Organizing Heterogeneous Scene Collections through Contextual Focal Points

Kai Xu[1,3] Rui Ma[2] Hao Zhang[2] Chenyang Zhu[3] Ariel Shamir[4] Daniel Cohen-Or[5] Hui Huang[1]
[1]Shenzhen VisuCA Key Lab / SIAT [2]Simon Fraser University
[3]HPCL, National University of Defense Technology [4]The Interdisciplinary Center [5]Tel Aviv University

Abstract

We introduce *focal points* for characterizing, comparing, and organizing collections of complex and heterogeneous data and apply the concepts and algorithms developed to collections of 3D indoor scenes. We represent each scene by a graph of its constituent objects and define focal points as *representative* substructures in a scene collection. To organize a heterogeneous scene collection, we cluster the scenes based on a set of extracted focal points: scenes in a cluster are closely connected when viewed from the perspective of the representative focal points of that cluster. The key concept of representativity requires that the focal points occur *frequently* in the cluster and that they result in a *compact* cluster. Hence, the problem of focal point extraction is intermixed with the problem of clustering groups of scenes based on their representative focal points. We present a *co-analysis* algorithm which interleaves frequent pattern mining and subspace clustering to extract a set of *contextual* focal points which guide the clustering of the scene collection. We demonstrate advantages of *focal-centric* scene comparison and organization over existing approaches, particularly in dealing with *hybrid* scenes, scenes consisting of elements which suggest membership in different semantic categories.

Links: DL PDF WEB VIDEO CODE

"I can think of no better expression to characterize these similarities than 'family resemblances'; for the various resemblances between members of a family: build, features, colour of eyes, gait, temperament, etc. etc. overlap and criss-cross in the same way."

— Ludwig Wittgenstein [1953]

1 Introduction

Recent works on organizing and exploring 3D visual data have mostly been devoted to object collections [Ovsjanikov et al. 2011; Jain et al. 2012; Kim et al. 2012; van Kaick et al. 2013; Huang et al. 2013b]. In this paper, we are interested in analyzing and organizing visual data at a larger scope, namely, 3D indoor scenes. Even a moderately complex indoor scene would contain tens to hundreds of objects. Compared to the individual objects therein, a scene is more complex with looser structural and spatial relations among its components and a more diverse mixture of functional substructures. The latter point is attested by *hybrid* scenes which contain elements reminiscent of different semantic categories. For example, the middle scene in Figure 1 is partly a bedroom and partly a living room. The greater intra-class variabilities and richer characteristics in scene data motivate our work to go beyond providing only a holistic and singular view of a scene or a scene collection.

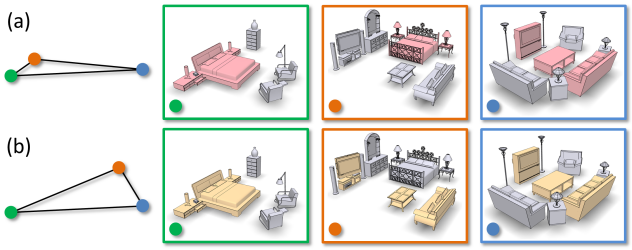

Figure 1: *We analyze and organize 3D indoor scenes in a heterogeneous collection from the perspective of* focal points *(sub-scenes in color). Scene comparisons may yield different similarity distances (left) depending on the focal points.*

We introduce the use of *focal points* for characterizing, comparing, and organizing collections of complex data and apply the concepts and algorithms developed to 3D indoor scenes. In particular, we are interested in organizing scenes in a *heterogeneous* collection, i.e., scenes belonging to multiple semantic categories. Analyzing complex and heterogeneous data is difficult without references to certain points of attention or focus, i.e., the focal points. For example, comparing New York City to Paris as a whole will unlikely yield a useful answer. The comparison is a lot more meaningful if it is focused on particular aspects of the cities, e.g., architectural style or fashion trends. One of the natural consequences of the focal point driven data view is that scene comparison may yield different similarity distances depending on the focal points; see Figure 1 for an illustration, as well as the accompanying video.

We represent an indoor scene by a graph of its constituent objects. A focal point, or focal, for short, is a *substructure* in a scene and corresponds to a subgraph. However, we are not interested in all sub-scenes. A key premise of our work is that meaningful focals should be determined *contextually*, in a set (Figure 2), and through a *co-analysis*. To illustrate, there are probably too many notable

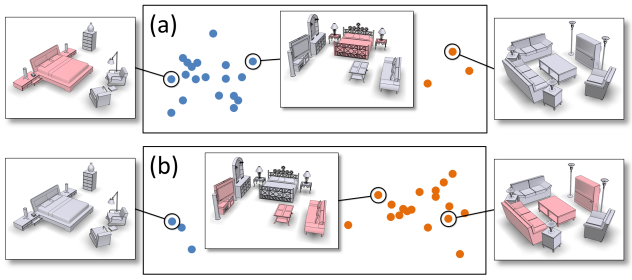

Figure 2: *Focal points (marked red in the scenes) are* contextual *and depend on scene composition in a collection. With more bedrooms (a) or more living rooms (b), different focals were extracted and hybrid scenes are pulled towards one of the clusters.*

