# Semi-Supervised Co-Analysis of 3D Shape Styles from Projected Lines

FENGGEN YU, State Key Lab for Novel Software Technology, Nanjing University
YAN ZHANG*, State Key Lab for Novel Software Technology, Nanjing University
KAI XU†, National University of Defense Technology
ALI MAHDAVI-AMIRI, Simon Fraser University
HAO ZHANG, Simon Fraser University

We present a *semi-supervised co-analysis* method for learning 3D shape styles from *projected feature lines*, achieving style patch *localization* with only weak supervision. Given a collection of 3D shapes spanning multiple object categories and styles, we perform style co-analysis over projected feature lines of each 3D shape and then backproject the learned style features onto the 3D shapes. Our core analysis pipeline starts with mid-level patch sampling and pre-selection of candidate style patches. Projective features are then encoded via patch convolution. Multi-view feature integration and style clustering are carried out under the framework of *partially shared latent factor* (PSLF) learning, a multi-view feature learning scheme. PSLF achieves effective multi-view feature fusion by distilling and exploiting consistent and complementary feature information from multiple views, while also selecting style patches from the candidates. Our style analysis approach supports both unsupervised and semi-supervised analysis. For the latter, our method accepts both user-specified shape labels and style-ranked triplets as clustering constraints. We demonstrate results from 3D shape style analysis and patch localization as well as improvements over state-of-the-art methods. We also present several applications enabled by our style analysis.

Additional Key Words and Phrases: Style analysis of 3D shapes, projective shape analysis, semi-supervised learning

**ACM Reference format:**
Fenggen Yu, Yan Zhang, Kai Xu, Ali Mahdavi-Amiri, and Hao Zhang. 2018. Semi-Supervised Co-Analysis of 3D Shape Styles from Projected Lines. 1, 1, Article 1 (February 2018), 17 pages.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

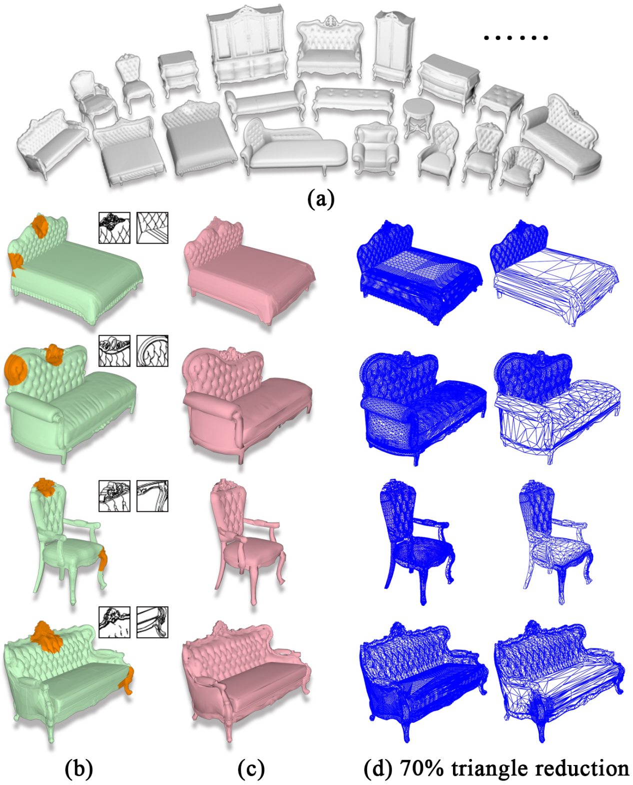

Fig. 1. Given a heterogeneous 3D shape collection (a), we perform style co-analysis over projected feature lines (see insets) to spatially located style patches (b) and cluster the shapes based on their styles — all the four shapes in color belong to the same cluster. Spatial localization of style patches enables applications such as style-preserving mesh simplification (c-d). Note the denser triangle distributions near style patches (d).

## 1 INTRODUCTION

Styles are generally regarded as distinctive and recognizable forms which permit the grouping of entities containing these forms into related categories [Wikipedia 2016]. It follows that stylistic forms that serve to characterize a common style tend to share strong similarities, while between different style categories, these forms often exhibit clear distinctions. As a result, style analysis is best conducted in the context of a *set* of entities and naturally lends itself as a *clustering* problem. For 2D or 3D shapes, the shape styles are typically perceived by humans as apparent geometric features or patterns; see Figure 1 (left). The ability to extract such style features allows them to be compared, altered, or preserved.

Clustering analysis has been performed in earlier works on shape styles. However, the studied styles were either pre-determined [Xu et al. 2010] or characterized by hand-crafted rules [Li et al. 2013]. Most recent attempts have been on supervised learning of style similarity [Garces et al. 2014; Liu et al. 2015a; Lun et al. 2015] via crowd-sourcing to collect user-specified style rankings and then performing metric learning rather than style clustering. But these works do not spatially locate the stylistic features or patches over the analyzed shapes. In a most recent work, also based on supervised learning, Hu et al. [2017] learn to spatially locate style-defining elements or patches over a set of 3D shapes, where an expert-specified style clustering is given over the shape collection.

In this paper, we are interested in exploring a "middle ground", via *semi-supervised* learning with *weak supervision*, for generic style

*Fenggen Yu and Yan Zhang are co-first authors.
†Corresponding author: Kai Xu (kevin.kai.xu@gmail.com)

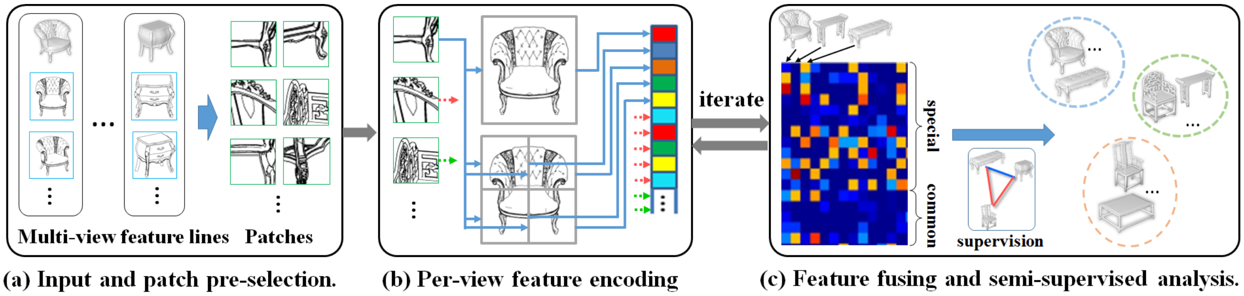
Multi-view feature lines
Patches
iterate
special
common
supervision
(a) Input and patch pre-selection.
(b) Per-view feature encoding
(c) Feature fusing and semi-supervised analysis.

Fig. 2. Our style co-analysis algorithm contains three stages: (a) Patch sampling and pre-selection and candidate style patches. (b) View feature encoding based on patch convolution, and (c) Multi-view feature integration using partially shared latent factor (PSLF) learning. The PSLF performs unsupervised or semi-supervised style clustering and patch filtering in an interleaving fashion.

analysis of 3D shapes. Semi-supervised learning is attractive as it can take advantage of strong techniques for unsupervised clustering and discriminative analyses without the need to collect large amounts of user data. After all, style analysis is essentially a grouping problem, while style patch extraction is a discriminative feature selection problem. That said, with semi-supervised learning, human is not out of the loop. The learning process naturally incorporates user feedback to reflect the subjective nature of style perception, while keeping such feedback to a minimum.

Specifically, we introduce a semi-supervised *co-analysis* method which simultaneously achieves style clustering and style patch localization, with only weak supervision over a *heterogeneous* collection of 3D shapes spanning multiple object categories and multiple styles. Unlike Hu et al. [2017], the input collection is not clustered by experts and our method supports both unsupervised and weakly-supervised analysis with minimal style annotation. In terms of shape styles, like all previous works [Hu et al. 2017; Liu et al. 2015a; Lun et al. 2015], our analysis also focuses on *element-level* styles of 3D shapes [Lun et al. 2015] which are *decorative* in nature. Such styles include those that are perceivable as patterns along shape contours or over shape surfaces. They are ubiquitous in man-made shapes including furniture, buildings, automobiles, kitchen utensils, and many engineering products and household items.

Our core analysis problem consists of a *clustering* of the input shape collection and a *selection of style feature* patches from the shapes which accentuate the shape clustering. While latent features are sufficient for style comparisons, to spatially locate shape styles, one must eventually extract and discriminate between *spatially explicit* or *visually apparent* shape features. Working with these features for style analysis is also well motivated by the visual and perceptual nature of style recognition by humans: styles are *seen* as visual patterns. In our work, we perform style analysis of 3D shapes via *projective* analysis. Specifically, we project a 3D shape over different views and work with projections of geometric feature lines detected over the 3D shape in object space for our style analysis and the extraction of shape styles.

Even though the projected feature lines do not exist in the "real world", they are believed to possess deep similarities to other more detailed and explicit visual representations as well as real scenes they depict [Sayim and Cavanagh 2011]. In addition, feature lines are remarkably efficient at conveying shape and meaning while reducing visual clutter [Rusinkiewicz et al. 2008]. For our purpose, feature lines are well-suited to depict decorative style patches of 3D shapes. It is worth emphasizing that our feature lines are detected in 3D object space based on geometry analysis (and then projected); they tend to be more reliable than lines extracted from rendered images which could be influenced by illumination and viewing artifacts. On the technical front, projective analysis puts many effective learning methods designed for image data, e.g., [Bansal et al. 2015; Gong et al. 2014; Li et al. 2015], at our disposal. Furthermore, it works more robustly with different 3D geometry representations and various shape imperfections including noise, incompleteness of shapes, and non-manifold geometries [Wang et al. 2013].

Given a heterogeneous collection of 3D shapes spanning multiple object categories and styles, our method performs style co-analysis over projected feature lines from each 3D shape and backproject the learned style features onto the 3D shapes at the end. As shown in Figure 2, our core analysis pipeline consists of three stages: 1) mid-level patch sampling and pre-selection of candidate style patches; 2) view feature encoding based on patch convolution; and 3) multi-view feature fusion and style clustering under the framework of *partially shared latent factor learning* or PSLF [Liu et al. 2015b], which selects the final style patches from the candidates.

PSLF fits well with projective analysis, as it is designed for multi-view feature analysis and learning. As a means for feature fusion, PSLF deciphers the *consistent* and *complementary* information from multi-view features and integrates them informatively. We show that it is competitive with conventional feature fusion approaches, in particular, max-pooling in multi-view CNN [Su et al. 2015]. Furthermore, PSLF discovers shape styles by clustering shapes and selecting the most discriminative mid-level patches which accentuate the clustering; this is consistent with how styles are typically characterized [Wikipedia 2016]. This makes our PSLF-based feature learning and encoding more interpretable, especially in the context of view-based 3D shape style analysis, differentiating it from widely adopted end-to-end deep learning methods. To support both unsupervised and semi-supervised style analysis, we develop a *constrained* formulation of PSLF which accepts both user-specified shape style labels [Hu et al. 2017] and style-ranked triplets as classical metric learning [Garces et al. 2014; Liu et al. 2015a; Lun et al. 2015]. This represents a key difference to most deep learning based solutions which rely on strong supervision.

Figure 1 shows a sample result, where our style co-analysis was performed on 400 mixed furniture pieces (top row). Four shapes deemed to belong to the same style cluster (not all shapes in the cluster appear in the figure) are shown with their style patches highlighted, both on the shape and also in feature lines (insets).

Our work makes the following main contributions:

- To the best of our knowledge, our method represents the first semi- or weakly-supervised co-analysis of 3D shape styles, leading to style clustering and style localization.
- Our method supports both semi-supervised and unsupervised style analysis, combining local feature learning and global discriminative style extraction.
- Our analysis focuses exclusively on projective feature lines, while previous works employed much richer feature sets [Hu et al. 2017; Liu et al. 2015a; Lun et al. 2015]. Yet, as we shall demonstrate, we can obtain clear improvements over all of these methods, owing in part to our multi-view style analysis with advanced feature fusion via PSLF.
- We can spatially locate visually apparent stylistic shape elements or patches without any direct user involvement to manually mark any style patches over 3D shapes. Our semi-supervised analysis takes the same types of user input as classical metric learning.

We demonstrate the effectiveness of our method for style analysis and patch localization, in particular, clear improvements over state-of-the-art supervised methods [Hu et al. 2017; Lim et al. 2016; Liu et al. 2015a; Lun et al. 2015]. We also develop several applications that can take advantage of the detected styles. For example, with the style patches spatially located, we can perform style-preserving mesh simplification, as shown in Figure 1. Triangle distributions before and after simplification, shown in Figure 1(d), clearly exhibit that triangle reduction happens mostly over non-style regions.

