Co-Locating Style-Defining Elements on 3D Shapes

RUIZHEN HU
Shenzhen University
WENCHAO LI
Shenzhen Institutes of Advanced Technology
OLIVER VAN KAICK
Carleton University
HUI HUANG
Shenzhen University and Shenzhen Institutes of Advanced Technology
MELINOS AVERKIOU
University of Cyprus
DANIEL COHEN-OR
Shenzhen University and Tel Aviv University
and
HAO ZHANG
Simon Fraser University

This work was supported in part by NSFC (Grants No. 61602311, No. 61522213, and No. 61528208), 973 Program (Grant No. 2015CB352501), Guangdong Science and Technology Program (Grants No. 2014TX01X033, No. 2015A030312015, and No. 2016A050503036), Shenzhen Innovation Program (Grant No. JCYJ20151015151249564), Natural Science Foundation of SZU (Grant No. 827-000196), and NSERC (Grants No. 611370 and No. 2015-05407).

Authors' addresses: R. Hu and W. Li, College of Computer Science & Software Engineering, Shenzhen University, 821 CSSE Building, 3688 Nanhai Ave., Shenzhen, Guangdong, 518060, P.R. China; emails: {ruizhen.hu, manchiu.lee.9}@gmail.com; O. van Kaick, School of Computer Science, Carleton University, 5302 Herzberg Building, 1125 Colonel By Drive, Ottawa, ON, K1S 5B6, Canada; email: Oliver.vanKaick@carleton.ca; H. Huang (corresponding author), College of Computer Science & Software Engineering, Shenzhen University, 839 CSSE Building, 3688 Nanhai Ave., Shenzhen, Guangdong, 518060, P.R. China; email: hhzhiyan@gmail.com; M. Averkiou, Dept. of Computer Science, University of Cyprus, 1 University Avenue, Building FFT01, room B107, Nicosia 1678, Cyprus; email: melinos.averkiou@gmail.com; D. Cohen-Or, School of Computer Science, Tel Aviv University, Schreiber Building, room 216, Tel Aviv 69978, Israel; email: dcor@tau.ac.il; H. Zhang, School of Computing Science, Simon Fraser University, 8027 TASC I, 8888 University Drive, Burnaby, BC, V5A 1S6, Canada; email: haoz@cs.sfu.ca.

DOI: http://dx.doi.org/10.1145/3092817

We introduce a method for *co-locating style-defining* elements over a set of 3D shapes. Our goal is to translate high-level style descriptions, such as "Ming" or "European" for furniture models, into *explicit* and *localized* regions over the geometric models that characterize each style. For each style, the set of style-defining elements is defined as the union of all the elements that are able to discriminate the style. Another property of the style-defining elements is that they are frequently occurring, reflecting shape characteristics that appear across multiple shapes of the same style. Given an input set of 3D shapes spanning multiple categories and styles, where the shapes are grouped according to their style labels, we perform a cross-category co-analysis of the shape set to learn and spatially locate a set of defining elements for each style. This is accomplished by first sampling a large number of candidate geometric elements and then iteratively applying feature selection to the candidates, to extract style-discriminating elements until no additional elements can be found. Thus, for each style label, we obtain sets of discriminative elements that together form the superset of defining elements for the style. We demonstrate that the co-location of style-defining elements allows us to solve problems such as style classification, and enables a variety of applications such as style-revealing view selection, style-aware sampling, and style-driven modeling for 3D shapes.

CCS Concepts: • **Computing methodologies → Shape analysis;**

Additional Key Words and Phrases: Style elements, style-aware sampling

ACM Reference Format:

Ruizhen Hu, Wenchao Li, Oliver van Kaick, Hui Huang, Melinos Averkiou, Daniel Cohen-Or, and Hao Zhang. 2017. Co-locating style-defining elements on 3D shapes. ACM Trans. Graph. 36, 3, Article 33 (June 2017), 15 pages.
DOI: http://dx.doi.org/10.1145/3092817

1. INTRODUCTION

In recent years, there has been an increasing interest in studying the style of shapes [14, 16, 17, 19, 33]. In the visual arts, style is defined as "a distinctive manner which permits the grouping of works into related categories" [6] or "any distinctive, and therefore

Fig. 1. These pieces of furniture have known coherent styles. Can we analyze their geometry and extract locatable style elements which define the different style groups?

recognizable, way in which an act is performed or an artifact made or ought to be performed and made" [9]. Although these definitions imply that notions of style provide a rationale for grouping objects or artifacts, the apparent challenge, as well as the intriguing aspect, of the style analysis problem is that characterizations of individual styles are inherently abstract, ambiguous, and subjective. Human notions of style are typically conceived at a high level and stated in vague and non-descriptive terms, for example, classic versus contemporary, realistic versus cartoonish, and it is often difficult to break them down to low-level, objective descriptors. However, styles of all kinds are constantly imitated, prompting the inevitable question of how styles can be concretely extracted and applied.

In this article, we study the problem that concerns the *what* and *where* of *style elements*, where the elements refer to local regions of a shape that define its style. Specifically, we aim to translate high-level and non-descriptive languages of object styles typically used by humans, for example, "Japanese" or "Ming" for furniture, into *explicit* and *localized* geometric elements or regions over the models that characterize the styles (see Figure 1). These elements would then explicitly reflect the "distinctive manners which permit the grouping of shapes into style categories" from Fernie's definition. In addition to enabling analysis, the spatially located concrete elements can be manipulated directly, for example, for style-driven modeling. In contrast, some of the most recent works on style analysis either take a wholistic view of shape styles, learning a style compatibility measure without separating individual style elements [16], or require a geometric match between portions of two shapes to yield a style similarity measure [17]. To date, works that deal directly with style-defining properties either explicitly specify the properties *a priori* [33] or rely on hand-crafted rules to define the stylistic elements [14].

We take as input a set of diverse shapes organized into different style labels, where the grouping is provided by human experts; see Figure 2. Since the collection of shape styles studied in our work (see Section 6) are closely tied to domain knowledge, expert-annotated inputs are expected to be more reliable with less data contamination compared to crowdsourced user groupings. Using such an input to co-locate style elements is also well-motivated since it supports *feature selection* for separation of shape styles.

Our goal is to find a set of *defining* elements for each style group. To distinguish one style from the others, it is typically sufficient to learn a set of *discriminative* elements, which corresponds to a *minimal* set of elements that tell one style apart from the others. However, to fully characterize a style and enable applications that manipulate the styles, we need a set of elements that capture a *more complete* characterization of the style. This more complete set is precisely what we define as a set of *defining elements*. For example, to differentiate between the Children and European furniture in Figure 2, it is sufficient to look at whether the shape parts are smooth or adorned with embellishments. However, to confer a shape with the Children style, we may need to modify the shape beyond simply removing the decorations from the parts, for example, we need to add round corners to the parts.

In contrast to previous works on saliency [12, 28], we note that style-defining elements are different from geometrically salient regions. Salient regions are typically unique or distinctive when compared to other regions of the same shape [28]. On the other hand, style-defining elements should be widespread across shapes of the same style and form a type of collective property. For example, as seen in Figure 2, smooth and round patches appear all over the Children furniture, but rarely appear in other styles such as Ming and Japanese. Therefore, these elements are distinctive features for a particular (Children) style in the context of other styles but not necessarily for individual shapes. In addition, in our results in Figure 21, we show how view-point selection based on saliency highlights different shape aspects than when using style-defining elements.

Thus, given the goal of extracting style-defining elements, we pose our search as a feature selection problem [32], where the selected "features" would lead us to the elements we seek. Note that we reserve the term "element" to refer to concrete, locatable regions over a 3D shape. "Features," visually explicit or latent, serve to *describe* the elements themselves. In our method, we first sample a set of candidate elements from all the input shapes and describe each element with multiple features. Specifically, the elements are geometric patches extracted from the surfaces of the input shapes. Our goal is then to select the candidate elements that, when present on a shape, are able to define the style of the shape and distinguish it from other styles. For this step, we introduce a novel iterative method for finding a set of style-defining elements based on feature selection applied to the candidate elements. Specifically, the feature selection allows us to find sets of discriminative elements that distinguish among different styles. The union of all the sets of discriminative elements then provides a *superset* of elements that constitute the style-defining elements. Although there is redundancy in this set if our only goal is style classification, the redundancy provides a more complete characterization of a style by capturing the possible elements that can be present on a shape to define the style, which can be of use to certain applications beyond classification.

The explicit description and location of the style-defining elements can benefit several applications for content creation, since it may be difficult for an artist or automatic algorithm to create a shape in a given style just from an abstract high-level description. The defining elements are thus a concrete description of how to grant a specific style to an object. Also, given the nature of the elements we select, we do not require a correspondence between source and target shapes, in contrast to style analysis methods that transfer styles between 3D shapes by analogy [19], that match portions of shapes [17], or that require a consistent segmentation of shapes across the input set [16].

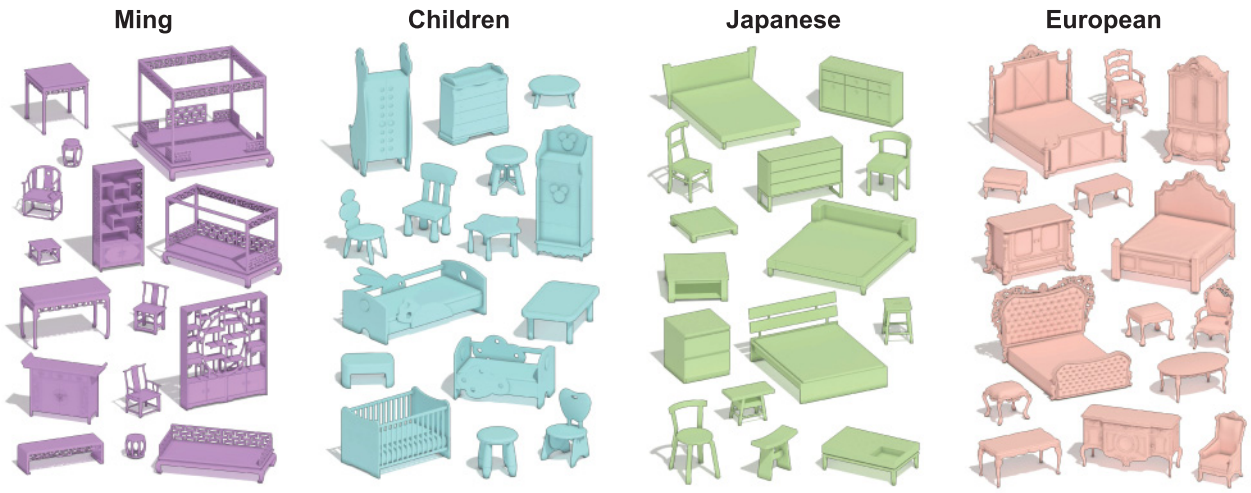

Fig. 2. A sample of a set of shapes given as input to our style co-analysis. The style groups have the labels on top and are rendered in different colors. Note that our method is applicable to a wide variety of shape categories and styles beyond furniture, as shown in Section 6.