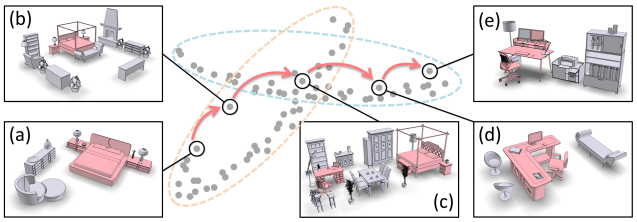

Figure 3: *Focal-driven scene clustering produces overlapping clusters. An exploratory path, (a) to (e), through an overlap, which often contains hybrid scenes (c) possessing multiple focals, can smoothly transition between the scene clusters. These scene clusters often characterize meaningful scene categories. In this example, the transition is from bedroom scenes to offices.*

aspects about Paris. When putting London and Paris together, one's focuses narrow down to, e.g., European capitals. If we throw New York and Milan into the mix, then most people are first reminded that the four cities are the fashion capitals of the world.

In this work, we are interested in extracting contextual focal points that are *representative* in a given scene collection. For a focal to be representative, it must occur sufficiently *frequently*. However, frequency analysis alone is insufficient. For example, chairs are likely to be found in almost all scenes, but they can hardly be regarded as representative of any meaningful scene groups, e.g., bedrooms or living rooms. We stipulate that representativity is also tied to a notion of coherence or *compactness* of the group of scenes the focal point is to represent or characterize. Therefore, frequency analysis for focal extraction is intermixed with clustering, which computes compact groups of scenes, where the scenes in each cluster are closely connected when viewed from the perspective of the representative focals of the cluster. Once again, the representative focals occur frequently in the cluster and they must also induce a compact cluster. To solve the two coupled problems simultaneously, we develop a co-analysis algorithm which interleaves *frequent pattern mining* [Han et al. 2007] and *subspace clustering* [Vidal 2011].

Focal points play a key role in our organization of a heterogeneous scene collection. First, we define compactness of a cluster based on a *focal-centric* scene-to-scene similarity, which builds on the rooted walk graph kernels of Fisher et al. [2011] and assigns higher weights to walks which originate from the representative focals of that cluster. Secondly, the scene organization is given by the clustering of scenes based on the representative focals extracted. Some scenes may contain multiple focals, thus belong to multiple clusters. Such scenes, typically of a hybrid nature, provide linkages or gateways between scene clusters, allowing an exploration of the scene organization to naturally transition between meaningful scene categories, as illustrated in Figure 3.

Our main contribution is a focal-driven analysis and organization of heterogeneous data collections. While we only consider 3D indoor scenes in this paper and we are not aware of previous works on co-analysis and organization of heterogeneous scene collections, the analysis is general and not confined to scene data. Important characteristics of our work which set it apart from previous approaches to organizing data collections include:

- Data are not compared holistically without discrimination. We develop a focal-centric scene descriptor for scene comparison, which supports scene analysis in *perspective*.
- Similarity distance between two scenes may be *non-unique*, i.e., it is based on the focals designated for comparison.
- Multiple views on scene data depend on focal points, leading to overlapping clustering of a scene collection, rather than a partition. The resulting organization is particularly suited for retrieving and exploring complex and hybrid scenes.

We show advantages of focal-centric scene comparison and organization over existing approaches, particularly in dealing with hybrid scenes. We also demonstrate new capabilities offered by the new data organization for scene retrieval and exploration.

2 Related work

Background. At a conceptual level, our work can be seen as a realization of the notion of "family resemblances" from the seminal work of Wittgenstein [1953]. A scene collection forms the "family", and the extracted focals represent the resemblances which "overlap and criss-cross" among the scenes. Works from cognitive psychology, in particular those by Rosch [1975], provided evidences that *perceptual and semantic categories are naturally formed in terms of focal points or prototypes* (see account in [Tversky 1977]), though the so-called "cognitive reference points" in her work referred to *whole* representatives of a category instead of featured substructures. The role of context in measuring data similarity has long been studied in various fields, e.g., [Biberman 1994; Jeh and Widom 2002]. Our work presents an algorithm for identifying conceptual focals which serve as reference points for comparing scenes in a heterogeneous collection.