## 2 RELATED WORK

In this section, we cover and discuss works related to shape style analysis and our approach for style clustering and style patch extraction on 3D surfaces via machine learning. We describe how our method is different from these existing approaches.

*Co-analysis.* Our approach falls in the realm of co-analysis techniques [Mitra et al. 2013; Xu et al. 2017], most of which have been designed to work with homogeneous shape collections. Our method can work with a heterogeneous shape collection owing to its localized feature encoding and analysis of decorative styles. Semi-supervised learning has been mainly employed to solve labeling problems such as shape segmentation [Wang et al. 2013] and classification [Huang et al. 2013]. In our work, we rely on unsupervised and semi-supervised PSLF learning to extract spatial style patches.

*Projective shape analysis.* Analyzing 3D shapes through 2D projections has been a common practice with successful applications such as shape classification [Su et al. 2015], retrieval [Chen et al. 2003; Xie et al. 2015], and segmentation [Wang et al. 2013], to name a few. Main benefits of the projective approach include robustness with imperfect 3D representations and exploitation of image-based learning, especially deep learning techniques. Our work offers a new application, namely analysis of shape styles, by utilizing another useful property of 2D projections, i.e, their ability to reveal shape styles visually in the form of feature lines.

*Multi-view learning.* In multi-view learning, a concept is learned from data represented in multiple forms or views, e.g. [Chaudhuri et al. 2009; Liu et al. 2015b]. The goal is to discover *consistent* and *complementary* information among multiple views of the data, with both types of information supporting the concept. Specifically, consistent information should be shared across most views in identifying the concept, while complementary information is something that is distinctively reflected from one or few views and complementary to other such information. In our work, the concept to learn is shape styles and the multiple views are provided by the multi-projection feature lines of the 3D shapes. Decorative shape styles may be shared consistently across multiple views, e.g., stylistic details over the surface of a sofa. They can be also exclusive to one or few views to complement other style features such as an emblem on top of a bed's headboard. As a result, our style analysis problem is well-suited for a multi-view learning approach.

*PSLF.* Partially shared latent factor (PSLF) learning is a multi-view learning method [Liu et al. 2015b] which performs joint analysis over a set of data to extract *both* the consistent and complementary information among multiple data views. PSLF is essentially a dimensionality reduction technique that is realized through a non-negative matrix factorization (NMF). Our method adopts PSLF to learn style features and their spatial locations via projective co-analysis. We also adjust the original objective function of PSLF to enable semi-supervised learning that accepts both user-specified shape labels and style-ranked triplets as classical metric learning.

*Mid-level patches.* Mid-level image patches, e.g., object parts or salient regions, are neither too local nor too global and they have been effective for tasks such as object detection [Bansal et al. 2015], indoor scene classification [Doersch et al. 2013], and unsupervised visual discovery [Raptis et al. 2012; Singh et al. 2012]. While unsupervised approaches typically perform the analysis purely at the patch level [Singh et al. 2012], weakly supervised approaches such as those presented by [Bansal et al. 2015; Doersch et al. 2013], typically detect mid-level patches or compute patch clusterings to attain maximal adherence to the image or object labels.

Most closely related to our method is the work by Lee et al. [2013], which aims to discover mid-level patches that are the characteristic of historic and geographic styles of objects in images. They start with a generic visual element detector serving a similar role as our pre-selection of patches, and then rely on image labels of date and location to train a regression model to identify style-aware mid-level patches. In contrast, we rely on PSLF to perform shape clustering and style patch selection in an interleaving manner. Our approach can be supervised or semi-supervised, and for the latter, both style labels and user-specified style rankings are accommodated.

*Convolutional activation features.* Convolutional neural networks (CNNs) have been extensively employed for various feature learning tasks recently, e.g., [Donahue et al. 2013; Razavian et al. 2014; Zeiler and Fergus 2014]. Training a full CNN for feature learning is expensive in terms of data labeling and computation. In our work, we explore the limit of unsupervised and semi-supervised feature learning for shape style analysis requiring minimal data annotations.

Deep convolutional activation features, including those for mid-level visual elements, have been employed as descriptors for generic visual recognition [Bansal et al. 2015; Gong et al. 2014; Li et al. 2015]. We use the discriminatively detected mid-level patches as filters to perform feature encoding based on a *sliding-window convolutional* operation, similar to [Bansal et al. 2015]. Although these content features are discriminative for object characterization, it is not yet clear whether they would attain the same level of success for style recognition since style and content features do not always correlate with each other. We use these features as per-view initial features and conduct multi-view feature fusion and selection via PSLF.

*Shape style analysis.* Some earlier works on shape styles assume that the style is given. For example, Xu et al. [2010] have worked exclusively with anisotropic part proportions. Some other works perform style analysis on shapes which belong to the same semantic category [Huang et al. 2013; Kalogerakis et al. 2012]. More recent attempts generally take the view that human style perception transcends shape content [Lun et al. 2015]. In an earlier work, Li et al. [2013] have handcrafted several rules as an attempt to characterize style features for 2D curves. Another line of works focus on analogy-based style transfer, e.g., Ma et al. [2014], where the goal is to determine what editing operations on a query shape $A'$ mimic the style change which would transfer a given shape $A$ to a given shape $B$. Lim et al. [2016] propose to analyze 3D shape styles with deep metric learning, based on multi-view rendering input.

*Supervised learning of style similarity.* Recent works on shape style analysis combine crowd-sourcing and metric learning to learn a generic style similarity. Most notably, Lun et al. [2015] and Liu et al. [2015a] both work with heterogeneous 3D shape collections and rely on crowd-sourced style ranking triplets to learn a style metric. Most recently, Lim et al. [2016] added deep learning to this framework. The key difference between our method and these works is that we learn to spatially locate style patches and they do not. Also, there are differences in what is learned and for what target applications. Our work learns what makes a piece of furniture Chinese/Country and a building Gothic/Greek and where the style regions are. The core problems we face are style classification/clustering and style patch extraction. In contrast, Lun et al. [2015] and Liu et al. [2015a] focused on style-driven shape retrieval, trying to learn a specialized shape similarity. Yet, our method can be customized to accomplish tasks Lun et al. [2015] and Liu et al. [2015a] were designed to do, making it more general.

On the technical front, several other differences exist: 1) our method can be both unsupervised and semi-supervised; 2) our method employs projective analysis and relies on different features; 3) our method adapts PSLF clustering to help us select and locate style feature patches, while both Liu et al. [2015a] and Lun et al. [2015] adapt metric learning to learn a global style metric.

*Supervised style localization.* The work by Hu et al. [2017] also learns to locate style patches, but it takes as input a collection of 3D shapes with expert-provided style clustering. In contrast, the input to our work is only such a shape collection, *without any style clustering*. In our semi-supervised version, the user can provide style labels or style ranking triplets, but only over a very small percentage of the data. Therefore, their work is exclusively on feature selection based on a given style clustering while we need to solve both clustering and feature extraction simultaneously.

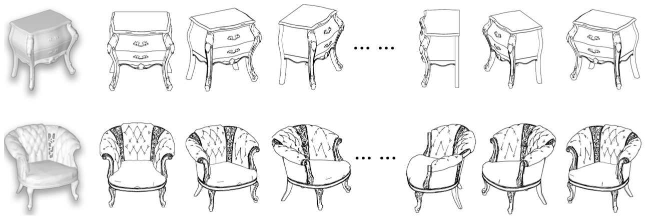

Fig. 3. Some multi-view feature lines from 3D shapes.

Another difference, a subtle one, is that Hu et al. [2017] aim to locate *style-defining* patches, i.e., all patches which together define a particular style, while our analysis seeks to find *style-discriminative* patches, i.e., those which can distinguish a style from the others. In their work, style-defining patches are simply a combination of style-discriminative patches. Technically, the feature learning schemes of the two methods are different and they operate on different features: we work with projected feature lines and Hu et al. [2017] employed similar features as Lun et al. [2015].

## 3 OVERVIEW

Given a heterogeneous collection of 3D shapes in several styles, we perform projective style analysis based on multi-view projected feature lines of each 3D shape. Our method contains three stages: patch sampling and pre-selection, view feature encoding based on patch convolution, and multi-view feature integration with partially shared latent factor (PSLF) learning. The PSLF interleaves style clustering and patch filtering in an unsupervised or semi-supervised fashion. Figure 2 illustrates an overview of our method.

*Multi-view feature lines.* For each 3D shape, we render it from the views of 12 virtual cameras located circularly around the shape in every 30 degrees. These cameras are elevated 30 degrees from the ground, pointing towards the centroid of the shape. For each view, we extract both suggestive contours [DeCarlo et al. 2003] and dihedral angle based feature lines [Gal et al. 2009], leading to an image of 200 × 200 size. While the former captures contours and creases of a smooth manifold, the latter is especially useful for extracting sharp feature lines from man-made shapes which can be potentially non-manifold. See Figure 3 for a few examples of multiple-view projected feature lines.

*Patch sampling and pre-selection.* We select a set of representative mid-level patches from all projected feature line images, that are used as the convolutional kernels in the feature encoding of the projections. Specifically, we first randomly sample a set of points on each shape. For each sample point and each view in which this point is visible, a patch is generated as the window centered at the projection of the point. We then perform $k$-means clustering to extract a set of representatives as the cluster centers (Figure 2(a)).

*Per-view feature encoding.* In the second step, a feature map is extracted for each feature line image via patch convolution, where the pre-selected mid-level patches serve as the convolution kernels. Such a convolutional feature encoding is known to be shift-invariant, since a convolution kernel may be activated at an arbitrary position in an image. This trait fits well to our problem since local style patches may appear in multiple spatial locations. To extract multi-scale features, we also perform patch convolution for sub-images, as shown in Figure 2(b). The final feature is a concatenation of per-region features after pooling, similar to [Bansal et al. 2015].

*Multi-view feature integration.* The core step of our algorithm is to fuse the features extracted from different views while clustering the shapes based on the fused features. This leads to a multi-view feature representation for each shape. We adopt the partially shared latent factor (PSLF) learning [Liu et al. 2015b] to implicitly separate the input multi-view features into parts which are shared by multiple views and those which are distinct to a specific view. The final multi-view feature is compact and comprehensive, encoding both shared and distinct information in different views.

*Unsupervised and semi-supervised style analysis.* Based on the clustering result, we re-select the representative mid-level patches to learn more and more discriminative ones with respect to the evolving style clusters. This will in turn update the feature encoding in the next iteration. Such cluster-and-select process iterates until the clusters and patches become stable. Our process can be performed unsupervised to cluster models. To impart human knowledge about shape styles into the analysis, we realize semi-supervised style clustering within the PSLF framework, achieving both meaningful style clustering and informative feature learning. Specifically, we present two semi-supervised clustering methods, accepting either user-prescribed style labels on a small portion of the shape collection or triplets of shapes indicating their style similarity (Figure 2(c)). Finally, we backproject the learned discriminative patches from projective space to surfaces of the input 3D shapes to extract and visualize the style patches over these 3D shapes.

## 4 SEMI-SUPERVISED PROJECTIVE STYLE CO-ANALYSIS

In this section, we describe our semi-supervised projective style co-analysis method in detail.

### 4.1 Patch sampling and pre-selection

We first sample 2D patches from the multi-view feature lines to bootstrap our style analysis. To ensure a uniform coverage of a shape surface, instead of sampling the patches in 2D projections, we sample 3D points on the shape surface and then use the 3D points as seeds to generate 2D patches through projection. 3D sampling also facilitates the back-projection of 2D patches into 3D for locating style patches. In practice, we sample 30 seed points on a 3D surface, project them onto the 2D views in which these points are visible. We then, for each projected 2D point, we extract a 2D square patch centered at this point. For a 200 × 200 image, we extract about 30 patches, where the patch size is chosen experimentally; see discussion in Section 5.

To select a set of representative patches for each view, we perform $k$-means clustering over all patches in that view in HOG feature space [Dalal and Triggs 2005]. The cluster centers are selected as the representative patches. In practice, we extract 50 representative patches for each view. Figure 4 demonstrates the sampled mid-level patches as well as the selected representatives.