To demonstrate the advantages of our method, we present several applications that are made possible by one of the key abilities of our approach: the spatial localization of elements. The applications include style-revealing view selection, style-aware sampling, and style-driven modeling. In addition, we show results on using the style-defining elements to analyze and classify the styles of various collections of man-made objects, and present comparisons of our method to alternative approaches that could be used to address the selection of elements, concluding that our method is more effective than these alternatives.

2. RELATED WORK

In this section, we cover works related to the analysis and comparison of image or shape styles, following with a discussion on feature selection, which is the main building block in our method.

Style and content analysis on images. There have been many works on style analysis for images, where style can be loosely seen as a set of characteristics that allow a meaningful grouping of images. Here, we discuss the works that are most relevant to our method.

Doersch et al. [5] find image patches that are characteristic of a specific geospatial location. The patches are extracted from a large dataset of images and can be seen as the elements that define the style of a city like Paris or some other specific location. Rather than using clustering to find representative patches, which may be influenced negatively by the large amount of irrelevant patches, they address the problem by sampling patches that are discriminative of a location. In their problem setting, an individual patch can solely serve as a representative of a location. Therefore, there is no need to perform any feature selection or build sets of style-defining elements, which is the goal in our work.

Instead of searching for *static* image patches that define a location, Lee et al. [13] focus on *dynamic* visual elements and discover image patches that gradually change in time or location. After detecting visual elements that are style-sensitive, their method establishes correspondences between elements in the dataset and models their range of variation across time or space. Arietta et al. [1] generalize the idea of a style label to any non-visual attribute, such as housing prices in a city. Similar to Lee et al., the method first detects visual elements that discriminate an attribute but then trains a predictor that relates the visual elements to the attributes.

Several methods for summarizing image content have also been proposed in the literature, such as seam carving [2] and bidirectional similarity [29]. Instead of naively resizing or cropping an image, the goal of these methods is to preserve the distinctive visual content of the image. Our approach also follows this general principle of detecting distinctive visual elements of 3D models, although the elements we detect are not generic, but related to the style labels of the shapes.

Style transfer by analogy. Earlier works on style transfer focused on curves and drew inspiration from the seminal work by Hertzmann et al. [10] on curve analogies, where a curve is synthesized by analogy to an exemplar. This method follows an approach similar to texture synthesis, where portions of an exemplar curve are randomly transferred to a base curve to confer the base with the style of the exemplar. Rather than using a single exemplar, Freeman et al. [7] combine strokes from a training set of hundreds of lines in the same style to synthesize a curve with a consistent style. Recently, Lang and Alexa [11] synthesize a curve from an exemplar by using a hidden Markov model that captures the distributions of features along the exemplar curve.

Berger et al. [3] learn different styles and levels of abstraction of face sketches drawn by different artists. Their approach analyzes the characteristics of both strokes and the structure of the faces, and learns several stroke properties that define each artist's style. Majerowicz et al. [20] generate arrangements of objects placed on horizontal support-surfaces, such as shelves and cabinets, from a single exemplar that can be a photograph or 3D scene. Their method also considers different features and their locations. However, the features are replicated from a single exemplar.

Ma et al. [19] apply the analogy approach to 3D shapes. Rather than simply transferring the style of an exemplar to a base shape, they take as input a source shape similar to the exemplar and a target shape and synthesize a new shape that follows the structure of the target but possesses the style of the exemplar. Their approach requires a correspondence between the source and base shapes and an analogy relationship between the source and target. The latest work by Lun et al. [18] takes the above analogy idea further by enforcing functionality preservation of the target shape. The style of the

exemplar is transferred to a target with a sequence of element-level operations. The method learns a cross-structural element compatibility metric and only applies operations on the target shape, which do not affect the functionality of the target.

It is interesting to observe that *automated* style transfer between shapes can be thought of as a way for style identification. Namely, whatever shape elements or properties that are determined to be transferable are style-defining. In this light, style identification by transfer and analogy is, in essence, a shape *differentiation* over the exemplar pair. In contrast to these approaches, our style extraction does not require a correspondence between a base and exemplar shape, nor a mapping between portions of the shapes as in the curve analogy works. In addition, we learn style-defining elements from a set of examples, in a *co-analysis* framework, which allows us to obtain a more general model for a style.

Style co-analysis. Rather than working with individual shapes or a training set, style co-analysis can work with a style-content table of a set of shapes to extract features that characterize a style. Xu et al. [33] pre-determine the style-defining features, namely, part proportions, and perform a *forward* co-analysis to group the feature variations that define different styles. In comparison, we formulate our problem as an *inverse* analysis, since we select style-defining elements from a large pool of candidates and are able to locate them on specific regions of the shapes. Our input is a style grouping and not a more granular style-content table. Li et al. [14] restrict their analysis to curve styles of 2D shapes that are decorative in nature and group stylistic curve features based on a set of hand-crafted rules. In contrast, our style analysis is data-driven and more general, since we define a varied set of candidate elements, and then extract the relevant style-defining elements from this set via feature selection.

Learning style similarities. An alternate line of works proposes to define global style similarity measures, where multiple features are considered when quantifying the similarity between the styles of two models, but without explicitly selecting sets of style-defining elements. Most notably, Lun et al. [17] define a structure-transcending similarity measure to compare the style of two shapes. The method searches for pairs of matching salient geometric elements across the shapes, and determines the amount of similarity based on the number of matching elements. The measure is tuned with crowdsourced training data that captures examples of style similarities. Liu et al. [16] also make use of crowdsourced relative assessments of style compatibility, but focus on the domain of 3D furniture models. Their metric for stylistic compatibility is based on obtaining a consistent segmentation of the input shapes and quantifying the similarity of their styles with part-aware feature vectors. The key distinction between our method and these works is that we explicitly identify the style-defining elements over the input shapes. Another technical difference lies in the input specifications: our method takes style grouping information from expert annotations while their methods were built on large sets of data triplets with style ranking information collected via crowdsourcing.

Furthermore, the style of other types of geometric datasets can also be compared with similar approaches. Garces et al. [8] propose a style similarity measure for clip art illustrations that is learned from crowdsourced similarity estimates, while O'Donovan et al. [22] propose a learning approach to obtain a perceptual similarity measure of fonts. In contrast to these works, our goal is to find a complete set of style-defining elements and locate them on the shapes, rather than defining only a global similarity.

Shape comparison and retrieval. In our method, we do not require correspondences between shapes to extract collection-wide style elements. To achieve that, we encode shapes as bag-of-words representations. There have been many works in shape retrieval on how to encode shapes efficiently for comparison [30], and also different bag-of-words representations have been proposed [15, 31]. In our method, we encode shapes with a word-frequency representation. We compare this choice to an alternative encoding in Section 6, to show that our representation is adequate for our setting.

Distinctive regions of 3D surfaces. Earlier work by Shilane et al. [26] shares some resemblance to our method. That work also attempts to extract *distinctive* regions over 3D shapes when given a multi-category shape collection. The key conceptual difference between the two works is that we seek *style*-defining elements, while their work seeks *content*-discriminating regions. They define the distinction of a surface region over a 3D object by how useful the region is in distinguishing the object from other objects belonging to different categories, in the context of shape retrieval. Specifically, the distinction for a region is high if, in a shape-based search of a database with the region as the query, the resulting ranked retrieval list would consist mostly of objects of the same category near the front. Clearly, our interpretation of style-defining elements and the method of element extraction are both different. We co-analyze a shape collection via classification tasks, while their method performs shape retrieval based on pairwise similarities. On the technical front, their retrieval is based on a single shape descriptor (the spherical harmonic descriptor), while our distance metric is learned over a set of features for each candidate element. We believe that style-discriminating and style-defining shape elements tend to be more local and more subtle, compared to content-discriminating surface regions, so the ensuing analysis calls for a feature representation and selection scheme that is more involved.

Feature selection. In statistical pattern recognition, there is a vast literature on feature selection. In the context of classification, the goal of these methods is to examine a large set of features to find a subset that is sufficient for discriminating a given class from others, eliminating redundancy and irrelevant features in the process. Feature selection methods can be grouped into three broad categories: filter, wrapper, and embedded methods [32]. Filter methods select features according to their statistical properties, such as correlation. Wrapper approaches test the performance of subsets of features by training a classifier with these features and evaluating their classification accuracy. Thus, wrapper methods tend to be more demanding than filter approaches due to the repeated training and use of a classifier. Embedded methods perform feature selection while training a classifier and, thus, offer a better balance between performance and accuracy.

In our work, we use a filter method, the minimal-redundancy-maximal-relevance (MRMR) criterion [23], to efficiently select a set of candidate elements. MRMR incorporates several criteria that have been shown to lead to a quality feature selection. We also refine the candidate set with a wrapper method, specifically, a standard *forward sequential feature selection* method [32]. We provide more details on these methods in Section 5. In Section 6, we also compare to an embedded method, L1-regularized logistic regression [25], as an alternative to our feature selection.

3. OVERVIEW

The input to our analysis is a set of shapes grouped into different styles (Figure 2), where the shapes do not have to belong to the same category. The goal of our method is to extract defining elements for each style and co-locate them across the set.