Scene analysis. As the most familiar environments to humans, indoor scenes are ubiquitous in graphics applications such as virtual reality, gaming, and design. Much research in vision and graphics has been devoted to recognizing, classifying, and retrieving indoor scenes, e.g., [Rasiwasia and Vasconcelos 2008; Quattoni and Torralba 2009; Fisher et al. 2011; Juneja et al. 2013; Xu et al. 2013; Zhao et al. 2014], among others. Our work recognizes the difficulty in comparing complex scenes globally, e.g., via the classic graph kernels [Fisher et al. 2011]. We propose extracting and utilizing focal substructures for scene analysis. Of relevance are works which extract distinctive regions [Shilane and Funkhouser 2007; Juneja et al. 2013] that are representative of a semantic category. The focals we extract are not meant for scene recognition but organization; one focal may be shared by scenes from different categories.

Object collections and co-analysis. There have been a growing body of work on unsupervised co-analysis [Xu et al. 2012; Huang et al. 2012; van Kaick et al. 2013; Huang et al. 2013a; Zheng et al. 2013] and organization of 3D object collections [Ovsjanikov et al. 2011; Jain et al. 2012; Kim et al. 2012; Huang et al. 2013b]. Similar works exist on image collections, e.g., for image co-salience detection [Cheng et al. 2014]. In most cases, co-analysis operates on objects belonging to the *same* semantic category. An exception is the recent work of Huang et al. [2013b] which performs qualitative analysis on heterogeneous object collections. However, their object comparison employs global shape descriptors while still resulting in unique qualitative distances, in terms of number of "hops" in a tree representation, between objects.

Another recent work, the co-hierarchical analysis of van Kaick et al. [2013], also employs a clustering approach and the clustering partitions a set of shapes into different modes of structural variation. While hierarchical models offer the flexibility to account for structural variations, they still provide only a *single* view on each shape. Our representation allows multiple views of a scene model, each of which may be seen as from the perspective of a particular focal point. Moreover, our analysis produces overlapping clusters which characterize the underlying data with larger granularity.

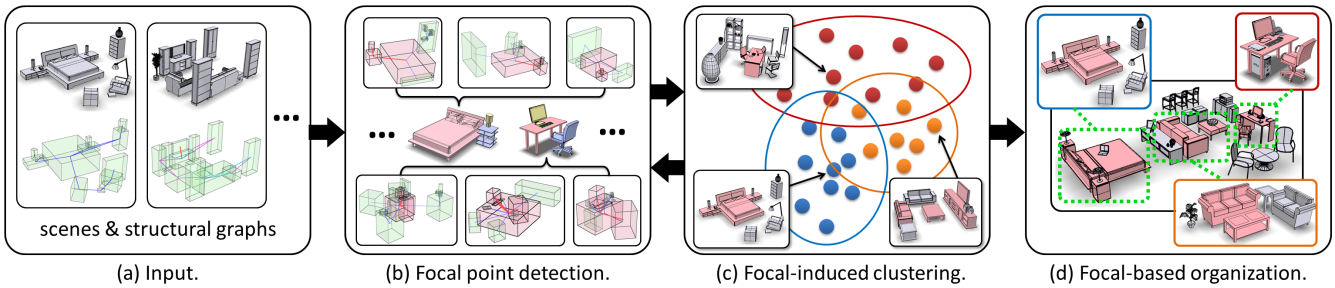

Figure 4: *An overview of our algorithm. The input is a heterogeneous collection of 3D indoor scenes. We represent each scene by a structural graph (a). The co-analysis algorithm is iterative, between (b) and (c). Each iteration involves an interleaving optimization consisting of focal point detection (b) and focal-induced scene clustering (c). After the set of contextual focals are obtained, the entire scene collection can be organized with the focals serving as the interlinks between scenes from various clusters (d).*

Contextual analysis. Part-in-whole or object-in-scene types of retrievals have been studied in semantic analysis of 3D objects or indoor scenes. Shapira et al. [2009] define the context for a shape part within an extracted part hierarchy. The series of work from Fisher et al. rely on spatial and semantic relations among the scene objects for context-based object search [Fisher and Hanrahan 2010; Fisher et al. 2011] or object replacement for scene synthesis [Fisher et al. 2012]. In all of these works, substructures in a scene provide the contexts for characterizing individual objects therein. We treat the substructures as explicit scene features, i.e., potential focals, and perform contextual analysis in a larger scope.

One possible way to find salient substructures in a scene collection is to extract object groups based on co-occurrences of object *categories*, like in the work of Xu et al. [2013]. In contrast, we group scene objects, rather than object categories, to form focals. Furthermore, the grouping in Xu et al. [2013] is based on frequency analysis only, while we perform both frequent pattern mining and subspace clustering for focal point extraction. Singh et al. [2012] detect mid-level discriminative patches from a set of unlabeled images by alternating between clustering and training discriminative classifiers. A similar idea is then applied to extract, from a large repository of geo-tagged imagery, visual features which are both frequently occurring and geographically distinctive under weak supervision [Doersch et al. 2012]. Our co-analysis is unsupervised, driven by a novel cluster compactness objective for both focal selection and focal-induced clustering.