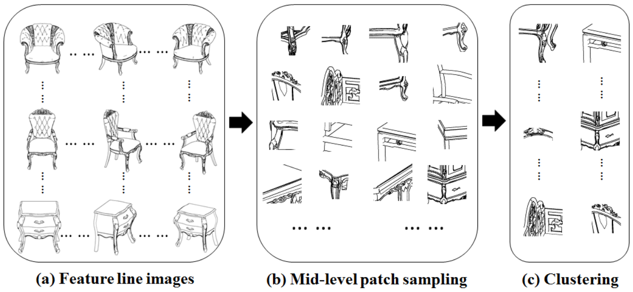

Fig. 4. Patch sampling and pre-selection. (a) Input feature line images. (b) Initial patch sampling. (c) Mid-level patches via $k$-means clustering.

### 4.2 Per-view feature encoding

Inspired by the work of Bansal et al. [2015], we perform feature encoding for each feature line image via patch convolution, using the representative mid-level patches as convolution filters. Note that we qualify these features as being "convolutional" since they are obtained by patch convolution. They are *not* learned using a convolutional neural network, but extracted by directly applying the convolution operations over input images. The main rationale behind our approach is that the characteristic mid-level patches, which were analyzed from a relevant image collection [Bansal et al. 2015], should be expected to encompass informative visual cues for an object class. Our iterative patch selection is conducted over relevant image collections obtained by clustering.

We take one mid-level patch as a convolution filter and use it to convolve the input image in a sliding-window fashion. Such a convolution operation is conducted in HOG feature space: both the input image and mid-level patches are represented by HOG feature maps. To compensate the global feature encoding of full image convolution, we also perform the above process over the sub-image obtained by dividing the original image into four parts. We then perform max pooling over the convolution activations over the spatial pyramid of two levels of resolution (Figure 5). Consequently, each convolution filter produces a 5-dimensional feature vector. The ultimate feature for a feature line image is constructed by concatenating the feature vectors of all convolution filters, leading to a $5K$-dimensional feature vector for $K$ mid-level patches.

### 4.3 Multi-view feature fusion

Having a feature vector for each view of a given shape, we perform multi-view feature fusing, to extract a new feature for the shape.

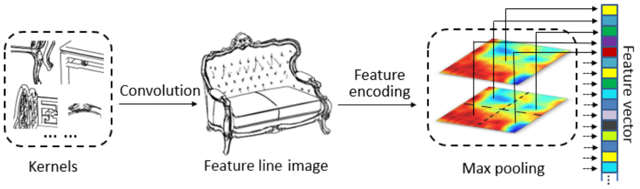

Fig. 5. Feature representation. For each feature line image, after patch convolution followed by max pooling, we obtain a new feature vector.

This feature instils the information from jointly analyzing a set of shapes. To achieve this, we adopt the partially shared latent factor (PSLF) framework [Liu et al. 2015b], which is a clustering method coupled with multi-view feature integration. Note that the "views" in the original work generally refer to different aspects, attributes, or observations of the data in question. In our case, the views are given by feature lines in multiple projections of 3D shapes.

PSLF employs non-negative matrix factorization (NMF) [Lee and Seung 1999] to learn a compact yet comprehensive partially shared latent representation. Given a collection of $N$ shapes with feature lines in $P$ views, the learning objective of PSLF is:

$$\text{min.} \quad \sum_{p=1}^{P} \pi^p ||\mathbf{X}^p - \mathbf{U}^p\mathbf{V}^p||_F^2 + \lambda||\Pi||_2^2, \tag{1}$$

$$\text{s.t.} \quad \mathbf{U}^1, \ldots, \mathbf{U}^p, \mathbf{V}^1, \ldots, \mathbf{V}^p, \Pi \geq 0, \sum_{p=1}^{P} \pi^p = 1,$$

where the *input feature* matrix $\mathbf{X}^p \in \mathbb{R}^{M \times N}$ contains the per-view feature of all $N$ shapes, for the $p$-th view, with each column corresponding to one shape. $M$ is the length of the feature vector of a given shape. $\mathbf{U}^p \in \mathbb{R}^{M \times K}$ is the basis matrix of view $p$, while $\mathbf{V}^p \in \mathbb{R}^{K \times N}$ is the matrix of $K$ latent factors, $K \ll M$, or the *fused feature* matrix. $\Pi = (\pi^1, \pi^2, \ldots, \pi^p)$ is the weights for different views. $\lambda$ controls the smoothness of $\Pi$. A large value for $\lambda$ leads to smoother view weights. Essentially, PSLF learns the fused feature matrix $\mathbf{V}$ and the basis matrix $\mathbf{U}$, while tuning the weights of different views, all in an unsupervised manner, by minimizing the reconstruction error with respect to the input features. The projection of the fused features over the basis leads to a clustering of input features. The PSLF factorization for a set of input features in one view is illustrated in Figure 6(a).

PSLF assumes that only parts of the latent factors are shared across all views and the other ones are separately embedded in individual views. Thus, the factor matrix of view $p$ is separated into two parts: $\mathbf{V}^p = \left[\mathbf{V}_s^p, \mathbf{V}_c\right]$, where $\mathbf{V}_s^p$ represents the specific information extracted from view $p$ and $\mathbf{V}_c$ the common shared by all views. The basis matrix is also divided into two parts: $\mathbf{U}^p = \left[\mathbf{U}_s^p, \mathbf{U}_c^p\right]$, with $\mathbf{U}_s^p$ being the specific part corresponding to the shared latent factors and $\mathbf{U}_c^p$ the common part.

An important parameter of PSLF is the proportion of common part: $\eta = (K_c/(K_s + K_c))$, where $K_c$ and $K_s$ are respectively the dimensions of the common and specific latent factors and $K_s + K_c = K$ holds. In this setting, when $\eta$ is larger, the role of consistency is more.

After performing the feature matrix factorization for all shapes being co-analyzed, we obtain the partially shared latent factor matrix $\mathbf{V} \in \mathbb{R}^{(K_s \times P + K_c) \times N}$, which contains the fused feature for each shape. The matrix $\mathbf{V}$ contains the unique feature part of each view $\mathbf{V}_s^1, \ldots, \mathbf{V}_s^p$ and the common part for all views $\mathbf{V}_c$. (i.e., $\mathbf{V} = \left[\mathbf{V}_s^1; \ldots; \mathbf{V}_s^p; \mathbf{V}_c\right]$), see Figure 6(b) for illustration.

An important feature of PSLF is that it is able to learn both consistent and complementary information from the data views. When applied to style analysis, our experiments (see Section 5) confirm that both types of information affect the learned shape styles, demonstrating the adaptability of PSLF to our problem.

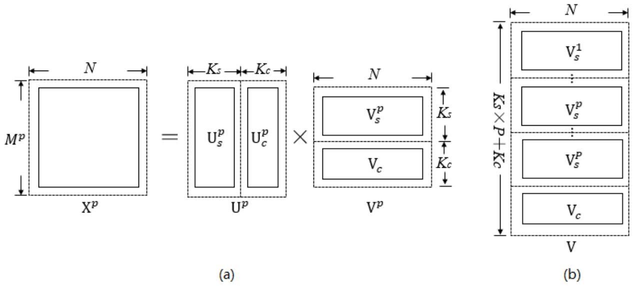

Fig. 6. Matrix factorization in the $p$-view and required partially shared latent factor matrix. (a) Input matrix $\mathbf{X}^p$ represents feature vector matrix of a shape collection in $p$-view. (b) Partially shared latent factor matrix $\mathbf{V}$.

## 4.4 Style clustering

PSLF was originally proposed for multi-view, semi-supervised clustering and feature learning [Liu et al. 2015b]. To accommodate both user-provided style ranking triplets and style labels to constrain semi-supervised analysis using the learning framework provided by PSLF, we must modify the original PSLF formulation.

*Label-constrained style clustering.* We modify the original objective function of PSLF to incorporate user-specified labels in constrained clustering. Suppose $\mathbf{V}_l \in \mathbb{R}^{(K_s \times P + K_c) \times N_l}$ is the feature vector matrix of $N_l$ number of shapes with labels. Assume, w.l.o.g., the first $N_l$ columns of $\mathbf{V}$ correspond to the labeled shapes. $\mathbf{Y} \in \mathbb{R}^{C \times N_l}$ is the corresponding label matrix, where $C$ is the number of style categories. The objective we optimize for is:

$$\min \sum_{p=1}^{P} \pi^p ||\mathbf{X}^p - \mathbf{U}^p\mathbf{V}^p||_F^2 + \lambda||\pi||_2^2 + \beta||\mathbf{V}_l - \mathbf{W}\mathbf{Y}||_F^2 + \gamma||\mathbf{W}||_{2,1} \tag{2}$$

$$\text{s.t.} \quad \mathbf{U}^1, \ldots, \mathbf{U}^p, \mathbf{V}^1, \ldots, \mathbf{V}^p, \Pi \geq 0, \sum_{p=1}^{P} \pi^p = 1$$

where $\pi^p$, $\mathbf{X}^p$, $\mathbf{U}^p$, $\mathbf{V}^p$ and $\lambda$ are defined the same as before. $\mathbf{W} \in \mathbb{R}^{(K_s \times P + K_c) \times C}$ is the basis matrix obtained for labeled shapes. $\beta > 0$ is a parameter for tuning the importance of user-specified labels. $\gamma$ controls the weight of $\ell_{2,1}$ regularization term.

In this new optimization, $\beta||\mathbf{V}_l - \mathbf{W}\mathbf{Y}||_F^2 + \gamma||\mathbf{W}||_{2,1}$ is the semi-supervised term, where the NMF of $\mathbf{V}_l$ produces the basis matrix $\mathbf{W}$ constrained with $\mathbf{Y}$. Note that this optimization not only predicts cluster labels for unlabeled shapes, but it also updates the fused feature matrix $\mathbf{V}$. Therefore, the output fused features in $\mathbf{V}$ have also incorporated the user constraints.

Since the basis $\mathbf{W}$ defines the cluster centers obtained from constraints, the fused feature matrix of the $(N - N_l)$ unlabeled shapes, $\mathbf{V}_u \in \mathbb{R}^{(K_s \times P + K_c) \times (N - N_l)}$ (so we have $\mathbf{V} = [\mathbf{V}_l, \mathbf{V}_u]$) can be defined by: $\mathbf{V}_u = \mathbf{W}\mathbf{Y}_u$. $\mathbf{Y}_u \in \mathbb{R}^{C \times (N - N_l)}$ is the label prediction matrix for the unlabeled shapes. These shapes can be assigned with the label corresponding to the largest probability.

*Triplet-constrained style clustering.* Exact style labels are sometimes hard to perceive even by humans. It is relatively easier to provide similarity-based supervision, e.g., shape A is style-wise closer to B than to C. This kind of user input has been extensively used in crowd sourcing [Liu et al. 2015a; Lun et al. 2015]. To incorporate triplet-based constraints into our clustering, we decompose each triplet into two pair-wise constraints, i.e., *must-link* and *cannot-link* between a pair of data points, which is a standard form of constraints used by semi-supervised learning [Chen et al. 2008].

Our method imposes such pair-wise constraints over the similarity matrix obtained by the unsupervised analysis of PSLF: $\mathbf{A} = \mathbf{V}^T\mathbf{V}$, where $\mathbf{V}$ is the partially shared latent factor matrix discussed in Section 4.3. Specifically, we modify the similarity matrix as follows:

$$\mathbf{A}' = \mathbf{A} + \mathbf{N}_\mathrm{m} - \mathbf{N}_\mathrm{c}. \tag{3}$$

where $\mathbf{N}_\mathrm{m} = \mathbf{I}((i,j) \in C_\mathrm{mustlink})$, $\mathbf{N}_\mathrm{c} = \mathbf{I}((i,j) \in C_\mathrm{cannotlink})$, with $i$ and $j$ as indices of a pair of shapes, $C_\mathrm{mustlink}$ and $C_\mathrm{cannotlink}$ collect the sets of shape pairs with must-link and cannot-link constraints, respectively, and $\mathbf{I}$ is an indicator matrix.

To conduct constrained clustering again using the PSLF framework, we perform another non-negative factorization over the modified similarity matrix: $\min ||\mathbf{A}' - \mathbf{Y}\mathbf{S}\mathbf{Y}^T||_F^2$, where $\mathbf{S} \in \mathbb{R}^{C\times C}$ contains cluster centers and $\mathbf{Y} \in \mathbb{R}^{N\times C}$ is cluster indicator.

*Unsupervised style clustering.* Having computed the fused feature matrix $\mathbf{V}$ in Section 4.3, performing unsupervised style clustering is straightforward. To do so, we utilize the self-tuning spectral clustering [Zelnik-Manor 2004]. This method produces the state-of-the-art clustering results while determines the number of clusters automatically. However, directly clustering the fused features may not generate the optimal results since the per-view features, computed with random patches, may not be the most relevant. To this end, we devise an iterative algorithm that interleaves clustering and cluster-guided patch re-selection, which will be discussed in the next subsection. In fact, such iterative cluster improvement can be also performed in semi-supervised analysis, through imposing the user constraints in every clustering.