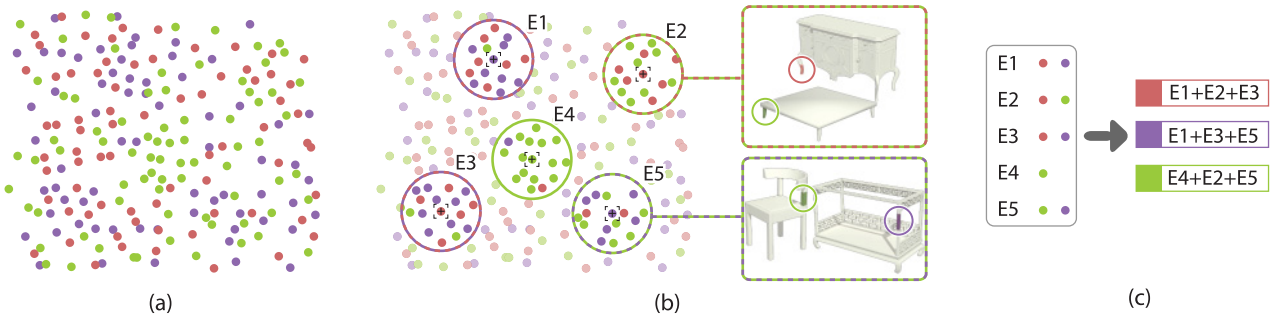

Fig. 3. Overview of our method for the discovery of style-defining elements. (a) We collect a set of initial elements from regions of the shapes, shown here as points in a 2D embedding. The distances between points reflect their similarity in terms of the features that describe them. Each element has a style label (point color). No clear clusters are present. (b) We sample candidate elements (indicated by the crosses at the circle centers) with an analysis of density, and learn a similarity measure for each element based on its nearest neighbors (points inside the circles). The insets show the geometric patches corresponding to the elements in the embedding. (c) We combine the candidate elements to discover sets of style-defining elements, for example, E1 + E2 + E3 define the "red" style.

One possible way of carrying out the co-location would be to first compute a correspondence between all of the input shapes, since a correspondence would indicate that two localized elements on corresponding regions of two different shapes should be recognized as being the same defining element. However, establishing correspondences for shapes in a varied set is quite challenging, especially since the input contains shapes from different categories. Instead, we follow an indirect approach and avoid the use of correspondences by working in a feature space. Specifically, we describe a shape as a "bag of words" from a dictionary of geometric elements extracted from all input shapes. By learning what are the relevant "words" that classify the shapes into the given style groups, we are then able to obtain the sets of style-defining elements. By construction, the "words" of the dictionary are localized. Therefore, the defining elements are associated with specific regions of the shapes.

As motivated in the Introduction, the defining elements should be widespread across shapes of a style and be able to provide a complete characterization of the style. Thus, our analysis consists of two major steps, illustrated in Figure 3. First, our method extracts a large set of initial elements from the input shapes. Specifically, we consider geometric patches extracted from the surfaces of the shapes, where each patch is described by a set of features, for example, a histogram of the mean curvature on the patch. We then sample elements from the initial set to create a set of candidate elements, where the candidates occur frequently within at least one style set. We call them *candidates*, since not all of them are selected as defining elements. The initial sampling and candidate selection is explained in more detail in Section 4. In the second step of the method, we perform a feature selection on the candidates to obtain sets of elements that are able to discriminate among individual styles. The elements are then grouped into larger sets of *style-defining elements*, implying that they form a complete set of elements that not only discriminate but fully describe the style. We give more details about this step in Section 5.

4. CANDIDATE ELEMENTS

In this section, we describe the first step of our method, where we create a large set of initial elements and then sample from these elements to build the set of candidates.

Extraction of initial elements. We first create a large set of initial elements $\mathcal{I}$ from all of the input shapes, where the elements are geometric patches extracted from the surfaces of the shapes. To handle non-manifold shapes and polygon soups, we represent each shape with a set of points. We sample $N_s = 20{,}000$ points on each shape's surface using Poisson disk sampling. We then uniformly sample $N_p = 200$ points from the input point cloud, and use the sampled points as patch centers. We grow each patch from its center while its geodesic radius is below a threshold τ of the shape's bounding box diagonal. We explore different values for τ in the discussion. The geodesic neighborhood of a point is based on the graph of k-nearest neighbors (KNN) of all points, where $k = 6$.

Each element is then represented by a set of geometric features. We first compute a series of point-level features for each patch. We use features similar to those proposed by Lun et al. [17], including the curvature of the points, saliency of point sets, ambient occlusion, average geodesic distance, point height and upright angle, shape diameter function (SDF), and PCA-based features. For each point-level feature, we then collect all its values appearing in a patch and create a histogram. We use from 16 to 64 bins for the histograms, depending on the features. We enrich the patch-level features with the point-feature histogram and D2 shape distribution of the patch. Note that, for the computation of these features, we assume that the input shapes are upright-oriented.

Sampling of candidate elements. We sample a set of candidate elements $\mathcal{C}$ from the set of initial elements $\mathcal{I}$, where the elements $\mathcal{C}$ are relevant to style analysis. In our problem, a substantial number of elements are similar to each other, and not all elements are relevant to characterize the different styles. This happens because our data is composed of geometric patches that are less distinctive than, for example, image patches. Thus, to avoid working with a large amount of similar, irrelevant elements, we sample a set of candidate elements with a density analysis applied to the intrinsic space of element similarities. We compute the density of all the elements with the same style label, and only keep elements that are density peaks. The density peaks correspond to the cluster centers of initial elements within a style set. By selecting only the density peaks, which are surrounded by a high concentration of similar elements, we avoid selecting additional elements that are similar to each other. At the same time, since the elements are density peaks, they occur frequently within the style.

To compute the density in a robust manner, we use the clustering method of Rodriguez and Laio [24]. The assumption of this method

Fig. 4. Elements selected on each shape that correspond to density peaks. Note how the patches are representative of different stylistic features, such as curved extremities and corners.

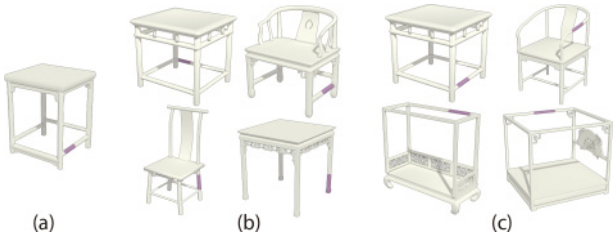

Fig. 5. Refinement of the similarity measure for the element shown in (a). (b) The initial measure finds similar elements in other shapes by giving equal weight to all features. (c) The refined measure considers only the features relevant to the style. We see how the similar elements are composed only of horizontal tubular regions after the refinement, which can have different heights instead of all being near the bottom of the shapes.

is that cluster centers are surrounded by elements with lower local density and are at a relatively large distance from elements with higher density, which enables the detection of several types of cluster configurations. In detail, the distance d_{ij} between two elements is computed by comparing the features of elements with the normalized correlation. The local density ρ_i of an element e_i is then defined as

$$\rho_i = \sum_j \chi(d_{ij} - d_c), \tag{1}$$

where $\chi(x) = 1$ if $x < 0$, and 0 otherwise. The parameter d_c is a cutoff distance; we explain its selection below. We then define the distance δ_i of an element to elements of higher density as

$$\delta_i = \min_{j:\rho_j > \rho_i} d_{ij}, \tag{2}$$

with the distance for the element of highest density being a special case defined as $\delta_i = \max_j d_{ij}$. We then take the top K elements of a shape that have the largest values for δ_i, where K is determined by using the variation of $\gamma_i = \rho_i \delta_i$ as suggested by Rodriguez and Laio [24]. Note that, since Equation (2) considers the relative densities of elements, the clustering is robust in relation to the choice of d_c. We set d_c as the percentile corresponding to 2% of the pairwise distances. In this way, we obtain the set of elements $\mathcal{C}$. Figure 4 shows examples of the density peaks selected with this method.

Refinement of element distance. The normalized correlation that compares two patches equally considers all their features, and does not give more weight to the features that are truly relevant to a given style. Therefore, to enable a more style-sensitive comparison of elements, we learn a specialized distance measure for each candidate element, using a discriminative learning algorithm. An example of the effect of this refinement through learning is shown in Figure 5. The specialized distance measures are then used in the steps of our method described in the next section.

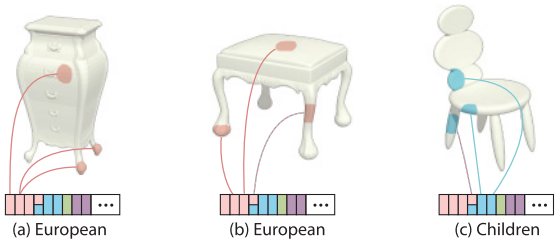

Fig. 6. Illustration of the term vectors of three shapes with two different styles, where elements of the same style are shown with the same color. In all of the vectors, the entries that are in the same position correspond to a common element. For example, the shape in (a) has two instances of the element corresponding to the second entry, while (b) has one such instance. The shape in (c), whose style differs from that of (a) and (b), has instances of different entries, although the fourth element is shared by the two styles.

In detail, we train a linear SVM detector for each candidate element, and use the nearest neighbors of the element as examples for learning. We first detect the style labels that dominate the neighborhood of a candidate element. We compute the histogram of labels that appear in the neighborhood, and initialize the set of dominant styles with the style that has the highest bin count in the histogram. Any labels with a bin count higher than half of the maximum are then also added to the set of dominant styles. We take the elements in the neighborhood that are labeled with the dominant styles as positive examples for learning, and elements with other labels as negative examples [5]. After learning the similarity measure of the element, we refine its neighborhood by finding its nearest neighbors with the learned measure. By iterating this procedure, we obtain a specialized distance measure for each element that better accounts for stylistic similarities (Figure 5). Note that, once the dominant styles of an element are determined, they are kept fixed throughout the iterations.

For the SVM detector, we learn a weight vector $\mathbf{w}_i$ for each candidate element $c_i \in \mathcal{C}$. The similarity of an element e_j to a candidate element c_i is then given by

$$S_{c_i}(e_j) = \mathbf{w}_i^T \mathbf{x}_j, \tag{3}$$

where $\mathbf{x}_j$ is the feature vector of e_j. We learn the weights with the convex optimization described in the work of Shrivastava et al. [27]. It has been shown that the performance of the detector is not compromised, even if some of the training samples are false negatives, attesting for the robustness of using this procedure.