Frequent pattern mining. Frequent pattern mining has been an extensively studied topic in data mining [Han et al. 2007]. The most relevant works are those designed for frequent subgraph mining, e.g., [Yan and Han 2002], which are primarily based on subgraph isomorphism testing. Directly adapting these methods to our problem setting is infeasible since the relations among objects in our input graphs are loose and possibly uncertain. We adopt inexact subgraph matching formulated by graph edit distances [Riesen et al. 2010] where the edit cost is defined based on spatial arrangements between scene objects. It is also worth noting that frequency of occurrence is not the only criterion for focal point selection. The subsequent cluster analysis further adjusts the extracted focals.

Subspace clustering. Subspace clustering clusters high-dimensional data into multiple subspaces, each modeled by a subset of features [Vidal 2011]. At a high level, the clustering problem we face has a similar setting as subspace clustering, where focals act as the feature subsets and characterize the subspaces that contain the clusters of scenes. Subspace analysis via spectral clustering has been one of the most effective approaches to subspace clustering [Wang et al. 2011a]. However, spectral clustering always produces a partition. In our work, we perform cluster attachment to reveal cluster overlap based on their representative focals, making the obtained clusters better reflect the complexity and heterogeneity of the data collection.

3 Overview

The input to our algorithm is a heterogeneous collection of 3D indoor scenes collected from public repositories. Such scenes typically come with semantic labels for the objects and the scenes themselves. Our analysis uses the object labels but never the scene labels. Our goal is to extract a set of contextual focals, as well as a clustering of the scenes based on these focals; see Figure 4.

For each scene, a structural graph is constructed which encodes two types of relationships between scene objects: support and proximity. Our main algorithm consists of a coupled optimization whose objective is to maximize the overall compactness of the scene clusters while ensuring that the focals represent their respective clusters effectively. A key is that each representative focal is sufficiently *discriminative* so that it is frequent only within the cluster it represents or characterizes. The optimization is *iterative*, where each iteration interleaves between cluster-guided focal point mining and focal-induced subspace clustering of the scenes; see Figure 5.

The first and initial phase of the optimization is to extract frequent substructures as focals from the input structural graphs, via subgraph mining (Section 4.1). Rather than relying on subgraph isomorphism, we perform inexact graph matching which insists on consistency of node labeling but not edge connection. The latter is to account for loose relations between corresponding objects across a large heterogeneous scene collection. The matching of such relations is based on a layout similarity measure between spatial arrangements of objects. This matching is confined by scene grouping resulting from the most recent clustering phase. Specifically, the subgraph matching is *weighted* so that the substructures found are frequent only within the clusters they characterize.

In the second phase, based on the extracted focals, we perform subspace clustering (Section 4.2) on the scenes. The structural graphs are clustered so that each cluster is characterized by a subset of current focals. Generally, the representative focals for a cluster are not unique. The clustering step seeks to maximize the compactness of all clusters, where compactness is defined by a scene-to-scene similarity based on *focal-centric graph kernels* (FCGK). We define FCGK based on the work of Fisher et al. [2011] which utilizes rooted walk graph kernels. However, instead of weighting equally walks from all sources, we weigh more heavily those walks which originate from representative focals in the graphs. The maximization is based an *iteratively reweighted subspace clustering* scheme

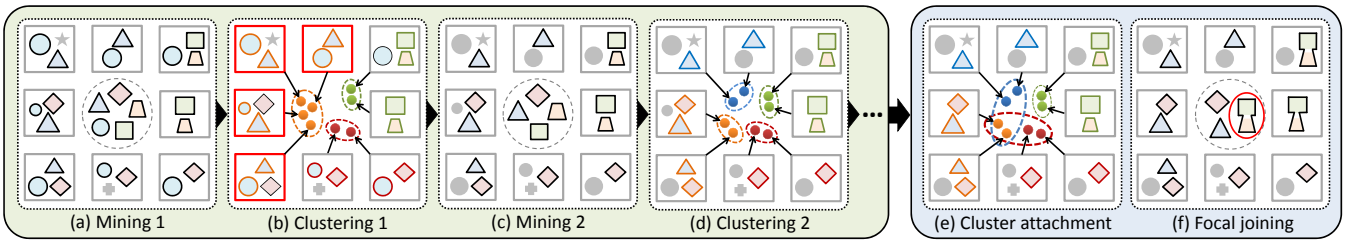

Figure 5: *Illustration of iterative optimization pipeline. A scene is depicted with a grey box enclosing several substructures represented by circles, squares, diamonds, triangles, etc. To initialize the interleaving optimization, we first detect a set frequent substructure shown in the middle in (a). Based on that, subspace clustering leads to incorrect clusters (marked in red) due to the trivial substructure (circle) occurring in most scenes. Then we perform cluster-guided weighted mining which eliminates the trivial substructure. Following that, a more accurate clustering result is obtained in (d) based on the new set of discriminant substructures in (c). Finally, we perform cluster attachment to reveal overlapping clusters (the red and yellow clusters in (e)), as well as focal joining to discover non-local focal points (marked in red in (f)).*

we develop, which gradually increases cluster compactness.