### 4.5 Cluster-guided style patch selecting

The PSLF clustering has been so far based on the patches pre-selected by plain clustering without feature selection (Section 4.1). In fact, PSLF clustering couples feature selection and more importantly, incorporates the user constraints in the semi-supervised setting, making it both objectively informative and subjectively desirable. Therefore, it is preferable to use the PSLF clustering to guide a re-selection of mid-level patches, leading to more discriminative patches, specifically tuned for the unsupervised or semi-supervised tasks. The re-selected patches can in turn be used to update the PSLF clustering, via further purifying the clusters.

Based on the PSLF clustering results, we re-select discriminant mid-level patches for each view, to be those which are frequent only within one cluster [Xu et al. 2014]. For each style cluster $C_l$, we define the support weight of shape $i$ as $(\omega_{li})_{i=1}^n$, that measures the support of shape $i$ to any mid-level patch. A mid-level patch is determined as frequent if its weighted sum of support, denoted by discriminant score $\delta_{lj}$, is greater than a threshold $\delta_l^t$:

$$\mathcal{K}_l = \left\{ j | \delta_{lj} > \delta_l^t \right\}, \text{ where } \delta_{lj} = \left| \sum_{i=1}^{n} \omega_{li} \left( 2x_{ij} - 1 \right) \right|, \tag{4}$$

and $x_{ij}$ is an indicator function showing that shape $i$ supports patch $j$. If shape $i$ belongs to $C_l$, weights $\omega_{li}$ are positive, otherwise, they are negative. The discriminant score favors a patch that is frequent in cluster $C_l$ and penalizes its occurrence in other clusters. Therefore, the patches in $K_l$ are frequent mainly within cluster $C_l$ that is regarded as discriminant. Specifically, we define $\omega_{li} = x_{li}/C - 1/N_p$, where $x_{li} = I(i \in C_l)$ with $I(\cdot)$ being a 0-1 indicator function and $\delta_l^t = \mu N_p / C$ holds. $C$ is the number of clusters where $N_p$ is the total number of patches. We use $\mu = 0.07$ for all the datasets we have tested. The final set of patches takes the union of pre-cluster discriminant patches: $\mathcal{K} = \bigcup_{l=1}^C \mathcal{K}_l$.

After the mid-level patch re-selection, we repeat the process of per-view feature extraction, feature fusion for all 3D shapes and unsupervised or semi-supervised PSLF clustering. This cluster-and-select process iterates until the clusters and patches become stable. The final result comprises of purified style clusters, together with a set of style-characterizing mid-level patches, or *style patches*.

*Style patch extraction on shape surfaces.* One of our goals is to extract style patches on 3D shape surfaces based on the co-analyzed style patches in 2D. This can be done by backprojecting the 2D patches onto 3D surfaces. With the 3D patch sampling and projection scheme in Section 4.1, we can easily locate the surface region corresponding to a 2D patch. Note, however, the final style patches are selected from only a few 3D shapes, but not all. To locate the style patches on other shapes, we need to compare them against the patches sampled from those shapes, based on HOG features. Finally, we sort sampled areas on the 3D shape to find style patches according to the number of style patches back-projected to them. Figure 7 shows a few input shapes with the style patches highlighted in orange color. We also conducted a user study to verify the validity of our detected style patches in Section 5.

## 5 RESULTS, EVALUATION, AND APPLICATIONS

In this section, we first introduce our dataset and then show experimental results and evaluation. Our method is evaluated extensively over the largest collection of 3D models to date for style analysis, spanning six object categories, as shown in Table 1. We present a set of evaluations to examine the effect of our algorithmic choices. Our method is also compared with state-of-the-art approaches to shape style analysis and with other alternatives for input representation (projected feature lines vs. rendered images) and feature fusion (PSLF vs. max-pooling). Finally, we demonstrate several applications that exploit the spatial localization feature of our method. Additional results from our experiments, as well as the full user study data, can be found in supplementary material.

*Datasets.* The 3D models in our benchmark datasets have mostly been collected from the Internet (e.g. ShapeNet and Trimble 3D Warehouse) and previous published works. These datasets include a total of around 2,600 three-dimensional models arranged into six collections: Mixed Furniture 1, Mixed Furniture 2, Building,

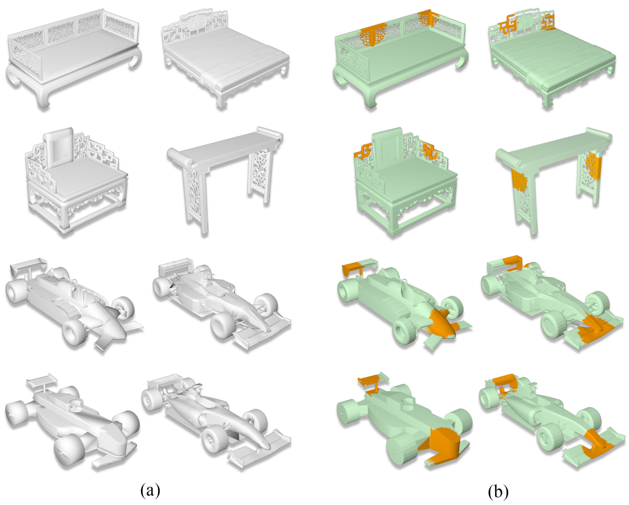

Fig. 7. Style patches on shape surfaces. (a) The input shapes. (b) The orange areas are the style patches.

| Shape collection | #Shapes | #Style classes |
|---|---|---|
| Mixed Furniture 1 | 120 | 4 |
| Mixed Furniture 2 | 400 | 5 |
| Building | 329 | 4 |
| Chair | 516 | 9 |
| Car | 1,050 | 6 |
| Vase | 194 | 5 |

Table 1. 3D shape collection for style analysis.

Chair, Car, and Vase. Each collection contains models with multiple styles, where for models sharing the same style, their geometry and structure can be quite different. In addition, even when all models fall under the same general category (e.g., chairs), their styles, geometries, and structures can be significantly different. Note that our method only assumes that the 3D objects have been upright oriented, but not necessarily consistently aligned.

*Style labels.* Our selection of style labels are based on common or professional knowledge accessible from publications and human experts. For example, furniture styles include “Simple Chinese”, “Noble Chinese”, “European”, “Country”, and “Modern”, according to their decorative styles [Morley 1999]. Buildings are labeled based on their geographic-temporal styles such as “Gothic”, “Greek”, “Byzantine”, and “Asian” [Lun et al. 2015]. In the final step, all the style labels for all object collections have been validated by four experts, who are professors from industrial engineering and architectural design. Table 1 shows the number of styles.

*Ground truth for style clustering.* We asked three of the experts to assign each 3D model in our dataset to one of the available style classes, based on the style labels obtained as described above, for all six object collections. After that, we asked the fourth expert to verify the style assignments. A few iterations were performed to arrive at the final labeling, which serves as the ground truth for style clustering of the models, per object collection. All the models and style labels can be found in the supplementary material. Note that these style clusterings are constructed only for evaluation; they are not used as training data for our method.

*Parameters.* There are five tunable parameters in the PSLF optimization. Specifically, the smoothness of different view weights is controlled by $\lambda$; non-negative parameter $\beta$ is to trade off between the objective of non-negative reconstruction and the $l_{2,1}$-norm regular item; the weight of the $l_{2,1}$-norm regular item is tuned by $\gamma$; common latent factor space shared between multiple views is controlled by $\eta$, where $0 < \eta < 1$, and finally, $\Pi = (\pi^1, \pi^2, \ldots, \pi^P)$ controls the weights of different views for all models. All the experiments have been conducted with a fixed parameter setting: $\beta \approx 0.05$, $\lambda \approx 20$, and $\gamma \approx 10$, and $\eta = 0.2$ in PSLF; view weights are set as $\Pi = (\frac{1}{P}, \frac{1}{P}, \ldots, \frac{1}{P})$ with $P$ views. We also examined the effect of different patch sizes on the purity of style clustering and found that a size of $48 \times 48$ generally leads to the best results overall; varying the patch sizes did not affect the purity more than 5%. Thus, we fixed the patch size to $48 \times 48$ in all of our experiments.

## 5.1 Style analysis results and evaluation

We first evaluate the performance of our method for style clustering and style patch localization. Since it is difficult to collect consistent ground truth data for style patches, we instead conduct a user study where human participants are asked to judge the results produced by our algorithm. For style clustering with style labels set up, we evaluate our results using the standard clustering *purity* measure. Let $C$ be the set of clusters from a clustering result for a dataset, and let $L$ be the set of ground truth clusters. For any cluster $c \in C$, its precision against a ground-truth cluster $l \in L$ is defined as, $P(c, l) = \frac{|c \cap l|}{|c|}$. The purity measure reflects an average of weighted precision for each cluster and is defined as:

$$\rho(C, L) = \sum_{c \in C} \frac{|C|}{|N|} max\left(P(c, l)\right) \tag{5}$$

Note that purity depends on its relative maximum precision on ground truth and therefore it can comprehensively reflect classification or clustering precision.

*Style patch localization.* To verify that the final patches returned by our style analysis indeed represent shape styles, we conduct a user study which compares our results to those annotated by human experts. We randomly selected 20 small sets of models from the six object collections, where each set consists of models belonging to the same style class based on our ground truth. For each set, we asked human experts to identify style-defining patches by painting over the shapes. We then conducted a user study where participants are provided with three types of patches (in color) for the same model set: randomly selected patches, expert-annotated patches, and patches returned by our style analysis algorithm. Figure 8 shows a sample query for one of the 20 model sets. Note that in the study, the three choices were randomly ordered in each query. Each subject is asked to choose which of the three choices would best reflect the style of the set of models shown. In the study, 20 model sets were

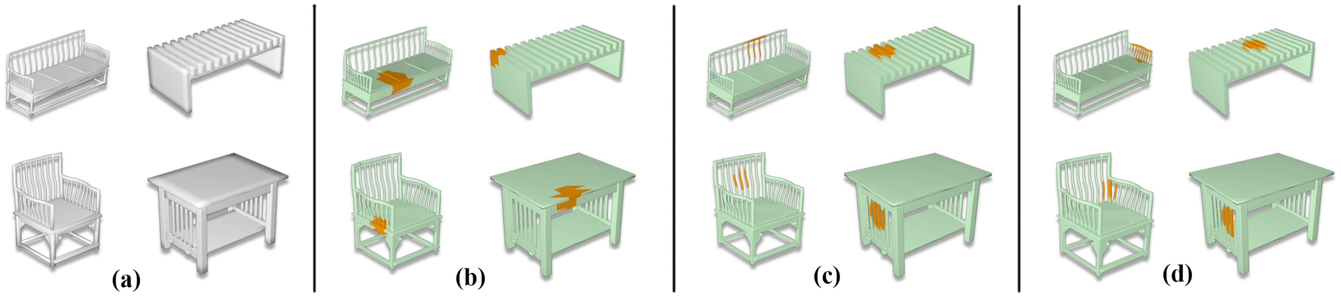

Fig. 8. A sample query for our user study for style patch localization. (a) Input shapes. (b) Randomly selected patches. (c) Expert-annotated style patches; (d) Patches returned by our style analysis method.

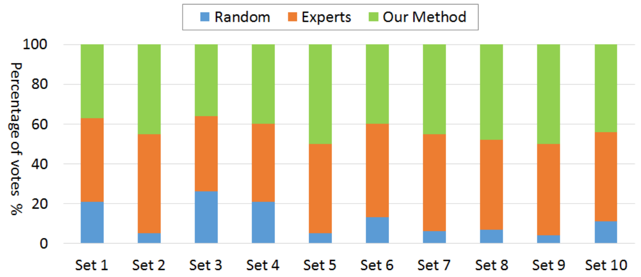

Fig. 9. Percentage of user votes for three types of patches: random (blue), expert-annotated (orange), and those returned by our method (green).

presented to 58 human participants with different backgrounds and no prior knowledge about our method.

A subset (10 out of 20 model sets) of results from this study are plotted in Figure 9, where the percentage of the user selection of each choice is shown. The full set of results can be found in the supplementary material. As indicated in Figure 9, participants' preference of our method is fairly close to that of the expert annotations.