5. STYLE-DEFINING ELEMENTS

In the second step of our method, we combine the elements $\mathcal{C}$ into sets of discriminative elements $\mathcal{E}_i$, and use them to create the sets of defining elements $\mathcal{D}_j$ of each style j. We solve these two tasks with a novel iterative method composed of an inner loop that performs feature selection and an outer loop that builds the sets of defining elements by repeatedly invoking the inner loop.

Shape representation. Following the concept of bag-of-words, we represent each shape as a vector $\mathbf{t}$ capturing the frequency of the elements $\mathcal{C}$ in the shape, which we call the *term vector* of the shape. This is illustrated in Figure 6. Thus, $\mathbf{t}$ is an m-dimensional vector, where m is the number of elements in $\mathcal{C}$. Each entry of $\mathbf{t}$ is an integer number that counts how many times the corresponding candidate element appears in the shape, normalized by the total number of elements in the shape. An entry can have the value zero, if the corresponding element does not appear in the shape.

To create the term vector for a shape S, we first extract elements from S with the same procedure that we use to obtain the initial elements for the training shapes. We then keep only the elements that correspond to the ones in $\mathcal{C}$ and create a term vector for the shape by setting the entries corresponding to the detected elements. To find what elements of S correspond to elements in $\mathcal{C}$, we use the SVM-based measure to compute the stylistic similarity of two patches. We recall that the measure was trained for each individual element during the candidate sampling. Thus, given an element $e \in S$, for every candidate $c \in \mathcal{C}$, we compute the similarity $S_c(e)$ with Equation (3). We then say that we detected e as an instance of c, if the similarity is larger than a threshold τ_p. We set $\tau_p = -1$, since this is the distance from the hyperplane that performs the regression in the SVM classifier to the closest negative training sample.

Discriminative elements. The inner loop of our method selects the style-discriminating elements with a combination of two feature selection algorithms: a filtering method followed by a wrapper method. Traditional wrapper feature selection algorithms sample many combinations of elements and test their discriminative power with the use of a classifier. They are effective in selecting discriminative features. However, the repeated use of a classifier for the consideration of many feature combinations makes this an inefficient process. Thus, we first select a candidate set of elements $\mathcal{F} \subset \mathcal{C}$ that are potentially discriminative with the use of a more efficient filtering technique. We perform the feature selection for each style separately, where each shape or term vector has a binary label.

To filter out most of the non-discriminative elements, we employ the minimal-redundancy-maximal-relevance (MRMR) criterion [23], and keep only 20 elements after the filtering. The idea behind this filtering approach is that elements that are discriminative are statistically dependent on the labels. This property can be quantified by searching for the maximal relevance in terms of the mutual information between term vector entries and labels. Moreover, since the combination of elements that work well individually does not necessarily lead to good classification performance, we also search for vector entries with minimal redundancy among themselves. Note that, the fact that sometimes the combination of features that work well individually decreases their classification performance, is a behavior that has been observed in feature selection research and linked to the redundancy of the features.

We then apply the *forward sequential feature selection algorithm*, a standard wrapper feature selection method [32], to select combinations of elements from $\mathcal{F}$ that are discriminative. This approach tests the discriminative power of elements with a classifier wrapped in a cross-validation scheme, and returns sets of discriminative elements $\mathcal{E}_i$, as we describe below. We use a k-nearest neighbor (KNN) classifier in our work. Note that the feature selection is now more efficient, as it is only applied to the candidate elements obtained with MRMR.

In more detail, suppose that we have a set of already selected elements $\mathcal{E}_i$, and the set of remaining elements $R = \mathcal{F} - \mathcal{E}_i$. Then, for each element $r_j \in \mathcal{R}$, we test the discriminative performance of $\mathcal{E}_i \cup \{r_j\}$ according to the KNN classifier. We average the results for 10 folds of cross-validation, that is, we divide the training set into 9 sets used for training and then test the performance on the remaining set, averaging the results for the 10 test sets. We add the r_j with the highest average performance to $\mathcal{E}_i$. The selection starts with an empty $\mathcal{E}_i$ and concludes when no elements can be added to improve the classification performance. Note that, in our problem setting, adding an element to the set implies that we consider its entry in the term vectors of the shapes.

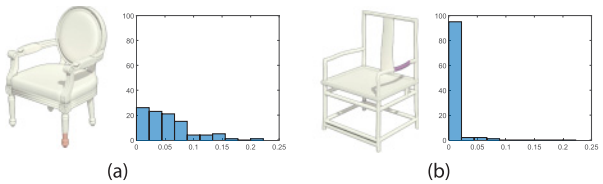

Fig. 7. Example of a positive and negative discriminative element selected for the European furniture style. For each element, we show the histogram of the term vector entries corresponding to this element, for all the shapes in the style. (a) According to the histogram, this is a positive element, since it appears in several shapes of this style. (b) This is a negative element, since it seldom appears in the shapes of the style (entries with value 0 are the most common).

One interesting aspect of the feature selection is that it selects both *positive* and *negative* discriminative elements. That is, we learn that certain elements need to be either *present* or *absent* from a shape to characterize a given style. To determine whether an element is positive or negative, we verify the presence of this element in all the shapes within the style discriminated by the element. This is verified by computing a histogram of the entry values in the term vector corresponding to the element, where we consider only the term vectors of shapes in the style. Based on this histogram, we can then determine the type of element, for example, if the element does not appear often in shapes of the style, meaning that the bin corresponding to value 0 is high, then it is a negative discriminative element. Figure 7 shows an example of positive and negative elements.

Defining elements. In the outer loop, we combine the selected discriminative elements, obtained by the repeated application of the inner loop, into a larger collection of style-defining elements. We start with an empty set of defining elements, and at each iteration of the inner loop, we augment this set with the discriminative elements discovered by the feature selection. We then remove the discovered elements from the working set, to select additional discriminative elements in the next iteration. We stop adding more elements when the classification performance falls below a threshold of 0.9. In this manner, at the end of the outer loop, we obtain a set that fully characterizes a style. An example of the construction of the sets of defining elements is shown in Figure 8.

Output. Given input shapes with n styles, the result of the method is a set of defining elements $\mathcal{D} = \{\mathcal{D}_1, \ldots, \mathcal{D}_n\}$, one for each style. We define the $\mathcal{D}_j$ for a style j as the union of all the sets of discriminative elements selected for this style, that is, $\mathcal{D}_j = \mathcal{E}_1^j \cup \ldots \cup \mathcal{E}_{m_j}^j$, where m_j is the number of sets of discriminative elements selected for style j. We can interpret a set $\mathcal{D}_j$ as the elements that, when present or absent of a shape, characterize this shape as belonging to style j. The sets $\mathcal{D}_j$ can then be used for classification or for applications such as style-aware modeling.

6. RESULTS AND EVALUATION

We evaluate our method for the discovery of style-defining elements in this section and show example applications in the next section.

Datasets. In our experiments, we use five sets of shapes organized by experts into different styles. The sets are: (1) *Furniture*, with 618 models, four styles (Children, European, Japanese, and Ming), and five types of content (beds, cabinets, chairs, stools, and tables); (2) *Furniture legs*, with 84 models and three styles (Cabriole,

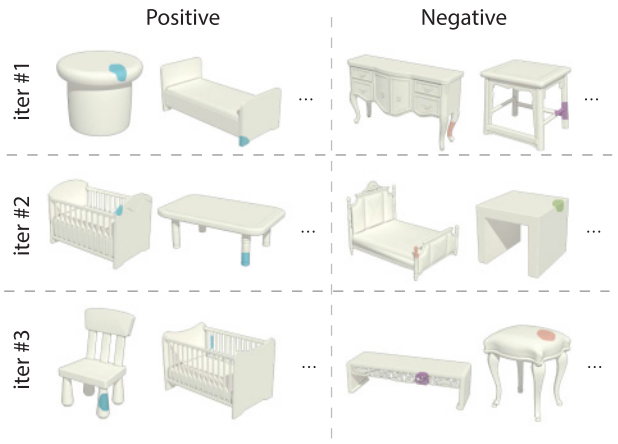

Fig. 8. Discovery of defining elements. Each row shows the discriminative elements selected at an iteration of the method, for the Children style. For each iteration, we show two positive elements on the left, and two negative elements on the right. All the elements together constitute the set of defining elements of the style.

Tapered, and Fluted); (3) *Buildings*, with 89 models and five styles (Asian, Baroque, Byzantine, Gothic, and Russian); (4) *Cars*, with 85 models and four styles (F1, Pickup, Sports, and Truck); and, (5) *Drinking vessels*, with 84 models, three styles (vessels suitable for Chinese liquor, sake, and wine), and two content types (bottles and cups). The sets are composed of shapes that we collected, with the exception of the buildings that were made available by Lun et al. [17]. Our dataset includes a similar variety of categories as previous works, and our furniture set is more than double the size of the largest set used in the recent works of Liu et al. [16] and Lun et al. [17].

Statistics. We extract 200 initial elements from each shape. As an example, for the furniture dataset, this sums up to a set of approximately 120K initial elements. After applying our method, we obtain from 5 to 40 defining elements for each style, depending on the size and variability of the input set. Running our method takes 17min on the large set of furniture models, and 2min for the small set of furniture legs, with an unoptimized MATLAB implementation running on an iMac with a 4GHz Intel Core i7 processor and 32GB of DDR3 memory. The candidate selection takes approximately 60% of the execution time, with the feature selection taking 40% of the time. After processing a set, detecting the style-defining elements on a query shape takes less than 1ms.

Analysis of elements. We first analyze different aspects of the style-defining elements, and provide visual samples of the results to provide an overview of the characteristics of the elements. We analyze mainly positive style-defining elements, since these can be displayed on the shapes. The negative elements take part in the application of style classification and style-driven modeling, discussed below.