Finally, once the clusters and focals are determined by the optimization, we perform *cluster attachment* and *focal joining* (Section 4.3). Some clusters share scenes containing multiple focals, each characterizing a different cluster. These clusters are naturally attached at the shared scenes. Within a cluster, multiple local substructures may occur concurrently across all or most scenes. These substructures are naturally joined to form non-local focals. Note that such non-local focals could not be detected via subgraph mining since only spatially close objects are connected in the graphs.

4 Focal-driven scene co-analysis

For each input scene, we construct a structural graph (Figure 6(b)) whose nodes are scene objects and edges encode spatial relationship, support and proximity, between objects; see Algorithm 1. Both nodes and edges are labeled, by object semantic labels and relationship types (support or proximity), respectively.

We first detect all support relationship between objects by testing vertical contacts between their shape geometries. Second, we add a proximity edge from any object that is not connected by a support edge, to the object which has the strongest connection with it, where connection strength (Equation 3) is defined as a part of layout similarity. Third, we ensure that any group of symmetric objects has symmetric connections to other objects, if any.

We detect all groups of mutually symmetric objects and examine for each group all outside objects connecting to that group. If more than two symmetric objects in the group have similar spatial arrangement (Equation 5) with respect to an outside object, we ensure they all connect to the outside object with edges of the same type, depending on their relationship against the outside object. To detect mutually symmetric objects, i.e., objects possessing similar geometry, we adopt the registration method described in [Wang et al. 2011b]. Finally, we detect the connected components in the current graph, and connect the components with proximity edges to make sure the entire scene is represented by a connected graph.

Our co-analysis operates on these structural graphs. The main algorithm involves a coupled optimization for both focal point mining and scene clustering. The objective of the optimization is

$$\max_{\mathcal{F},\Omega} \sum_{\ell=1}^{c} n_\ell \kappa_\ell(\mathcal{F},\Omega) \tag{1}$$

where $\mathcal{F} = \{F_k\}_{k=1}^{n}$ are the set of focal points, and $\Omega = \{\mathcal{C}_\ell\}_{\ell=1}^{c}$ the set of clusters. κ_ℓ denotes the compactness of cluster $\mathcal{C}_\ell$ based on FCGK, and n_ℓ is the size of cluster $\mathcal{C}_\ell$. We optimize iteratively with the iterations continuing until the overall compactness of the clusters converges, specifically, when the change of the objective function is less than 1.0×10^{-6}. In the following sections, we detail our co-analysis algorithm.

4.1 Focal extraction via graph mining

A substructure of a scene consists of a group of nearby objects along with their spatial arrangement; it is a subgraph. We could define focals as substructures that occur frequently across a large number of semantically related scenes, e.g., bedrooms. However, since scene labels can be unknown or ambiguous, especially for hybrid scenes, we do not use them. Instead, we couple focal detection with the identification of meaningful clusters. If a substructure occurs in a scene, we say that the scene *supports* that substructure. The notion of occurrence will be quickly relaxed by inexact graph matching, which is enabled by a similarity measure of spatial layout between substructures of scenes.

Layout similarity. We define a layout similarity between two substructures by examining the pair-wise spatial arrangement of oriented bounding boxes (OBBs) of the objects in the substructures. Suppose we are given two substructures represented by two subgraphs in the structural graphs of two scenes: $S_a \subset G_A$ and $S_b \subset G_B$. The layout dissimilarity between them is defined as:

$$D_{\text{layout}}(S_a, S_b) = \sum_{\substack{\{p,q\}\in S_a,\\ \{\theta(p),\theta(q)\}\in S_b}} d_{\text{arr}}(\langle p,q\rangle, \langle \theta(p),\theta(q)\rangle), \tag{2}$$

Algorithm 1: Structural Graph Construction

Input : scene $C = \{O_i\}_i$
Output: structural graph $G = \langle V, E\rangle$

$\forall O_i \in C, V \leftarrow V \cup \{v_i\}$; // *vertices*
$E \leftarrow E \cup$ `SupportEdge(`C`)` ; // *support edges*
$E \leftarrow E \cup$ `ProximityEdge(`C, E`)` ; // *proximity edges*
$\mathcal{U} \leftarrow$ `DetectSymGroup(`C`)`;
foreach $U \in \mathcal{U}$ **do** // *for each group of symmetric objects*
  foreach $v \in V - U$ **do** // *for each outside object*
    if $\exists s,t \in U; \langle v,s\rangle, \langle v,t\rangle \in E; d_{\text{arr}}(\langle v,s\rangle, \langle v,t\rangle) < 0.1$ **then**
      foreach $u \in U$ **do** // *do symmetric connection*
        $E \leftarrow E \cup \{\langle u,v\rangle\}$;
$E \leftarrow E \cup$ `ConnectComponents(`V, E`)`;
return G;

