature (1 saliency feature). We also compute the ambient occlusion for each point by shooting rays towards the hemisphere along its normal direction and counting the percentage of rays which do not intersect with the shape (1 saliency feature). Similarly to the curvature-related elementary distances, we include the absolute values of min/max curvature by value, the absolute values of min/max curvature by magnitude, the absolute value of the mean curvature and Gaussian curvature, as well as the mean magnitude of two principal curvatures (7 saliency features). Following the distinctness idea in [Shtrom et al. 2013], we compute histograms of various features and use the dissimilarity of the histograms between neighboring points as saliency features. Besides the original Simplified Point Feature Histogram whose bins count relative angular directions of the normals, we also compute spin images [Johnson and Hebert 1999] and 3D shape contexts histograms based on [Kalogerakis et al. 2010]. To measure distinctness among different range of contextual shape information, we use 3 levels of neighbor ranges (3×3 saliency features). Note that all of the saliency features above are calculated on points and the saliency of an element or a region is a sum of the point saliency which implicitly accounts for the area of the element or the region.

furniture (100 triplets)	MTurk	Expert	Casual
number of users	20	5	32
% consistency	88.2%	98.2%	94.9%
% consistency (i) vs (ii)	98.9%	99.5%	98.9%
building (100 triplets)	MTurk	Expert	Casual
number of users	20	5	32
% consistency	87.0%	92.1%	92.9%
% consistency (i) vs (ii)	97.5%	99.2%	99.4%
cutlery (50 triplets)	MTurk	Expert	Casual
number of users	15	5	32
% consistency	92.7%	95.7%	92.0%
% consistency (i) vs (ii)	99.6%	100.0%	100.0%

Table 5: *Pilot study statistics.*

Extra Study Statistics

Table 3 provides detailed participant statistics per data category, including age and gender. It also lists the numbers of reliable versus rejected respondents. In the last row, we show the total number of unique participants in our study; note that some participants completed questionnaires for more than one collection. Table 4 shows the distribution of the queries based on similarity bias. As shown roughly 48.3% of the queries were constructed using fine-grained similarity bias, 49.3% were constructed based on coarser temporal or geographic bias, and the remaining 2.4% were assembled at random. The statistics show the response distribution in each scenario. Lastly Table 5 summarizes the results of the pilot study per participant category.