To provide some insight on the general characteristics of the positive style-defining elements, we show in Figure 9 the full set of defining elements selected for two styles. For each style, we show a subset of three shapes that together display all the elements selected for the style. In addition, we show all the defining elements that exist on each individual shape. We see that the elements do not exhaustively cover the surface of the shapes but rather appear on key regions that define the styles, such as tubular structures and relief ornaments in the Ming style. We note that, since our elements

Fig. 9. Subsets of shapes covering the full set of defining elements selected for each style. For each shape, we also show the full set of defining elements that appear on the shape.

Fig. 10. A single style-defining element appearing on multiple shapes of the Ming style. The element captures shape regions rich in relief ornaments.

are patches of fixed size, they do not capture these key regions in their entirety but are located arbitrarily over these regions.

We also remark that the elements are not restricted to each single shape but are in correspondence across many shapes of the style, forming a collective characteristic of the style. To illustrate this point, Figure 10 shows an example of the same style-defining element appearing across multiple shapes of the Ming style. Therefore, each element in Figure 9 may correspond to more than one defining element, since it may be in correspondence with different defining elements on other shapes.

To analyze the occurrence of elements across multiple shapes in a quantitative manner, we analyze their *frequently-occurring* property for each category of shapes. For each shape, we compute the percentage of its defining elements that belong to each style in the set. We then display in a matrix cell (i, j) the average of the percentages of elements of style i that appear on all shapes of style j. The diagonals of the matrices in Figure 12 demonstrate that in a shape of style i, around 20% to 40% of its patches are on average defining elements of style i, while each other style accounts for at most 10% of elements. Thus, the matrices confirm that the defining elements of a shape are mostly restricted to the shape's style.

Another important property of the elements selected by our method is that they capture *distinctive characteristics* of the styles. This is demonstrated in Figure 11, which shows examples of style-defining elements selected for shapes of all styles in our datasets. Note how these elements capture important characteristics that define the styles, such as sharp edges for Japanese furniture, tubular structures for Ming furniture, relief ornaments for European furniture and furniture legs, defining parts such as seats and wheels for cars, and specific substructures for the roofs of buildings.

Quantitative evaluation. To perform a quantitative evaluation of our method, the most direct approach is to collect a ground-truth of defining elements from human subjects, and compare our results to such a user-provided data. However, manually creating a ground-truth is a highly complex problem due to the large number of shapes that need to be analyzed simultaneously. For example, our method finds on average 20 elements defining a style. It would be difficult for users to search for such large sets across multiple shapes and styles, even if the initial elements are already provided to the users. This is because the defining elements need to be

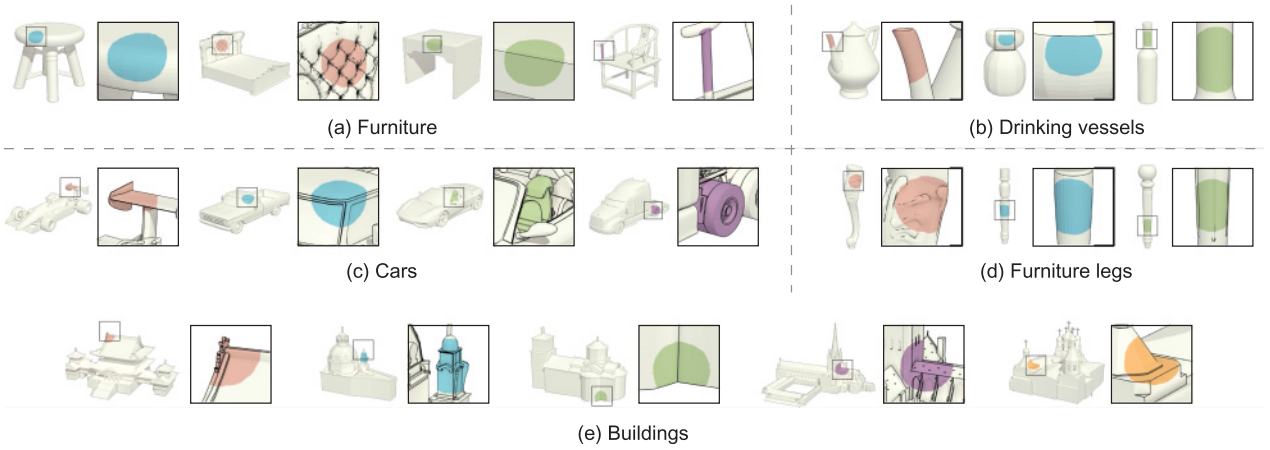

Fig. 11. Examples of style-defining elements selected by our method. We show one element per style. Note how the elements capture distinctive characteristics of each style.

consistent and common across styles. Therefore, for the evaluation, we opt to collect a ground-truth for a smaller, more tractable subset of shapes. In addition, we indirectly evaluate the correctness of the detected style-defining elements through shape classification tasks. We perform this second evaluation on the entire dataset without subsampling. These two evaluations are described as follows.

Comparison to ground-truth. We asked human subjects to create a ground-truth of defining elements for subsets sampled from each of our sets. Each subset contains three styles with 10 shapes per style, totaling 30 shapes. We randomly sampled shapes for each category, while ensuring that the selected subsets cover the diverse shape variations and types of content of the original sets. The subjects were computer science majors not involved in our project. We provided an interface where they could select defining elements from our initial elements. We show a screenshot of the interface in the Supplemental Material. Thus, the initial elements selected by the users are also patches of regular radius. This facilitates the comparison to the elements selected by our method and avoids considerable work from the users to draw irregular patches. We focus the evaluation on whether the users select the same regions of the shapes as our method, reflected by the centers of the elements. Each user was asked to select the five most defining elements for each style. Since we had five sets, one for each of the five object categories, and three styles, we obtained 75 elements per user. With 31 participants, we obtained a total of 2,325 elements.

We compare this ground-truth with the elements discovered by our method on the same subsets of shapes. One challenge is that we extract around 200 initial elements for each shape. Since some of them are geometrically similar, it is unlikely that two users as well as our method will select exactly the same elements. Nevertheless, if two elements are selected in a similar region of a shape and have similar characteristics, they should be considered as the same element, for example, any elements selected on the smooth portions of a stool's leg should be considered to be equivalent. Here, we could ask users to rate the element similarity. However, this is unpractical as hundreds of elements are involved. Thus, we use the framework of comparing elements with their individual similarity measures.

We train the measure for each defining element that we discovered, and also for each element selected by the users, using the refinement method introduced in Section 4. The general idea is illustrated in Figure 13: each element from either of the two sets has

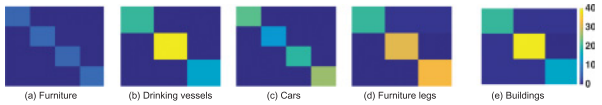

Fig. 12. Defining element frequency. Each entry (i, j) of a matrix shows the average percentage of defining elements of style i that appear in shapes of style j. We see how the diagonals dominate the matrices, indicating that a set of defining elements appears mostly in shapes of its respective style.

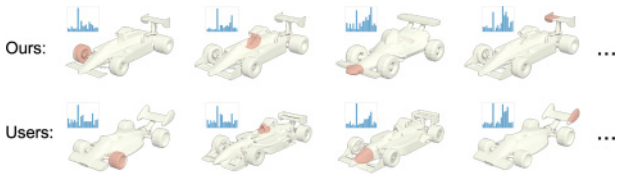

Fig. 13. Examples of feature weights that define the similarity measure of each element shown in red. The top row shows the elements discovered by our method, while the bottom row shows elements selected by the users.

an associated similarity measure, represented by a set of feature weights. For each set, we train the similarity of each element separately, using the element's nearest neighbors in the initial set. Note that we do not use any information discovered by our method to train the similarity measures of user-selected elements.

The problem of comparing elements is then reduced to comparing two sets, according to the individual similarity measures of the elements. For this task, we compute the agreement between the results of our method and the ground-truth in terms of recall and precision. Let $\mathcal{D}_g$ be the defining elements in the ground-truth and $\mathcal{D}_r$ the elements discovered by our method. We define

$$\text{Recall} = \sum_{e_g \in \mathcal{D}_g} \text{Similar}(\mathcal{D}_r, e_g)/|\mathcal{D}_g|, \tag{4}$$

where $\text{Similar}(\mathcal{D}_r, e_g)$ is 1 if there exists an $e_r \in \mathcal{D}_r$ such that $S_{e_r}(e_g) > \tau_p$, or 0 otherwise, with $\tau_p = -1$ as explained in Section 5. Note that, here, we use the similarity measures trained for the elements found by our method. Hence, recall corresponds to all of the elements in the ground-truth that also appear in our results.

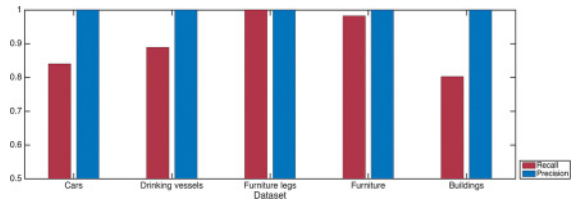

Fig. 14. Agreement between the defining elements selected by our method and a ground-truth created by human subjects, in terms of precision and recall. Note how the recall is at least 80% for all datasets, and the precision is 100%.

Fig. 15. Elements chosen by humans to compose the ground-truth, which were not selected by our method.

Similarly, we define

$$\text{Precision} = \sum_{e_r \in \mathcal{D}_r} \text{Similar}(\mathcal{D}_g, e_r)/|\mathcal{D}_r|, \tag{5}$$

which corresponds to all the elements in our results that are deemed to be "correct," that is, appear in the ground-truth, as recognized by the similarity measures trained for each ground-truth element. Note that we only compare positive elements discovered by our method, since users were instructed to only select positive elements.

We plot the precision and recall in Figure 14, for each category. We observe that, while our method recovers 80% or more of the elements selected by users, the method does not select any elements that were not chosen by users (precision is close to 100%). Thus, we conclude that, although the users selected some elements not found by our method, the elements discovered by our method are in close agreement with what users would select as defining elements. Moreover, in Figure 15, we show a few examples of the elements that users selected that were not found by our method. We observe that some of these elements are unique to only a few shapes, such as the element on the boundary between two parts of the wine glass. In contrast, our method selects elements on specific shape parts, such as the patches on the spouts of the drinking vessels in Figure 11(b), since these appear frequently across the set.