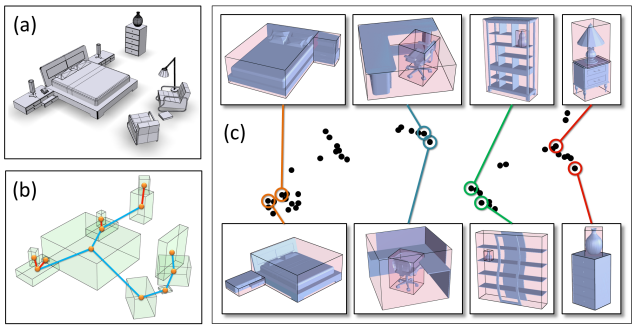

Figure 6: *The structural graph (b) of the input scene (a) encodes two types of relationship: support (red) and proximity (blue). (c) plots the layout similarity of object pairs after spectral embedding.*

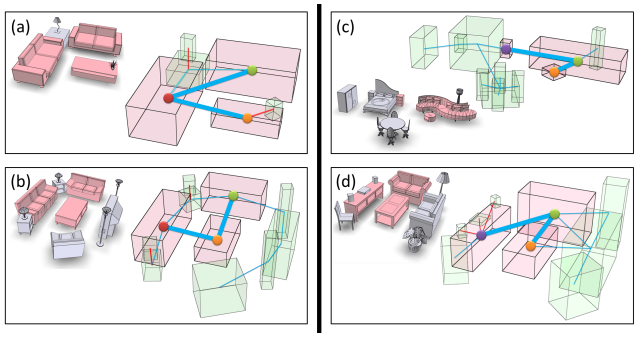

Figure 7: *Scenes (a) and (b) have the same sub-scenes represented with different subgraphs. (c) and (d) have the same subgraphs while the layouts of the corresponding sub-scenes are different.*

where $\theta(p) \in G_B$ is the corresponding object of $p \in G_A$. Such correspondences can be determined during subgraph mining, as described below. d_{arr} measures the *spatial arrangement* dissimilarity between two pairs of objects which is defined based on two factors. The first is the *connection strength* between objects p and q:

$$\gamma(p,q) = \frac{d_{\text{H}}(obb(p), obb(q))}{dl(p) + dl(q)}, \tag{3}$$

where d_{H} is Hausdorff distance, $obb(p)$ the OBB of object p, and $dl(p)$ the diagonal length of $obb(p)$. The second factor is the angle between the upright vector and the vector between p and q:

$$\rho(p,q) = angle(\mathbf{v}_{\text{dir}}(p,q), \mathbf{v}_{\text{upright}}), \tag{4}$$

where $\mathbf{v}_{\text{dir}}(p,q)$ is the vector from the larger object of the two to the smaller one and $\mathbf{v}_{\text{upright}}$ the upright vector. The dissimilarity of spatial arrangement between two object pairs $\langle p,q\rangle$ and $\langle s,t\rangle$ is then defined as:

$$\begin{aligned} &d_{\text{arr}}(\langle p,q\rangle, \langle s,t\rangle) \\ &= \alpha|\tilde{\gamma}(p,q) - \tilde{\gamma}(s,t)| + (1-\alpha)|\tilde{\rho}(p,q) - \tilde{\rho}(s,t)|. \end{aligned} \tag{5}$$

$\tilde{\gamma} = e^{-\gamma^2/(\sigma\gamma_{\max})^2}$ is normalized connection strength where $\sigma = 0.4$ and the maximum value $\gamma_{\max}$ is found for all pairs of objects. ρ is normalized similarly. We use $\alpha = 0.6$ in our implementation. Figure 6(c) shows a few examples of similar layouts.

Frequent substructure mining. Frequent subgraph mining extracts from a set of input graphs $\mathcal{G} = \{G_i\}_{i=1}^n$, a set of subgraphs $\mathcal{F} = \{F_k\}_{k=1}^d$, which frequently occur (more than a given threshold value $s_{\min}$) in the input graphs based on subgraph isomorphism. We define:

$$\mathcal{F} = \{F_k \mid |\mathcal{S}_k| = \sum_{i=1}^{n} x_{ik} > s_{\min}\} \tag{6}$$

where $x_{ik} = I(F_k \subseteq G_i)$ is an indicator function for subgraph isomorphism and $\mathcal{S}_k = \{G_i \mid x_{ik} = 1\}$ is the *supporter set* of F_k.