*Semi-supervised analysis.* Our semi-supervised style analysis accepts two types of user inputs: style labels or style ranking triplets. Figure 10 shows how the style clustering performance, in terms of purity (see Equation 5), changes as we increase the percentage of user labels or the number of user-specified ranking triplets, respectively. All results were obtained with iterative PSLF-based clustering. As expected, clustering improves as user inputs increase. However, the improvement appears to level off when the label percentage passes 30% or the number of triplets reaches around 300. Note that 300 only represents an extremely small number of triplets out of the total number of triplets for an object collection.

*Parameter analysis.* We examine two key parameters, view count and $\eta$, which have the most significant impact on results among all the parameters and they need to be carefully selected.

We tested our style clustering with different number of views, ranging from 1 to 12 views. For a given number of view, we enumerate all different combinations of views, picked from the full set of 12 views. The final results for each view count were averaged over all different combinations. As shown in Figure 11, the clustering results gradually improves with increasing views. The typical "leveling out" points appear to be 10-12 views.

We have also analyzed the impact of parameter $\eta$ on classification accuracy for all datasets, in Figure 12. The parameter controls the proportion of common latent factor space shared across different views in PSLF. It can be observed that 0.2 gives the best results. This also verifies that PSLF is well-suited for our problem when both the consistency and complementarity of different views are exploited. Thus, we fix $\eta = 0.2$ in all experiments throughout the paper.

## 5.2 Comparative studies

We now provide several comparative studies to validate important design choices made in our method and to show how well our semi-supervision performs on the task of style similarity learning as compared to state-of-the-art methods.

*Style clustering: PSLF vs. PCA and CCA.* We compare our PSLF-based style clustering method with PCA [Jolliffe 2002] and Correlation Analysis-based approaches (CCA) [Chaudhuri et al. 2009]. Both PCA and CCA are dimensionality reduction techniques, just like PSLF. In the experiment, the reduced dimensionality of PCA is the same as the dimensionality of the partially shared latent representation in PSLF. CCA is a two-view method; we have executed the algorithm with each pair of two-view data and report results obtained using the best pair. The comparison is conducted using triplet constraints (100 triplets) to drive semi-supervised style clustering (see Section 4.4, which is based on fused features of each method for each data set. Figure 13 suggests that using PSLF to fuse the features tends to produce higher style clustering purities than PCA and CCA. We believe that the underlying reason is that PSLF can improve its fused features with the aid of triplet constraints while PCA and CCA cannot. In turn, the improvement on feature fusion is responsible for higher-accuracy clustering results. Additional results obtained by changing the amount of user inputs can be found in the supplementary material.

*Input: projected feature lines vs. rendered images.* To validate our choice of projected feature lines as input for analyzing element-type shape styles, we make a comparison to the use of two alternatives: rendered images and feature lines extracted from rendered images. For the first alternative, we render a 3D shape using the same local illumination setting as multi-view CNN (MVCNN) [Su et al. 2015],

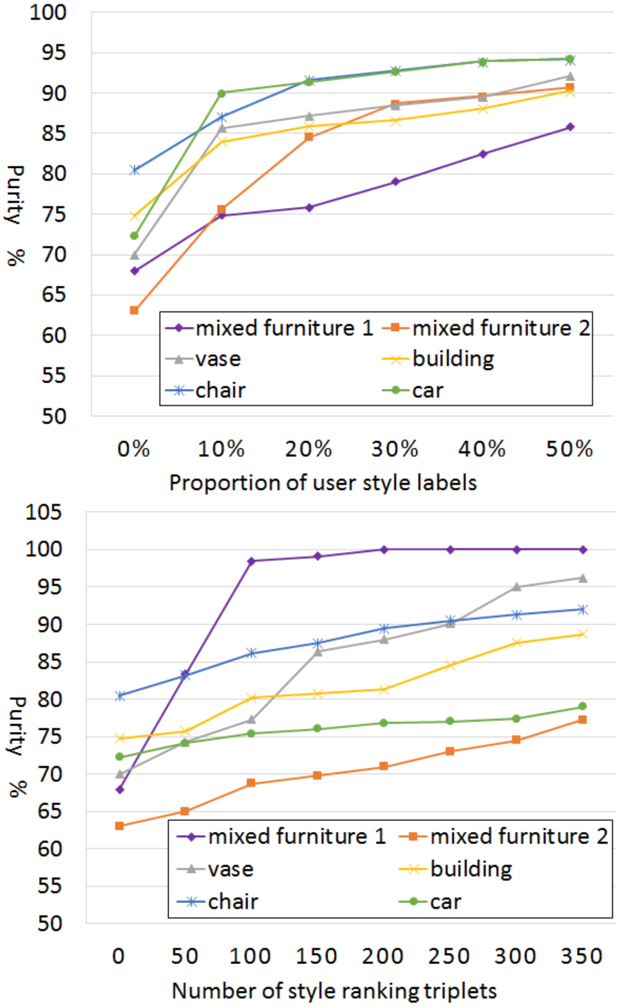

Fig. 10. Purity of unsupervised and semi-supervised PSLF clustering as user inputs increase. Top: user input as style labels, with unsupervised clustering corresponding to 0%. Bottom: user input as style ranking triplets, with unsupervised clustering corresponding to a count of 0.

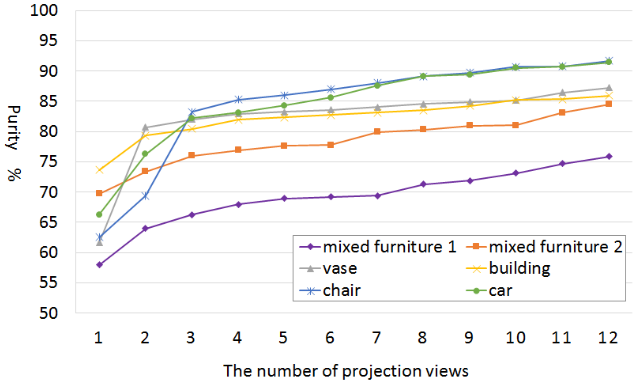

Fig. 11. Selection of the impact of shape projection views on our method.

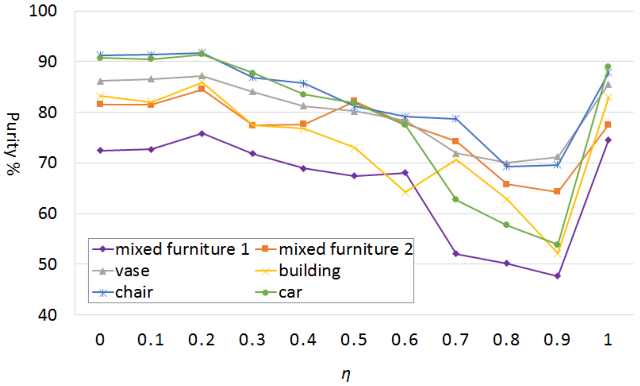

Fig. 12. Classification accuracy as functions of $\eta$. We conduct experiments for $\eta$ with 20% labels to illustrate its role in PSLF.

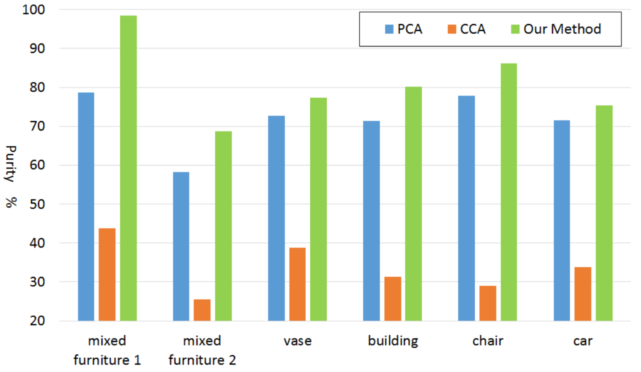

Fig. 13. Comparison on style clustering purity between our PSLF (green) and other feature fusion approaches including PCA (blue) and CCA (orange). User input consists of 100 ranking triplets.

obtaining a set of RGB images from the same views as those for our method. More advanced rendering options such as global illumination may increase the realism and clarity of the stylistic elements, but lighting, material, and texture have to be carefully set up. For the second alternative, we adopt the coherent line drawings (CLD) of Kang et al. [2007]. Figure 14 contrasts the three types of projective feature images for a building model.

We experimented on the three types of projective shape representations when they are plugged into our solution pipeline as input. We respect constraints given as 20% user-prescribed style labels (or 100 style ranking triplets; see results in supplementary material), while keeping all design choices (e.g., use of HOG features) and parameters in the algorithm the same. Results shown in Figure 15 (left) demonstrate consistent superiority of using projective feature lines for style clustering. Results under different constraint settings can be found in the supplemental material.

Next, we compare three types of projective shape representations when these inputs are fed to a deep neural network for features

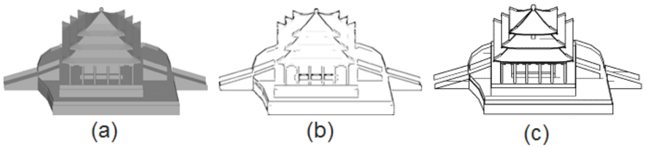

Fig. 14. Different projective shape representations. (a) Rendered images as in multi-view CNN. (b) CLD: feature lines extracted from rendered images. (c) Projected feature lines (detected in 3D object space) in our method.

learning before multi-view feature fusion. It is known that the neuron activations in the lower-level layers of CNNs are capable of capturing characteristic lines and corners in an image. For simplicity, we train a LeNet [Lecun et al. 1998] (5 convolutional layers plus 2 fully connected layers) with 20% labeled data from our datasets, to learn feature representations for each view separately. The per-view features are then fused with PSLF for final style clustering. For the two kinds of feature lines (our projected feature lines and CLD), the inputs to LeNet are represented in HOG space, while in the compared alternative, rendered images are used as input directly. Figure 15 (right) shows that projected feature lines again outperform other projective shape representations in deep learning setting.

The comparison results suggest that since our feature lines were detected in *3D object space*, they tend to be more reliable than their 2D counterparts; such features may not be fully recovered from rendered images by a base-line neural network. Moreover, our feature analysis is geometry-based which can effectively avoid rendering artifacts such as those resulting from poor lighting conditions; this may also contribute to their better performance as shown in Figure 15. Still, we should caution that these comparisons have only been made to simple off-the-shelf and base-line solutions in terms of rendering options and neural network architectures.

*Feature fusion: max pooling vs. PSLF.* A key component of multi-view style analysis is the integration or fusion of features extracted for multiple view channels. A commonly used fusion scheme is max pooling over the multi-channel features [Su et al. 2015]. However, such a simplistic operator is oblivious to the consistency or complementarity among the multiple channels which are both essential to effective feature fusion. To verify this, we train a VGGNet-16 [Simonyan and Zisserman 2014] to extract per-view features from projected feature lines, and then apply max pooling or PSLF as two options for feature fusion. The dimensionality of the fused feature is 512 for max pooling and 50 for PSLF. For the max pooling option, we follow the feature fusion either with CNMF [Liu and Wu 2010] or fully connected layers similar to MVCNN [Su et al. 2015] to perform style clustering, under the constraint of 20% data with known labels, same as PSLF. The comparison results in Figure 16 show that PSLF outperforms max pooling, whether followed by CNMF or MVCNN, over all the object categories. This verifies that PSLF, as a clustering method with a carefully designed feature fusion scheme, is especially suited for multi-view style analysis.

*Learning style similarity.* State-of-the-art methods for learning style similarities from style ranking triplets include the recent works of Lun et al. [2015], Liu et al. [2015a], and Lim et al. [2016]. Since our semi-supervised PSLF learning also accommodates style ranking triplets as user input (see Section 4.4), these methods and our method can be compared for the task of *predicting style similarity rankings* using the learned similarity distance. Our comparisons were conducted on datasets from the three previous works, respectively. As well, the set of style ranking triplets were also reused from their works. We split the set of triplets into a training set for learning and a testing set. Prediction accuracy is measured on how accurate the learned similarity distance would predict the similarity relations among the three data entities in a testing triplet.

| Scene category | [Liu et al. 2015a] | Ours | %triplets |
|---|---|---|---|
| living room | 73% | **85%** | 10% |
| dining room | 72% | **74%** | 30% |

Table 2. Comparing style ranking prediction accuracy with [Liu et al. 2015a] on their scene datasets. The last column shows % of style ranking triplets employed by our method for training as opposed to [Liu et al. 2015a].

| Category | [Lim et al. 2016] | Ours |
|---|---|---|
| building | 88.8% | **96.1%** |
| coffee set | 89.2% | **93.24%** |
| column | 98% | **100%** |
| cutlery | 81.2% | **96.39%** |
| dish | **90.8 %** | 82 % |
| furniture | 86.2 % | **97.47%** |
| lamp | 88.5 % | **100 %** |

Table 3. Comparing style ranking prediction accuracy with [Lim et al. 2016], over seven datasets from their work.