Similarity measures. Since the comparison to the ground-truth is computed with the similarity measure of each individual element, to further justify the use of the measures, we performed a second user study to confirm that the learned measures provide a reasonable assessment of stylistic element similarity. We sampled pairs of elements that are deemed either similar or dissimilar by the learned measures. We then showed users the same pairs of elements and asked them to assess whether the elements are similar or dissimilar in the context of a given style. To illustrate the context, a few shapes of the same style were also shown to the user. The users could also select a third option to indicate that they were unsure about the similarity. Finally, we computed the agreement between users and the learned measures.

We use the same set of shapes as in the previous user study (three styles for each of the five categories). We sampled 10 pairs of elements per style, totaling 150 pairs to evaluate, where half of the pairs were labeled as similar. About half of the pairs involve elements selected by our method, while the other half involve elements defined by the user ground-truth. We then asked 25 users to participate in the study and collected 60 answers from each user, obtaining 1,500 answers in total. The reason for the larger number of observations than pairs is that we asked 10 different users to classify each pair, to compute more reliable statistics. In this study, 18 users had a computer science background, while the remaining 7 users had various other backgrounds. A screenshot of the interface used for the user study is shown in the Supplemental Material.

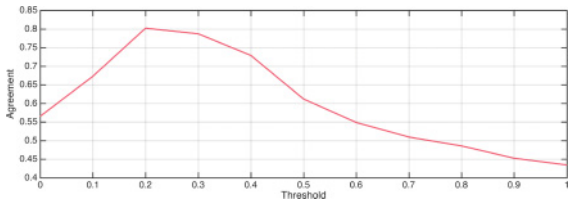

Fig. 16. Agreement between the users' similarity and a baseline similarity measure (the correlation distance between feature vectors of elements). Two elements are deemed similar according to a threshold on the correlation. We evaluate different thresholds (the x-axis). Note that the threshold 0.2 yields the maximal agreement of 80.0%, which is still lower than the agreement we obtain by using the SVM detectors learned for each element (85.5%).

We then compute the percentage of users that agree with the learned measures. We first eliminate inconsistent pairs (about 15% of all pairs), where a pair is said to be inconsistent if less than half of the users agree with whether the pair is similar or not. This situation can occur because each user could also select an "unsure" response. We then define a pair as in agreement with the learned measure, if the majority of users agree with the measure. With this calculation, we find that there is 85.5% agreement among the learned measures and the users' similarity, for all the sampled pairs.

To demonstrate that learning the similarity metrics per element is more effective than a simple baseline, we perform the same agreement calculation when using the correlation between feature vectors as a baseline element similarity distance. In this case, we need to select a threshold that indicates what correlation values imply that two elements are similar to each other. We show the result of this experiment in Figure 16 when choosing different threshold values. We observe that the optimal threshold is around 0.2, giving an agreement of 80%, which is still lower than the agreement of 85.5% provided by our approach of learning the similarity per element.

Style classification. We also evaluate our method by verifying the accuracy of classifying shapes into the different styles, according to our defining elements. Since the defining elements for the styles are different from each other, the term vectors of the shapes for a style are also different. Thus, to classify shapes into different styles, we set up a binary KNN classifier for each style. Given a specific style of one dataset, we take all the shapes and represent them with the term vectors of defining elements for the style, so we have positive and negative examples for training. Next, given an unknown shape, we represent it with a term vector for each style, by detecting the defining elements corresponding to each style, as explained in Section 5. Finally, by applying each classifier, we can estimate whether the shape belongs to a style or not.

As a result of the classification, a shape may get assigned to multiple labels. Thus, we verify the accuracy of the classification of a shape into the styles as follows. If a shape is assigned to its ground-truth style label, then we count a success to this label. However, if the shape is also assigned to labels other than its own by the classifiers of different styles, we count errors for those labels.

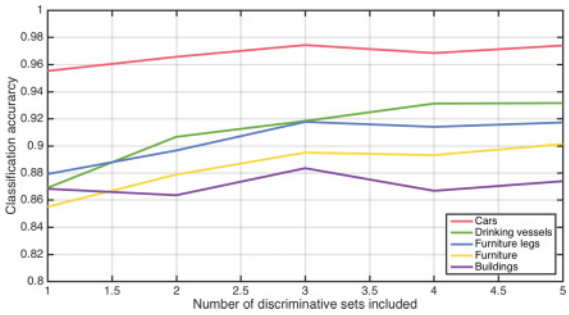

Fig. 17. Average accuracies of style classification in relation to the number of discriminative sets included in the final set of defining elements. We show an average for all the styles in each dataset.

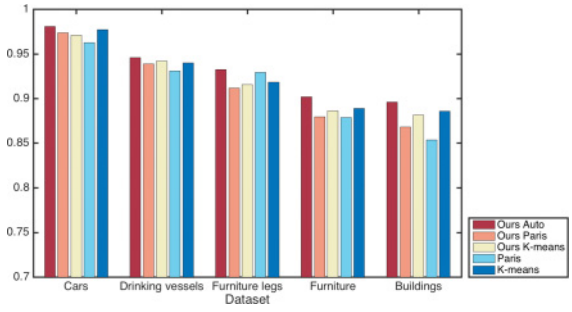

Fig. 18. Evaluation of different methods for selecting candidate elements. We observe that our method, denoted *Ours Auto*, leads to the best classification results. Please refer to the text for further details on the other methods.

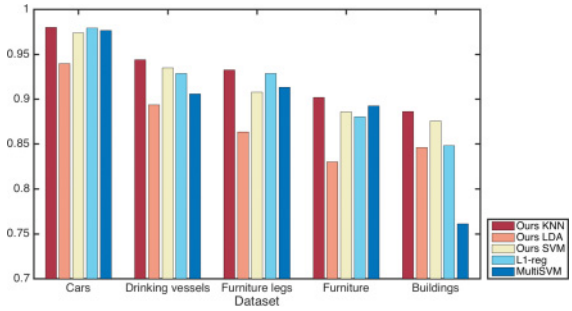

Fig. 19. Evaluation of different feature selection algorithms for the second step of our method. Note the better accuracy of our method when combined with a KNN classifier.

Moreover, to perform the evaluation on each dataset, we run a classification experiment with 10-fold cross-validation. That is, we randomly divide the input shapes into 10 folds of equal size. We learn the sets of defining elements from 9 folds used for training, and evaluate the classification accuracy on the remaining fold. We then average the accuracy for the 10 folds and all the style labels in a set.

We plot the results of this experiment for all datasets in Figure 17. The accuracy of classification is shown in relation to the number of discriminative sets that we select to compose the style-defining elements (x-axis). We see that the average accuracy is over 88%, when three sets are included. As expected, a few sets of discriminative elements are sufficient for classification, although having a more complete set of defining elements slightly increases the accuracy.

Evaluation of algorithm components and settings. To evaluate the effectiveness of the different components selected for our method, and to demonstrate that our method is robust under diverse settings, we compare our algorithm components to reasonable alternative approaches. We evaluate the first and second steps of our method, the candidate and feature selection, respectively. We also evaluate the construction of the initial elements, our shape representation, the effect that the size of the training set has on the results, and analyze the weights learned for the per-element similarity measures. For these components, we only provide a short summary here and give more details and evaluation plots in the Supplemental Material.

Candidate selection. We evaluate the first step of our method based on peak analysis by comparing it to two alternative approaches that could be used for candidate selection. *Paris* denotes a method where we select candidate elements with a sampling and filtering approach inspired by the work of Doersch et al. [5]. In short, we select an element if its k-nearest neighbors comprise only a few style labels and cover as many shapes as possible from the dominating styles. This implies that the element is weakly discriminative of a style. We provide more details on this method in the Appendix. *K-means* denotes a baseline method where we cluster the initial elements with k-means into 100 clusters. We then take the cluster centers as candidate elements. We also explore three different settings of our method. In *Ours Auto*, we select the candidate elements automatically, according to the density peak clustering [24]; in *Ours Paris*, we apply the filtering of the *Paris* method to the density peaks detected by our method; finally, in *Ours K-means*, we run our method requesting the same number of candidates as chosen for *K-means* (100 elements).

Figure 18 presents the result of this comparison. We observe that our method with automatic selection of peaks provides the best classification accuracy. We also see that our method with other settings is comparable for most sets. However, the filtering and k-means approaches lead to inferior results in some of the sets.

Feature selection. We evaluate the second step of our method by comparing it to two alternative approaches. *MultiSVM* denotes a method that uses a multi-label SVM classifier. Note that this method allows us to classify shapes into different styles, and thus we can compare its classification accuracy to our method. However, the multi-label classifier does not perform feature selection for each label and thus is not able to extract corresponding defining elements. *L1-reg* denotes an approach based on L1-regularized logistic regression [25]. We perform the L1 minimization on each style. This method can be used for feature selection by retrieving the elements whose optimized weights are non-zero. We also evaluate the second step of our method with different settings, where we substitute the classifier in our feature selection wrapper with three options: we consider a k-nearest neighbor classifier (denoted *Ours KNN*), a classifier based on discriminant linear analysis (*Ours LDA*), and an SVM classifier (*Ours SVM*).

Figure 19 shows the result of this comparison. We observe that our method leads to the best classification accuracies, especially when combined with a KNN classifier, although other classifiers and the L1-reg approach are comparable on several of the sets. This is reasonable, since all of the methods start from the same set of initial elements, and any effective feature selection method should lead to satisfactory results in this setting. However, it is worth mentioning that the comparison is in terms of shape style classification. When localized features are required, the L1-reg approach provides discriminative elements, but does not necessarily build a more

complete set of defining elements. For all the other experiments in the article, our method is used with the KNN classifier.

For the following evaluations, we only provide a summary in the article. More details are provided in the Supplemental Material.

Geodesic radius of elements. We evaluate the effect of different values of the main parameter used in the element construction, the geodesic radius τ of the patches, on the accuracy of classification. We observe that a radius of 0.09 leads to the overall best results, although different radii do not affect the accuracy by more than 5%. Thus, we use 0.09 in all of the other experiments in the article.