Directly applying frequent subgraph mining to structural graphs is ineffective since the the proximity relationships are not necessarily consistent across different scenes, e.g., see Figure 7(a,b). One may then resort to inexact graph matching, e.g., based on graph edit distance [Riesen et al. 2010]. However, the large search space of inexact subgraph mining makes such approaches prohibitive.

We propose a two-step scheme for frequent substructure mining (Algorithm 2) which carries out inexact graph matching efficiently. We first perform frequent subgraph mining based on exact subgraph isomorphism, using gSpan [Yan and Han 2002], with a relatively low minimal support threshold (Line 1 in Algorithm 2). Then, in the second step, we employ inexact subgraph matching [Riesen et al. 2010] to match the frequent subgraphs mined in the previous step against all graphs in the set, to expand their support (Lines 2-6). Note that in both steps, the matching of graph nodes is exact and based only on node labels.

To create tolerance for different proximity connection graph structure, we use *error correction* of the subgraphs by introducing three edit operations on graph edges: insertion and deletion of proximity-type edges, as well as substitution between two proximity edges. The edit cost of each operation is defined as the spatial arrangement dissimilarity (Equation 5) between the two pairs of objects involved. If the total edit cost $\delta(G_i, F_k)$ for matching F_k and G_i is less than $\delta^{\text{t}} = 0.1$, we add G_i to F_k's supporter set.

For a frequent subgraph F_k, we have obtained its embedding in any of its supporter graphs during the mining step, denoted as $G_i(F_k) \subseteq G_i$, $G_i \in \mathcal{S}_k$. However, the embedding of F_k in its supporters may have different layouts since the exact mining step is layout-oblivious, e.g., as shown in Figure 7(c,d). We locate and remove weak (or outlier) supporters in which the embedded subgraph has significantly different layout from those in the other supporters (Lines 7-10). Specifically, given a supporter $G_i \in \mathcal{S}_k$ of F_k, we compute the average dissimilarity between its corresponding embedding and those in all other supporters,

$$\varphi(G_i, F_k) = \sum_{G_j \in \mathcal{S}_k, i \neq j} D_{\text{layout}}(G_i(F_k), G_j(F_k)),$$

and filter out this supporter if the value exceeds a threshold $\varphi^{\text{t}} = 0.3|\mathcal{S}_k|$. Finally, we remove those subgraphs whose number of supporters falls below the minimal support threshold $s_{\min}$ (Line 12).

Cluster-guided weighted mining. Our goal is to detect representative focal points characterizing a meaningful clustering of the input scenes, and not substructures which are frequent over the entire collection. Therefore, instead of relying on the frequency criterion in Equation (6), we base our substructure mining on the *current clusters* and perform weighted subgraph mining [Tsuda and Kudo 2006]. For each cluster $\mathcal{C}_\ell$, we define *supporting weights* $(\varpi_{\ell i})_{i=1}^n$ as a measure of support of G_i to any substructure. A substructure is detected as frequent if its weighted sum of support, denoted by *discriminant score* $\eta_{\ell k}$, is greater than a threshold η_ℓ^{t}:

$$\mathcal{F}_\ell = \{F_k \mid \eta_{\ell k} > \eta_\ell^{\text{t}}\} \text{ where } \eta_{\ell k} = \left|\sum_{i=1}^{n} \varpi_{\ell i}(2x_{ik} - 1)\right|. \tag{7}$$

Algorithm 2: Extended Frequent Substructure Mining

Input : structural graphs $\mathcal{G} = \{G_i\}_i$, minimal support $s_{\min}$
Output: frequent substructures $\mathcal{F} = \{\langle F_k, \mathcal{S}_k\rangle\}_k$

$\mathcal{F} = \{\langle F_k, \mathcal{S}_k\rangle\}_k \leftarrow$ `MineSubgraph` $(\mathcal{G}, s_{\min})$;
foreach $G_i \in \mathcal{G}$ **do** // *expand support*
  foreach $\langle F_k, \mathcal{S}_k\rangle \in \mathcal{F}$ **do**
    $\delta(G_i, F_k) \leftarrow$ `ErrorCorrectMatch` (F_k, G_i);
    if $\delta(G_i, F_k) < \delta^{\mathrm{t}}$ **then**
      $\mathcal{S}_k \leftarrow \mathcal{S}_k \cup \{G_i\}$;
foreach $\langle F_k, \mathcal{S}_k\rangle \in \mathcal{F}$ **do** // *filter support*
  foreach $G_i \in \mathcal{S}_k$ **do**
    if $\varphi(G_i, F_k) > \varphi^{\mathrm{t}}$ **then**
      $\mathcal{S}_k \leftarrow \mathcal{S}_k - \{G_i\}$;
  if $|\mathcal{S}_k| < s_{\min}$ **then**
    $\mathcal{F} \leftarrow \mathcal{F} - \{\langle F_k, \mathcal{S}_k\rangle\}$;
return $\mathcal{F}$;