Figure 17 shows a comparison to Lun et al. [2015] on four of their seven object categories, where the number of triplets used for training varies from 50 to 550. The remaining three categories had much fewer available triplets and the corresponding comparison results can be found in the supplementary material. Our method leads to higher accuracies in all cases with only two exceptions: the lamp and dish datasets. Our performance on the dish set is below that of Lun et al. [2015] and we believe this is due to the fact that the dish shapes are mostly smooth and they lack line-type features, while our method relies only on features from projected feature lines. On the other hand, Lun et al. [2015] employs a large set of features, including projective ones. For the lamp set, since it contains a large number of models and variations, it is conceivable that by relying on a more limited feature set, our semi-supervised learning would require more training to reach a performance plateau.

We also compare style ranking prediction accuracy with Liu et al. [2015a] using the two scene datasets tested in their paper; see Table 2. We do not use all the ranking triplets as in their work. Instead, we randomly sample a subset. As we can see, with a relatively small percentage of triplets employed for training, our method is able to achieve comparable or better prediction accuracy.

Lim et al. [2016] propose to identify 3D shape styles based on deep metric learning. Their evaluation was performed on the same datasets as Lun et al. [2015]. Table 3 reports a comparison to Lim

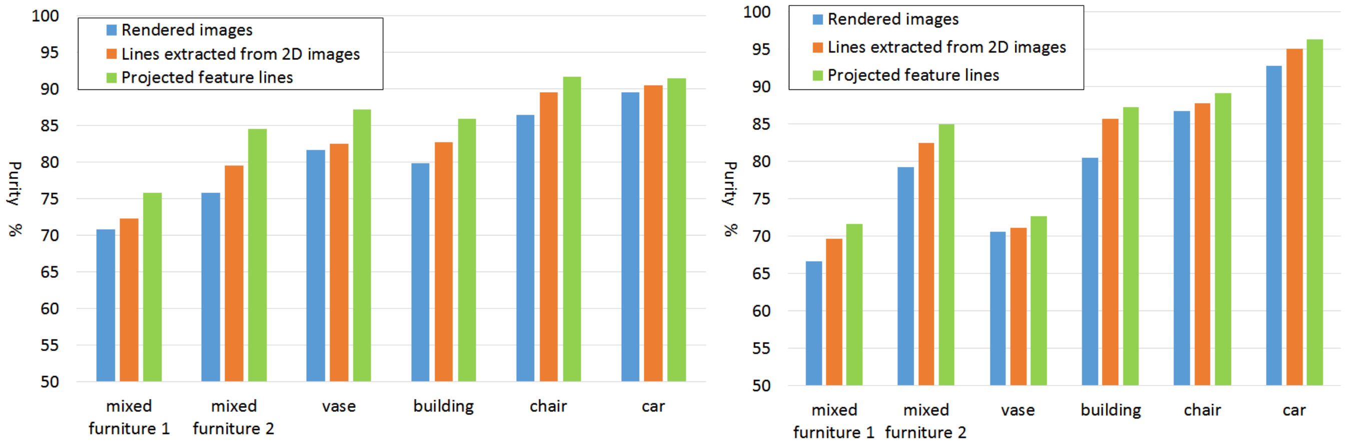

Fig. 15. Comparing style clustering purity over different projection methods (rendered images vs. lines extracted from 2D images vs. projected feature lines) and different feature extraction schemes (left: using feature extraction in our method; right: using deep feature extraction via LeNet).

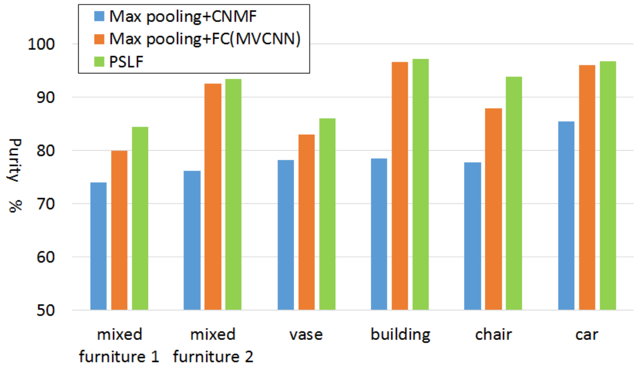

Fig. 16. Comparing three feature fusion schemes on clustering purity. In all three cases, max pooling with CNMF (blue), max pooling with fully connected layers as Multi-View CNN (orange), and PSLF (green), the same line features extracted from VGGNet-16 were employed.

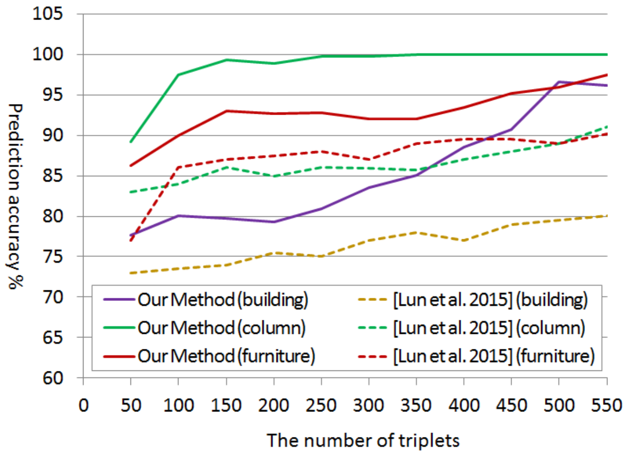

Fig. 17. A comparison on style ranking prediction accuracy with [Lun et al. 2015], over four of their object categories.

et al. [2016] on their seven object categories with the same triplets from [Lun et al. 2015] for training. Specifically, 550 triplets were used for the building, column, furniture and lamp set, 150 for the coffee set, and 100 for the cutlery and dish set. These models were trained with 3D shapes, except [Lim et al. 2016] which also uses photos. As can be seen from Table 3, our method outperforms the deep learning alternative from Lim et al. [2016] for all object categories except for the dish set. Again, we believe that this is due to the fact that most of the dish models do not possess rich line features. At the same time, it is demonstrated that projected feature lines and HOG-space feature encoding are especially suited to study decorative styles for the other object categories such as furniture and buildings. The results shown in this experiment, however, should not be interpreted as an indication of general superiority of our feature representation and learning scheme over deep learning based alternatives.

*Style classification.* We compare style classification results obtained using our method with those from the recent supervised method of Hu et al. [2017], over the five datasets used in their work: 1) Furniture (618 models in 4 styles); 2) Furniture legs, (84 models and 3 styles); 3) Buildings (89 models in 5 styles); 4) Cars (85 models in 5 styles); 5) Drinking vessels (84 models in 3 styles).

Similar to their work, we run a classification experiment with a 10-fold cross-validation. However, there is one difference. They learned the sets of style-defining patches to represent shapes and train *k*NN classifiers for each style on 9 folds. We, instead, used 9 folds as style labels in our method. We evaluate the classification accuracy on the remaining fold for both methods. Finally, we compute the average accuracy for the 10 folds for all style labels in each set, based on the ground-truth labels provided in Hu et al. [2017]; the results are shown in Figure 18. We can observe that our method outperforms theirs for all object categories except for buildings.

We attribute the general improvements over Hu et al. [2017] on this task to two possible factors. First, projected feature lines are

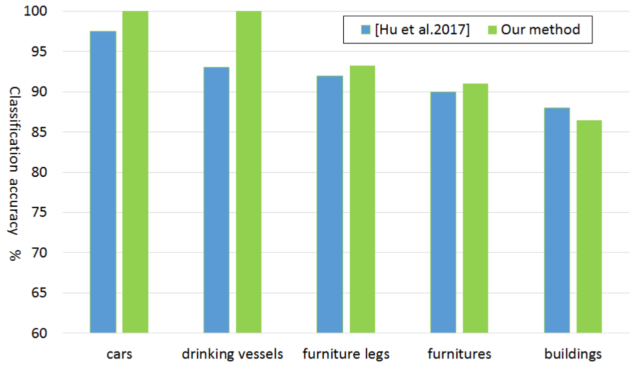

Fig. 18. Comparison on style classification between Hu et al. [2017] (blue) and our method (green) over datasets from their work.

more suited to the kind of decorative styles and man-made shapes in our experiments. Shape features considered in Hu et al. [2017] may be severely degraded due to low mesh resolution, tessellation quality, and other geometric imperfections, while feature lines are more robust against these issues. Second, their method relies on mid-level 3D patches, which only capture local geometry information. In contrast, our method captures both local and global information owing to feature encoding via patch convolution which is known to be more apt at hierarchical feature learning.

*Style patch localization.* Finally, we compare our method to Hu et al. [2017] on style patch localization, over their datasets, the same ones tested above for style classification. Since their method is supervised with given style clusters, we only compare the feature selection components of the two methods. That is, our method would take the same style clusters as input as for their method.

Comparing patch localization is not straightforward since the located patches for each shape is not unique for either method. To simplify matters, we carry out the comparison on 19 representative style patches obtained by Hu et al. [2017]. These 19 patches come from 19 shapes (one patch per shape) encompassing all five object categories; they were selected and shown in Figure 11 of the paper. Hu et al. [2017] qualified them as style patches that "capture *distinctive characteristics* of the styles." For each of the 19 shapes, the representative style patch from our method is chosen as the one which is deemed to be a style patch over the most views.

Figure 19 shows a visual comparison on 10 out of the 19 shapes. We can observe that the style patches detected bear some similarities in general, but our style patches tend to be more feature-rich. The rest of the results can be found in the supplementary material.

For a quantitative comparison, we conducted a user study to evaluate the representative style patches found by the two methods on the 19 shapes. The study consists of 19 queries, one per shape. For each query, subjects were provided with the two style patches, marked as *A* and *B* and shaded in the same color, for the same shape. Then the subjects were asked to choose one of four possible answers in regards to the style patches: 1) Patch *A* represents the shape style better; 2) Patch *B* represents the shape style better; 3) Patches *A* and *B* both represent the shape style well; 4) Neither set of patches represents the shape style. For each test shape, our result and theirs are randomly ordered.

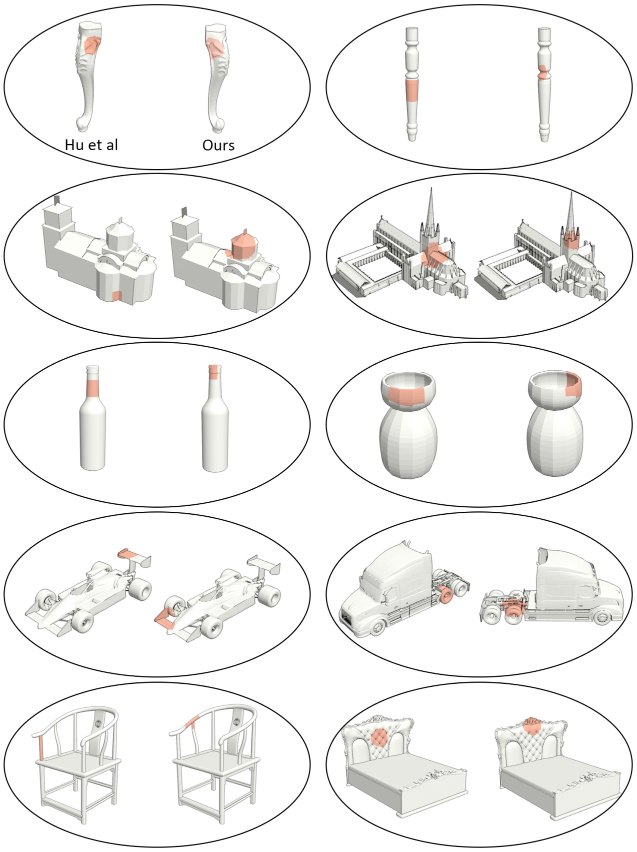

Fig. 19. Visual comparison of style patches located by our method (right one in each pair) vs. those found by [Hu et al. 2017] (left one in each pair).