Shape representations. We test two different representations to encode shapes in our method. We compare our bag-of-words encoding using element frequency with the encoding used by Arietta et al. [1], based on the scores returned by the per-element similarity measures. Although the two approaches are comparable for three sets, our representation leads to the best overall results.

Dataset size. We study how the size of the input dataset influences the classification accuracy, since the learning of the element similarity changes based on the input data provided. In an experiment with cross-validation, we observe that the classification accuracy is over 85% for training sets containing 50% of the shapes in the dataset, which corresponds to approximately 300 shapes for the larger set of furniture and 40 shapes for the smaller set of drinking vessels. We also observe that furniture require the most training data, while cars have a higher accuracy with a smaller number of training shapes. From this result, we speculate that the styles of cars are more distinctive and easily distinguished from each other, as appears to be the case from a visual analysis of the set.

Feature weights. Regarding the learning aspects of the method, we also investigate how the different features used to represent the elements are weighted by the similarity measures learned for each element. We observe that all of the weights are non-zero, implying that each feature is relevant for at least one of the datasets.

7. APPLICATIONS

We first present an application that uses the term vectors of the shapes for global style analysis. We then show several applications that benefit from the localization of elements.

Stylistic score. In addition to classification, we can use the representation of a shape to analyze how strongly the shape fits into a specific style. We retrieve the term vector of a shape relative to a specific style and count how many defining elements appear in the shape, that is, the sum of all the word frequencies. We use the resulting sum as a score that allows us to determine how strongly the shape fits into the given style, based on the defining elements that we detected. In Figure 20, we plot these scores on a graph for the shapes in the furniture set and two selected styles. We see how the European furniture shapes appear on the right side of the graph, implying that they fit the most within the European style. Similarly, we observe how the Ming furniture shapes occupy the top of the graph. The shapes of the Children and Japanese furniture styles have lower scores for the two styles. Compared to the L1-reg method, our grouping gives a clearer separation between the different styles, demonstrating the effectiveness of using the defining elements. Note that this type of embedding cannot be achieved with previous work, as it is different from a traditional embedding reflecting shape similarity.

Style-revealing view selection. The defining elements can also be used to select the pose of a shape that best reveals its style, that is,

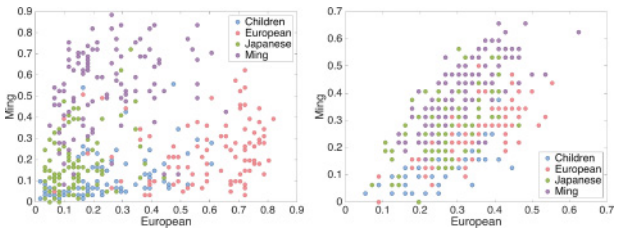

Fig. 20. Graphs reflecting how shapes from the furniture set fit within two selected styles. We compare our method (left) with L1-reg (right). Each marker is a shape, and the larger the values along an axis, the stronger the shape's relationship to the corresponding style. We observe how the grouping given by our method leads to a clearer separation between the different styles.

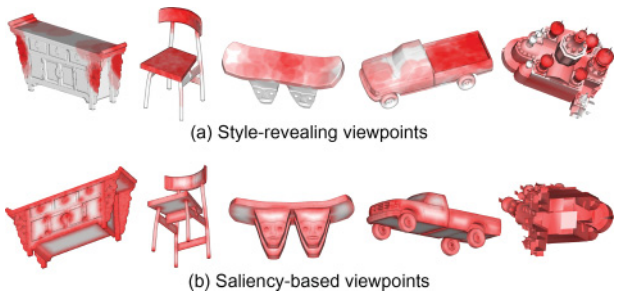

Fig. 21. Style-revealing view selection based on our defining elements in (a), compared to view selection with saliency in (b). The scalar field goes from white to red and denotes the level of stylistic relevance of each point, in (a), or saliency, in (b). We observe how the views selected by our method are more informative and do not occlude the stylistic shape regions.

the pose that clearly demonstrates that the shape possesses a given style. For this application, we first derive a style-revealing scalar field for a shape from the defining elements detected for its style. As explained in Section 4, each shape is represented as a set of points. Thus, given a style-defining element, we place a vote on all the shape points covered by this element. After voting for all the positive defining elements detected on the shape, we normalize the votes to the range [0, 1], to obtain a scalar field where points with values closer to 1 are more relevant to the style of the shape.

To select the style-revealing view of the shape, we place the shape at its center of mass and uniformly sample rotations in 3D space. We then choose the rotation that maximizes the sum of the scalar field for all the points that appear on the projected image of the shape, according to a fixed camera. In Figure 21(a), we show examples of views selected for different shapes, with their style-revealing scalar fields. In Figure 21(b), we show a comparison to the method of Shtrom et al. [28], which selects viewpoints by maximizing saliency. We observe that the views selected by our method ensure that regions of the shapes that are important for defining their styles are visible, such as the roofs of buildings and finishings of the furniture.

Style-aware sampling. Using the scalar field defined for the application of view selection, we can also perform style-aware sampling, to place more samples on regions that are relevant to the style of a shape. This can be useful for applications such as style-aware feature extraction and style-aware simplification, where we decimate a mesh while avoiding over-simplifying regions that contain a higher density of samples that are important for preserving the style of the shape. We perform the sampling with the method of Corsini et al. [4],

Fig. 22. Style-aware sampling. For each shape, we show our style-revealing scalar field on the left and the sampling on the right.

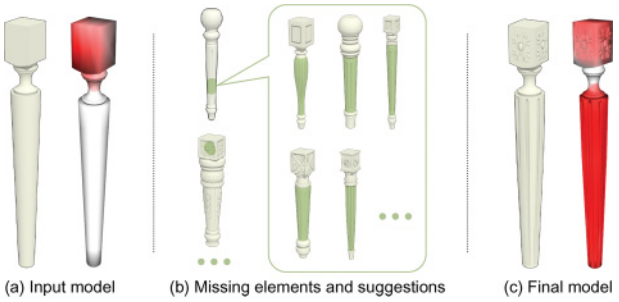

Fig. 23. Style-driven modeling enabled by our defining elements. Given the shape in (a), our system suggests positive style-defining elements that can be added to the shape to enhance its style, as seen in (b). The suggested elements are shown on their source shapes. The system also locates negative defining elements on the input shape that can be removed to improve its style. An artist can then choose to add or remove elements to create a shape with a more pronounced style, shown in (c). We plot the style-revealing scalar fields on the input and final shapes to show that the style of the shape has become more distinct after the modifications.

where we set the sampling importance of the points according to our style-revealing scalar field. Figure 22 presents a few examples of samplings, where we sampled 2,000 points per shape. We note how the samples follow the scalar field closely. As a consequence, the sample density is also higher near style-defining elements.

Style-driven modeling. If the style of a shape is not pronounced, then our defining elements can guide an artist in giving a more prominent style to the shape. We created a system based on the defining elements to facilitate modeling in such a context, which is illustrated in Figure 23. Given an input shape and a reference style, the system suggests positive defining elements that can be added to the shape to enhance its style. The suggested elements are shown on their source shapes, to provide a context to the artist of where to add the elements to the shape. The system also locates and suggests elements on the input shape that should be removed to enhance the style, which correspond to negative defining elements of the style. By following these suggestions, an artist can transfer geometric patches to the input shape and add or remove parts, to make the style of shapes more pronounced, as shown with the examples in Figure 24. For example, a bed is made to look more European by adding ornaments to its frame, or a stool is made more Children-like by removing the connecting bars between legs, since they are the locations of negative elements of the style. Thus, while the artist may start with a vague notion for modeling such as "I want to make this Japanese chair look more like a Ming chair," our system explicitly presents the *what* and *where* of the style to the artist, providing valuable visual guidance for modeling.

8. CONCLUSIONS AND FUTURE WORK

In this work, we present a method for discovering defining elements that characterize a given style, where the elements are co-located

Fig. 24. Style-driven modeling results. In the top row, we show how an artist transformed a Japanese chair into a Ming chair, with the style-elements suggested by our system shown in the box in the middle. In the remaining rows, we see additional modeling examples. For each pair, an artist added or removed defining elements to the shape on the left to obtain the shape with new style on the right. Removed elements are circled in red.

across the shapes of the style. The style-defining elements provide a more complete characterization than simply a set of discriminative elements and include positive and negative elements of the style. We demonstrate with a qualitative and quantitative evaluation that our method is effective in discovering these elements, while the elements themselves reveal features of the shapes that are relevant to the styles. We also present examples of style-aware applications that benefit from the relevance and localization of the elements.

Limitations. The main technical limitation of our work is that we prescribe an initial set of elements from where our method selects the defining elements. The prescribed elements are not universal, since the meaning of "style" can be vague. For example, styles that are defined by relative part proportions [33] cannot be captured by our set of elements. In more general terms, our style elements are isotropic and of a local nature, based mainly on the geometry of shapes. We focus on local elements, since several authors characterize the style of works of art based on local patterns, such as the categorization of furniture based mainly on the type of feet and finials that appear on the shapes [21]. Thus, our analysis does not consider stylistic features that are anisotropic, structural, and more global, which could complement the description of certain shape styles.

Moreover, one criterion that we use for the selection of defining elements is that they should appear frequently across the shapes of a style. Thus, our method may miss known defining elements, if they only appear seldom in every shape, such as the spires of Gothic cathedrals. In addition, a limitation of our evaluation is that we compared our results to a relatively small ground-truth created by humans, which is composed of 150 shapes. To assess our method in large-scale scenarios, we may need considerably more user input to assemble a large collection of ground-truth dataset.

Future work. We would like to extend our method to analyze and localize shape categorizations beyond style. One such possibility is to extract elements that define both style and content. In the

Supplemental Material, we show a few preliminary results toward this direction, by taking style-content tables as input. Applying our method to other settings may also require extending the types of elements that we use, to capture additional geometric and structural properties of the shapes that could be relevant to the specific categorization studied.

Moreover, the geometric patches that we currently use can be extended in various manners. Some of the defining elements that our method selects look visually similar to each other, although their corresponding weight vectors $\mathbf{w}_i$ are different; we present such an example in the Supplemental Material. This implies that when their similarity to other elements is being computed, the measure gives relevance to different types of features. Thus, we could explore these low-level feature differences to infer what kinds of geometric characteristics are being emphasized in each element. We could also learn a more advanced model for the detection of elements. For example, by learning the valid degrees of variation of each defining element, we could define a general model of elements that captures how they vary across diverse shapes. We would also like to consider developing anisotropic patch elements, and elements that capture structural properties of the shapes, such as part composition and symmetry.