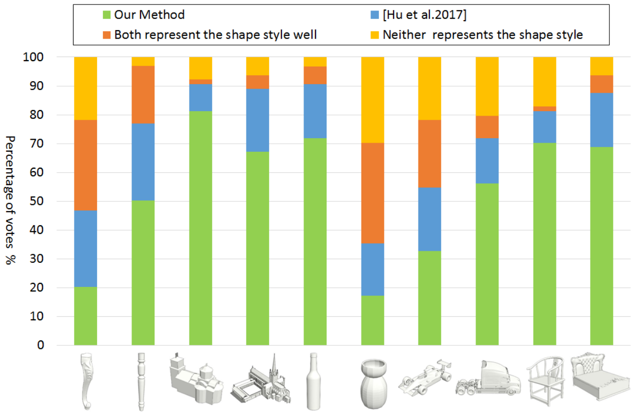

Fig. 20. Subset of results from the user study on style patch localization, comparing our method to that of [Hu et al. 2017]. We plot the percentage of user choices over the four different answers for 10 shapes.

A total of 58 subjects participated in the study; these subjects are all students from various disciplines including computer science, architecture, arts, and information management. Results on a subset of (10 out of 19) shapes are shown in Figure 20, where the percentages of user selections for different answers are plotted; the remaining results are available from the supplementary material. It is quite evident that style patches extracted by our method were generally more favored by the human subjects.

## 5.3 Applications

In this section, we present several applications of our style analysis method which are enabled by its ability to spatially locate style patches. Such a list of applications is far from exhustive though. Since our method can learn style similarity as previous works by Lun et al. [2015], Liu et al. [2015a], and Lim et al. [2016], while taking on the same kinds of input, it can support any application demonstrated in these works which utilize style similarities.

*Style-aware mesh simplification.* Spatial localization of shape styles allows a simple scheme to be developed for style-aware mesh simplification. The main extra step, after obtaining the style patches from our current method, is to extend or extrapolate the few style patch *samples* returned to entire style regions over the mesh surface. Then, we can apply a constrained version of quadric-based mesh simplification [Garland and Heckbert 1997] while keeping the style regions in tact. Figure 21 shows some results, where the number of triangles (after 70% reduction) are the same for the simplified models with and without preserving styles. Apparently, style-aware simplified models better resemble original models style-wise and have larger triangles over flat areas highlighted by red boxes.

For the patch-to-region extension, all we need to do is to compare the initially sampled patches (see Section 4.1) with the detected style patches and then mark all those with HOG-space similarity above a threshold as stylistic. All the style patches are finally back-projected to the 3D shape to aggregate into style regions over the shape. To improve style region boundaries, we re-sample three times as many initial patches as before to increase the resolution.

*Style-aware view selection.* Another application for spatial localization of style patches is style-aware view selection, where views that are deemed to be most *style-revealing* for a 3D object are identified. After style patches are obtained as discussed in Section 4.5, we use them to first detect similar (initially sampled) patches in HOG space for each considered view. We then select the view that has the maximum number of patches that are sufficiently similar to the style patches. Figure 22 shows some of the best object views obtained this way, as opposed to other views.

We compare our style-aware best view selection with human judgment. For a test 3D model, we let 58 human subjects select, among the 12 views employed in our multi-view style analysis, which one provides the "best view" for the model. In Figure 23, we show the percentage of subject votes for each view, for five selected 3D models. The results of this study reveal that our simple best view selection based on extracted decorative shape styles tends to obtain the same or close views as the human subjects would.

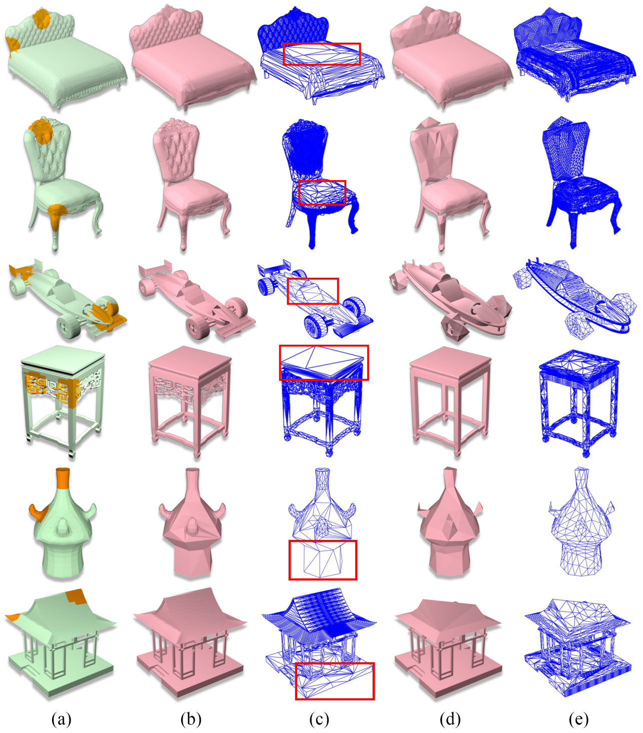

Fig. 21. Style-aware mesh simplification. (a) Original meshes with style patches. (b-c) Shaded and wireframe versions of simplified models with style preservation via constrained quadric-based decimation; red boxes highlight significant triangle reduction near non-style areas. (d-e) Simplified models without style preservation, via unconstrained decimation.

*Sketch-driven identification of 3D architectural styles.* Visual documentation of styles, particularly those of architectural models, are typically in the form of 2D sketches; they can be commonly found in professional guide or style books. Figure 24 shows 16 out of 160 building sketches we scanned from professional guidebooks, in four styles: Asian, Byzantine, Gothic, and Greece. More style sketches can be found in the supplemental material. Architectural models in the real world are three-dimensional, hence, to recognize their styles, a joint 2D-3D analysis is necessary to enable such recognition tasks based on 2D sketches. Our style analysis method based on projected feature lines seems ideally suited in this setting. Beyond style recognition, our ability to spatially locate style patches allows sketched styles from guidebooks to be identified on 3D models.

We scanned 160 building sketches, in four styles as shown in Figure 24, from professional architecture books. Each sketch image is re-sized to the same resolution as projected line images in our style analysis method, after the depicted buildings are properly scaled to fit the sketch image. Given a test 3D building model, we project it into 12 views to obtain the projected feature line images as before. Then we compare each of the 12 line images with the stored building sketches and let the closest sketch vote for its corresponding style. The recognized style of the 3D building is the one that received the most votes. When comparing two line/sketch images, we employ

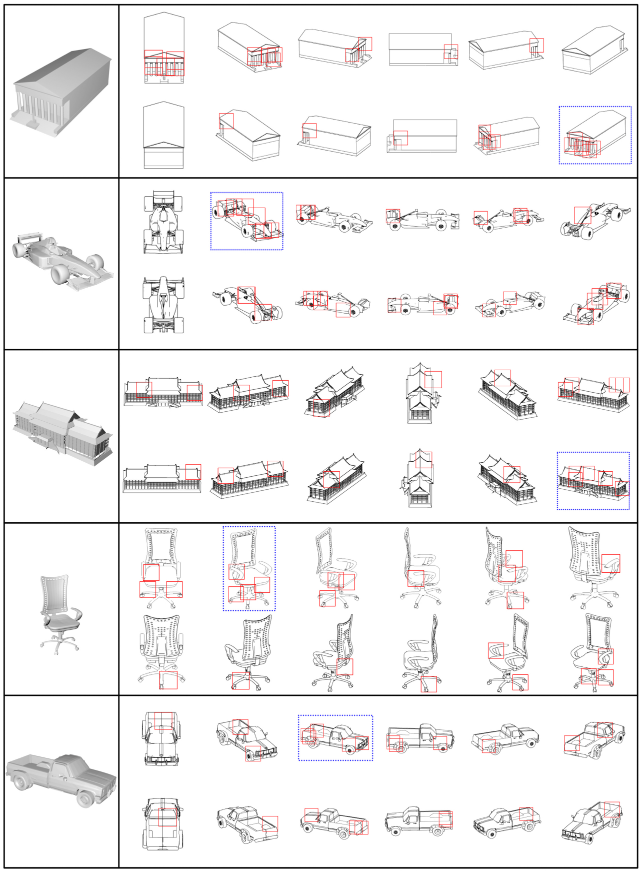

Fig. 22. Style-aware view selection. For each 3D model, we show it, as projected feature lines, in the 12 views employed by our multi-view style analysis method. Detected style patches and those deemed to be similar to them are shown in red boxes. The best view, one with the most red boxes, is highlighted by a blue box, and also shown in the left column.

the same HOG space patch convolution as before (see Figure 5) and use Euclidean distances between the resulting feature vectors. The convolutional kernels are given by representative patches obtained from the 160 building sketches via mid-level patch extraction and $k$-means clustering ($k = 10$) as before. Localization of style patches on the 3D building model can be achieved by comparing the representative style patches to patches sampled from the 3D model via HOG-space patch convolution and back-projection.

We use the 329 building models in our dataset to test the above simplistic style recognition and patch localization scheme. Some results are shown in Figure 25. The style recognition accuracy is 71.4%, which shows promise but is slightly lower than the 74.77% obtained by our unsupervised style clustering with PSLF feature fusion. There are a few possible reasons: a) there is no multi-view feature fusion since each building sketch in the training set has only one view; b) the building sketches may have different views than our projections; c) the building sketches contain richer set of feature lines than the projected feature lines extracted in our method.

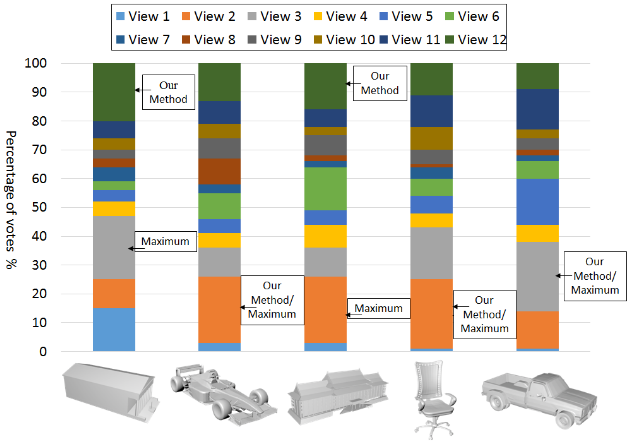

Fig. 23. Results of the user study on best view selection. Percentage of user votes for each view is shown. Views selected as best by most human subjects and views selected by our style-aware selection scheme are both indicated. At the bottom, we show best views selected by human subjects, which can be contrasted with the best views found by our method in Figure 22.

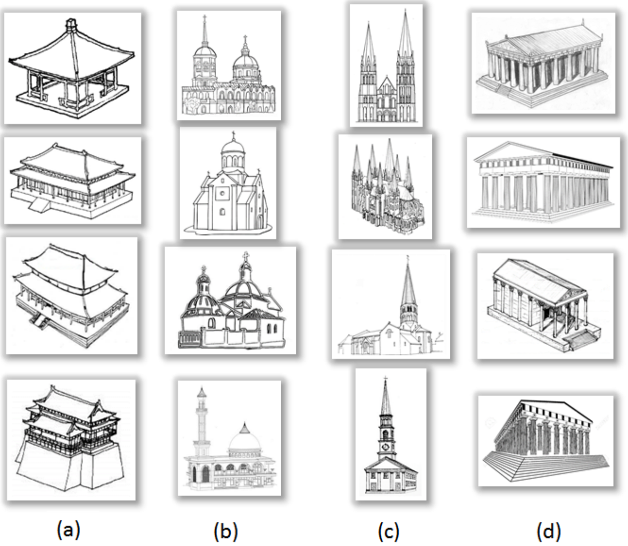

Fig. 24. Some 2D building sketches scanned from professional guidebooks depicting four styles: (a) Asian; (b) Byzantine; (c) Gothic; (d) Greece.

## 6 DISCUSSION, LIMITATION, AND FUTURE WORK

We propose what we believe to be the first semi-supervised method for analyzing and locating decorative style patches on 3D shapes. Our technique utilizes projective feature lines and multi-view feature encoding for style clustering and patch extraction. Our semi-supervision is able to take on the same kind of input, namely, crowd-sourced style ranking triplets, as recent works on style metric learning [Liu et al. 2015a; Lun et al. 2015]. We have shown that comparable accuracy on style similarity tests can be attained by our method with less user input than these recent works. As well, we

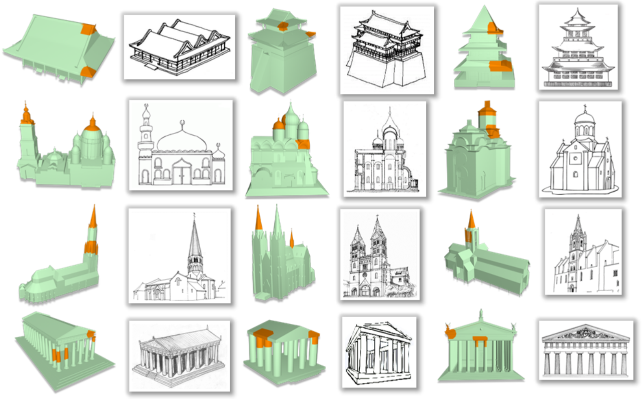

Fig. 25. Some examples of recognition and spatial localization of architectural styles on 3D building models. From the top to the bottom row: Asian, Byzantine, Gothic and Greece. For each example, the matched 2D style sketch is shown to the right. Style patches located on the 3D shapes are shown in orange color.

have demonstrated improvements on style classification and style patch localization over the most recent work by Hu et al. [2017].

One of the main limitations of our current style analysis is that by design, it can only extract stylistic elements that are visually apparent as feature lines and localized to the patch level. These do not include stylistic arrangement of patterns such as those involving symmetries and repetitions. The main difficulty is that these more global and structural styles may not be fully visible in projected images. Technically, our final style patch extraction hinges on the initial pre-selection of representative feature patches. In addition, for feature-lacking shapes such as a smooth dish or spoon, projected feature lines cannot be expected to reveal sufficient stylistic elements. Without sufficient features, our method may lead to undesirable constrained clustering results.

Our method has been shown to work quite effectively with relatively small amount of style-labeled data. In such a setting and over a limited set of style and object classes, our method is able to obtain comparable or better accuracy on the style similarity test against the deep metric learning method of Lim et al. [2016]. We believe that this is mainly attributed to the multi-view feature fusion power of PSLF. On the other hand, limited results from such a small-scale test should by no means imply general superiority of our method over deep learning based approaches. When large amounts of training data become available, deep learning may outperform PSLF. As a key feature of our method, it supports both unsupervised and semi-supervised style analysis and can potentially facilitate the production of large-scale style-labeled 3D data to serve supervised deep learning based style analyses. Combining the multi-view PSLF and deep feature learning is also an interesting possibility.

The focus of our current work is style analysis over 3D shapes, specifically, clustering and spatial localization of decorative shape styles. At the representation level, our choice of projected feature lines is mainly dictated by the task at end, i.e., style analysis, as such features are best suited to reveal decorative styles. We also emphasize the advantage of detecting the line features in the object space, rather than the image space after rendering. That being said, there has been significant progress on representation and feature learning using deep neural networks with modern architectures such as ResNet [He et al. 2016]. With sufficient training data, the deep learning alternatives may be expected to outperform traditional HOG-space feature encoding, with or without patch convolution, for generic analysis tasks. An advantage of our PSLF-based solution to representation learning is *interpretability*, especially in the context of view-based 3D shape style analysis. PSLF learns both shared and complementary features among different views, and integrate them into a compact, view-based representation for 3D shapes.

Beyond style-aware shape simplification and view selection, we believe that there is more to explore on the application front for style patch localization. The ability to identify these patches spatially allows them to be directly manipulated and applied. For example, style patches found on the legs of one chair can be transplanted [Takayama et al. 2011] to the legs of another piece of furniture. Also, the style patches can be isolated and encoded as details over base patches to form a library of style templates. These are both *style transfer* or style-aware shape or part synthesis tasks, where aesthetics, semantics, and mechanical stability of the new parts may all need to be accounted for. In addition, once we have a set of style patches in possession, we can identify or retrieve more patches from novel 3D shapes simply based on geometric similarities, either over shape surfaces or in projective space. Finally, it would be interesting to extend our style analysis to 3D scenes.

## ACKNOWLEDGEMENTS

We thank the anonymous reviewers for their valuable comments. This work was supported in part by NSFC (61572507, 61532003, 61622212) for Kai Xu, NSERC (611370) and a gift grant from Adobe Research for Hao Zhang, and NSERC PDF for Ali Mahdavi-Amiri. We are grateful to Wentao Wu for help and discussion.

## REFERENCES

Aayush Bansal, Abhinav Shrivastava, Carl Doersch, and Abhinav Gupta. 2015. Mid-level Elements for Object Detection. *arXiv preprint arXiv:1504.07284* (2015).

Kamalika Chaudhuri, Sham M Kakade, Karen Livescu, and Karthik Sridharan. 2009. Multi-view clustering via canonical correlation analysis. In *Proc. Int. Conf. on Machine Learning*. 129–136.

D.Y. Chen, X.P. Tian, Y.T. Shen, and M. Ouhyoung. 2003. On visual similarity based 3D model retrieval. *Computer Graphics Forum* 22, 3 (2003), 223–232.

Yanhua Chen, Manjeet Rege, Ming Dong, and Jing Hua. 2008. Non-negative matrix factorization for semi-supervised data clustering. *Knowledge and Information Systems* 17, 3 (2008), 355–379.

Navneet Dalal and Bill Triggs. 2005. Histograms of oriented gradients for human detection. In *Proc. CVPR*, Vol. 1. IEEE, 886–893.

Doug DeCarlo, Adam Finkelstein, Szymon Rusinkiewicz, and Anthony Santella. 2003. Suggestive contours for conveying shape. *ACM Trans. on Graph. (SIGGRAPH)* 22, 3 (2003), 848–855.

Carl Doersch, Abhinav Gupta, and Alexei A Efros. 2013. Mid-level visual element discovery as discriminative mode seeking. In *Proc. NIPS*. 494–502.

Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. 2013. Decaf: A deep convolutional activation feature for generic visual recognition. *arXiv preprint arXiv:1310.1531* (2013).

Ran Gal, Olga Sorkine, Niloy J Mitra, and Daniel Cohen-Or. 2009. iWIRES: an analyze-and-edit approach to shape manipulation. *ACM Trans. on Graph. (SIGGRAPH)* 28, 3 (2009), 33.

Elena Garces, Aseem Agarwala, Diego Gutierrez, and Aaron Hertzmann. 2014. A Similarity Measure for Illustration Style. *ACM Trans. on Graph.* 33, 4 (2014), 93:1–9.

Michael Garland and Paul S. Heckbert. 1997. Surface simplification using quadric error metrics. In *Conference on Computer Graphics and Interactive Techniques*. 209–216.

Yunchao Gong, Liwei Wang, Ruiqi Guo, and Svetlana Lazebnik. 2014. Multi-scale orderless pooling of deep convolutional activation features. In *Computer Vision–ECCV 2014*. Springer, 392–407.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *Proc. CVPR*.

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. on Graph.* (2017).

Qi-Xing Huang, Hao Su, and Leonidas Guibas. 2013. Fine-grained semi-supervised labeling of large shape collections. *ACM Trans. on Graph.* 32, 6 (2013), 190.

Ian Jolliffe. 2002. *Principal component analysis*. Wiley Online Library.

Evangelos Kalogerakis, Siddhartha Chaudhuri, Daphne Koller, and Vladlen Koltun. 2012. A probabilistic model for component-based shape synthesis. *ACM Trans. on Graph.* 31, 4 (2012), 55.

Henry Kang, Seungyong Lee, and Charles K. Chui. 2007. Coherent Line Drawing. In *Proceedings of the 5th International Symposium on Non-photorealistic Animation and Rendering (NPAR '07)*. 43–50.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324.

Daniel Lee and Sebastian Seung. 1999. Learning the parts of objects by non-negative matrix factorization. *Nature* 401, 6755 (1999), 788–791.

Yong Jae Lee, Alexei Efros, and Martial Hebert. 2013. Style-aware mid-level representation for discovering visual connections in space and time. In *Proc. ICCV*. 1857–1864.

Honghua Li, Hao Zhang, Yanzhen Wang, Junjie Cao, Ariel Shamir, and Daniel Cohen-Or. 2013. Curve style analysis in a set of shapes. *Computer Graphics Forum* 32, 6 (2013), 77–88.

Yao Li, Lingqiao Liu, Chunhua Shen, and Van Den Hengel Anton. 2015. Mid-level deep pattern mining. In *Proc. CVPR*. 971–980.

Isaak Lim, Anne Gehre, and Leif Kobbelt. 2016. Identifying Style of 3D Shapes using Deep Metric Learning. *Computer Graphics Forum* 5 (2016).

Haifeng Liu and Zhaohui Wu. 2010. Non-negative matrix factorization with constraints. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*. 506–511.

Jing Liu, Yu Jiang, Zechao Li, Zhi Hua Zhou, and Hanqing Lu. 2015b. Partially Shared Latent Factor Learning With Multiview Data. *IEEE Trans. on Neural Networks & Learning Systems* 26 (2015), 1233–1246.

Tianqiang Liu, Aaron Hertzmann, Wilmot Li, and Thomas Funkhouser. 2015a. Style Compatibility for 3D Furniture Models. *ACM Trans. on Graph.* 34, 4 (2015), 85:1–85:9.

Zhaoliang Lun, Evangelos Kalogerakis, and Alla Sheffer. 2015. Elements of style: learning perceptual shape style similarity. *ACM Trans. on Graph.* 34, 4 (2015), 84:1–84:14.

Chongyang Ma, Haibin Huang, Alla Sheffer, Evangelos Kalogerakis, and Rui Wang. 2014. Analogy-driven 3D style transfer. *Computer Graphics Forum* 33, 2 (2014), 175–184.

Niloy Mitra, Michael Wand, Hao (Richard) Zhang, Daniel Cohen-Or, Vladimir Kim, and Qi-Xing Huang. 2013. Structure-aware Shape Processing. In *SIGGRAPH Asia 2013 Courses*. 1:1–1:20.

John. Morley. 1999. The history of furniture : twenty-five centuries of style and design in the Western tradition. (1999).

M. Raptis, I. Kokkinos, and S. Soatto. 2012. Discovering discriminative action parts from mid-level video representations. In *Proc. CVPR*. 1242–1249.

Ali S Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. 2014. CNN features off-the-shelf: an astounding baseline for recognition. In *IEEE CVPR Workshop*. IEEE, 512–519.

Szymon Rusinkiewicz, Forrester Cole, Doug DeCarlo, and Adam Finkelstein. 2008. Line Drawings from 3D Models. In *SIGGRAPH Course*.

Bilge Sayim and Patrick Cavanagh. 2011. What Line Drawings Reveal About the Visual Brain. *Frontiers in Human Neuroscience* 5 (2011), 118:1–118:4.

Karen Simonyan and Andrew Zisserman. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. *Computer Science* (2014).

Saurabh Singh, Abhinav Gupta, and Alexei Efros. 2012. Unsupervised discovery of mid-level discriminative patches. *Computer Vision–ECCV 2012* (2012), 73–86.

Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. 2015. Multi-view convolutional neural networks for 3D shape recognition. In *Proceedings of the IEEE International Conference on Computer Vision*. 945–953.

Kenshi Takayama, Ryan Schmidt, Karan Singh, Takeo Igarashi, Tamy Boubekeur, and Olga Sorkine. 2011. GeoBrush: Interactive Mesh Geometry Cloning. *Computer Graphics Forum* 30, 2 (2011), 613–622.

Yunhai Wang, Minglun Gong, Tianhua Wang, Daniel Cohen-Or, Hao Zhang, and Baoquan Chen. 2013. Projective analysis for 3D shape segmentation. *ACM Trans. on Graph.* 32, 6 (2013), 192.

Wikipedia. 2016. Style (visual arts) — Wikipedia, The Free Encyclopedia. (2016). https://en.wikipedia.org/w/index.php?title=Style_(visual_arts)&oldid=713614541 [Online; accessed 7-May-2016].

Zhige Xie, Kai Xu, Wen Shan, Ligang Liu, Yueshan Xiong, and Hui Huang. 2015. Projective Feature Learning for 3D Shapes with Multi-View Depth Images. In *Computer Graphics Forum*, Vol. 34. Wiley Online Library, 1–11.

Kai Xu, Vladimir G Kim, Qixing Huang, and Evangelos Kalogerakis. 2017. Data-Driven Shape Analysis and Processing. In *Computer Graphics Forum*, Vol. 36. Wiley Online Library, 101–132.

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. on Graph.* 29, 6 (2010), 184:1–184:10.

Kai Xu, Rui Ma, Hao Zhang, Chenyang Zhu, Ariel Shamir, Daniel Cohen-Or, and Hui Huang. 2014. Organizing heterogeneous scene collections through contextual focal points. *ACM Trans. on Graph.* 33, 4 (2014), 35.

Matthew D Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014*. Springer, 818–833.

L. Zelnik-Manor. 2004. Self-tuning spectral clustering. *Advances in Neural Information Processing Systems* 17 (2004), 1601–1608.