Due to the need for domain knowledge in identifying many of the studied shape styles, for example, Japanese furniture, cabriole table legs, or Baroque architecture, we relied on expert annotations to form a ground-truth style grouping for our analysis task. Some recent works on style analysis, for example, References [16, 17], utilized crowdsourcing instead, where non-experts were asked to rank style similarities or compatibilities over triplets of shapes. Crowdsourced data should generally be expected to be noise-prone, and even more so when collected from a large crowd of non-experts on style grouping over moderately large shape collections. On the other hand, crowdsourcing can easily provide a large number of annotations, which can then be filtered for consistency. Thus, it would be interesting, although not straightforward, to explore how to turn a set of style-ranked triplets into a style grouping that would serve as input to our style analysis.

One final observation is that the defining elements of a style that we select with our method are not intrinsic to the style but are defined relative to other styles (the positive and negative elements). Thus, it may be interesting to explore the question of whether defining elements can be defined in an intrinsicmanner and extracted from a set of example shapes with a single style.

APPENDIX

SAMPLING AND FILTERING OF ELEMENTS

In this appendix, we describe the filtering approach inspired by the work of Doersch et al. [5], which can be used as an alternative for the selection of candidate elements. The goal of this approach is to select elements that weakly discriminate the styles. Thus, we seek elements whose neighborhoods are dominated by only a few style labels and at the same time cover as many shapes as possible. Toward this goal, we randomly sample 1,000 elements $\mathcal{I}'$ from the set of initial elements $\mathcal{I}$. Given a sampled element, we select its k-nearest neighbors from the full set of elements $\mathcal{I}$, according to the normalized correlation. We then analyze this neighborhood to compute a histogram H_N of style labels of the neighbors, and another histogram H_S of shape coverage by the neighbors. Specifically, H_S records the percentage of shapes from each style that is covered by the elements of the neighborhood. We then rank the elements based on $E(H_N) \cdot E(H_S)$, where E is the entropy of the histogram. The lower this product is, the higher the element is ranked. Finally, we pick the top ranked 100 elements to form $\mathcal{C}$, avoiding the duplication of any choice. Two elements are considered to be duplicated if their neighborhoods overlap in more than 30%, where the overlap of two neighborhoods is defined as the percentage of elements that appear in both neighborhoods. Note that, when we combine our method with this filtering, we replace $\mathcal{I}'$ with the density peaks, while the rest of the method remains the same.

ACKNOWLEDGMENTS

We thank the reviewers for their comments and suggestions.

REFERENCES

[1] Sean Arietta, Alexei A. Efros, Ravi Ramamoorthi, and Maneesh Agrawala. 2014. City forensics: Using visual elements to predict non-visual city attributes. *IEEE Trans. Visual. Comput. Graph.* 20, 12 (2014), 2624–2633.

[2] Shai Avidan and Ariel Shamir. 2007. Seam carving for content-aware image resizing. *ACM Trans. Graph. (Proc. SIGGRAPH)* 26, 3 (2007), 10:1–9.

[3] Itamar Berger, Ariel Shamir, Moshe Mahler, Elizabeth Carter, and Jessica Hodgins. 2013. Style and abstraction in portrait sketching. *ACM Trans. Graph. (Proc. SIGGRAPH)* 32, 4 (2013), 55:1–12.

[4] M. Corsini, P. Cignoni, and R. Scopigno. 2012. Efficient and flexible sampling with blue noise properties of triangular meshes. *IEEE Trans. Visual. Comput. Graph.* 18, 6 (2012), 914–924.

[5] Carl Doersch, Saurabh Singh, Abhinav Gupta, Josef Sivic, and Alexei A. Efros. 2012. What makes Paris look like Paris? *ACM Trans. Graph. (Proc. SIGGRAPH)* 31, 4 (2012), 101:1–9.

[6] Eric Fernie. 1995. *Art History and Its Methods: A Critical Anthology*. Phaidon, London, 361.

[7] William T. Freeman, Joshua B. Tenenbaum, and Egon C. Pasztor. 2003. Learning style translation for the lines of a drawing. *ACM Trans. Graph.* 22, 1 (2003), 33–46.

[8] Elena Garces, Aseem Agarwala, Diego Gutierrez, and Aaron Hertzmann. 2014. A similarity measure for illustration style. *ACM Trans. Graph. (Proc. SIGGRAPH)* 33, 4 (2014), 93:1–9.

[9] Ernst H. Gombrich. 1968. "Style". In *International Encyclopedia of the Social Sciences*, D. L. Sills (Ed.). Vol. 15. Macmillan, New York, 352.

[10] Aaron Hertzmann, Nuria Oliver, Brian Curless, and Steven M. Seitz. 2002. Curve analogies. In *Proceedings of the European Group Workshop on Rendering*. 233–246.

[11] Katrin Lang and Marc Alexa. 2015. The Markov pen: Online synthesis of free-hand drawing styles. In *Proceedings of the Symposium on Non-Photorealistic Animation and Rendering (NPAR'15)*. 203–215.

[12] Chang Ha Lee, Amitabh Varshney, and David W. Jacobs. 2005. Mesh saliency. *ACM Trans. Graph. (Proc. SIGGRAPH)* 24, 3 (2005), 659–666.

[13] Y. J. Lee, A. A. Efros, and M. Hebert. 2013. Style-aware mid-level representation for discovering visual connections in space and time. In *Proceedings of the International Conference on Computer Vision (ICCV'13)*. 1857–1864.

[14] H. Li, H. Zhang, Y. Wang, J. Cao, A. Shamir, and D. Cohen-Or. 2013. Curve style analysis in a set of shapes. *Comput. Graph. Forum* 32, 6 (2013), 77–88.

[15] Roee Litman, Alex Bronstein, Michael Bronstein, and Umberto Castellani. 2014. Supervised learning of bag-of-features shape descriptors using sparse coding. *Comput. Graph. Forum* 33, 5 (2014), 127–136.

[16] Tianqiang Liu, Aaron Hertzmann, Wilmot Li, and Thomas Funkhouser. 2015. Style compatibility for 3D furniture models. *ACM Trans. Graph. (Proc. SIGGRAPH)* 34, 4 (2015), 85:1–9.

[17] Zhaoliang Lun, Evangelos Kalogerakis, and Alla Sheffer. 2015. Elements of style: Learning perceptual shape style similarity. *ACM Trans. Graph. (Proc. SIGGRAPH)* 34, 4 (2015), 84:1–14.

[18] Zhaoliang Lun, Evangelos Kalogerakis, Rui Wang, and Alla Sheffer. 2016. Functionality preserving shape style transfer. *ACM Trans. Graph.* 35, 6 (2016), 209:1–14.

[19] Chongyang Ma, Haibin Huang, Alla Sheffer, Evangelos Kalogerakis, and Rui Wang. 2014. Analogy-driven 3D style transfer. *Comput. Graph. Forum (Proc. Eurographics)* 33, 2 (2014), 175–184.

[20] L. Majerowicz, A. Shamir, A. Sheffer, and H. H. Hoos. 2014. Filling your shelves: Synthesizing diverse style-preserving artifact arrangements. *IEEE Trans. Visual. Comput. Graph.* 20, 11 (2014), 1507–1518.

[21] Wallace Nutting. 1968. *Furniture Treasury*. Vol. 3. Macmillan Publishing.

[22] Peter O'Donovan, Jānis Lībeks, Aseem Agarwala, and Aaron Hertzmann. 2014. Exploratory font selection using crowdsourced attributes. *ACM Trans. Graph. (Proc. SIGGRAPH)* 33, 4 (2014), 92:1–9.

[23] Hanchuan Peng, Fuhui Long, and Chris Ding. 2005. Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Pattern Anal. Mach. Intell.* 27, 8 (2005), 1226–1238.

[24] Alex Rodriguez and Alessandro Laio. 2014. Clustering by fast search and find of density peaks. *Science* 344, 6191 (2014), 1492–1496.

[25] Mark Schmidt, Glenn Fung, and Romer Rosales. 2007. Fast optimization methods for L1 regularization: A comparative study and two new approaches. In *Proceedings of the European Conference on Machine Learning*. 286–297.

[26] Philip Shilane and Thomas Funkhouser. 2007. Distinctive regions of 3D surfaces. *ACM Trans. Graph.* 26, 2 (2007), 7:1–15.

[27] Abhinav Shrivastava, Tomasz Malisiewicz, Abhinav Gupta, and Alexei A Efros. 2011. Data-driven visual similarity for cross-domain image matching. *ACM Trans. Graph. (Proc. of SIGGRAPH Asia)* 30, 6 (2011), 154:1–10.

[28] E. Shtrom, G. Leifman, and A. Tal. 2013. Saliency detection in large point sets. In *Proceedings of the International Conference on Computer Vision (ICCV'13)*. 3591–3598.

[29] Denis Simakov, Yaron Caspi, Eli Shechtman, and Michal Irani. 2008. Summarizing visual data using bidirectional similarity. In *Proceedings of the Conference on Computer Vision and pattern Recognition (CVPR'08)*. 1–8.

[30] J. W. H. Tangelder and R. C. Veltkamp. 2008. A survey of content based 3D shape retrieval methods. *Multimed. Tools Appl.* 39, 3 (2008), 441–471.

[31] R. Toldo, U. Castellani, and A. Fusiello. 2010. The bag of words approach for retrieval and categorization of 3D objects. *Vis. Comput.* 26, 10 (2010), 1257–1268.

[32] Andrew R. Webb and Keith D. Copsey. 2011. *Statistical Pattern Recognition* (3rd ed.). Wiley.

[33] Kai Xu, Honghua Li, Hao Zhang, Daniel Cohen-Or, Yueshan Xiong, and Zhi-Quan Cheng. 2010. Style-content separation by anisotropic part scales. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)* 29, 6 (2010), 184:1–10.

Received September 2016; revised February 2017; accepted March 2017