By using positive weights $\varpi_{\ell i}$, if G_i belongs to $\mathcal{C}_\ell$, and negative otherwise, the discriminant score favors a substructure which is frequent in cluster $\mathcal{C}_\ell$ and penalizes its frequency in other clusters. Therefore, the mined substructures in $\mathcal{F}_\ell$ are frequent mainly within cluster $\mathcal{C}_\ell$. Specifically, we set $\varpi_{\ell i} = x_{\ell i}/n_\ell - 1/n$, where $x_{\ell i} = I(G_i \in \mathcal{C}_\ell)$, and $\eta_\ell^{\mathrm{t}} = \mu n/n_\ell$. We fix $\mu = 0.1$ in our algorithm. The final set of focal points takes the union of per-cluster discriminant substructures: $\mathcal{F} = \bigcup_{\ell=1}^{c} \mathcal{F}_\ell$, where c is the number of clusters. To achieve weighted mining, we evaluate the discriminant score of the individual substructures, which are efficiently enumerated by gSpan, and identify the discriminative ones based on the current clusters. Then we perform support expanding and filtering for the extracted substructures. In the first iteration, when clustering is missing, we use unweighted frequent substructure mining.

4.2 Focal-induced scene clustering

With the focals extracted, we perform subspace clustering to group the input scenes according to the extracted focals that they "share", i.e., the scenes contain and support the same focal. For each scene, we build a high-dimensional feature vector for clustering. The feature is defined by the set of all extracted focals in the most current focal mining step (Section 4.1). Each entry of the feature vector is an indicator of support of the scene to the corresponding focal, forming a Bag-of-Words (BoW) feature: $\mathbf{x}_i = (x_{ik})_{k=1}^{d}$. Subspace clustering is then performed over all input data represented in the feature space, $\mathbf{X} = [\mathbf{x}_i]_{i=1}^{n} \in \mathbb{R}^{d\times n}$, to extract clusters characterized by a low-dimensional subspace.

For subspace clustering, we adopt the method of Wang et al. [2011a] on subspace segmentation via quadratic programming (SSQP), a state-of-the-art spectral clustering based approach. The basic idea of SSQP is to express each datum $\mathbf{x}_i$ as a linear combination of all other data in the dataset, $\mathbf{x}_i = \sum_{j\neq i} z_{ij}\mathbf{x}_j$, while implicitly enforcing the coefficients z_{ij} to be zero for all $\mathbf{x}_j$ which belongs to different subspace from $\mathbf{x}_i$. To learn such a coefficient matrix $\mathbf{Z} \in \mathbb{R}^{n\times n}$, it solves the following constrained optimization problem:

$$\begin{aligned} \min_{\mathbf{Z}} f(\mathbf{Z}) &= \|\mathbf{X}\mathbf{Z} - \mathbf{X}\|_F^2 + \beta\|\mathbf{Z}^T\mathbf{Z}\|_1 \\ \text{s.t.} \quad & \mathbf{Z} > \mathbf{0}; diag(\mathbf{Z}) = \mathbf{0}, \end{aligned} \tag{8}$$

where $\|\cdot\|_F$ is the Frobenius norm and $diag(\mathbf{Z})$ the diagonal vector of matrix $\mathbf{Z}$. The ℓ_1-regularization term enforces sparsity of the solution, leading to feature selection for subspace clustering. The problem is a linear constrained quadratic programming which can be solved efficiently. The resulting coefficient matrix then forms an affinity matrix, $|\mathbf{Z} + \mathbf{Z}^T|/2$, based on which spectral clustering is applied to obtain the clustering result. To automatically determine the number of clusters, we employ self-tuning spectral clustering [Zelnik-Manor and Perona 2004]. In practice, the cluster count is relatively stable throughout the iterations since the structure of the BOW feature matrix does not change significantly.

Algorithm 3: Iteratively Reweighted Subspace Clustering

Input : structural graphs $\mathcal{G} = \{G_i\}_{i=1}^{n}$,
  BoW features: $\mathbf{X} = [\mathbf{x}_i]_{i=1}^{n}$, $(\mathbf{x}_i = (x_{ik})_{k=1}^{d})$
  weights: $\mathbf{W} = [\mathbf{w}_i]_{i=1}^{n}$, $(\mathbf{w}_i = (w_{ik})_{k=1}^{d})$
Output: subspace clusters $\{\mathcal{C}_\ell\}_{\ell=1}^{c}$

for $i = 1$ **to** n **do**
  $\mathbf{w}_i \leftarrow \mathbf{1}$;
repeat
  $\{\mathcal{C}_\ell\}_{\ell=1}^{c} \leftarrow$ `SubspaceClustering` $(\mathcal{G}, \mathbf{X}, \mathbf{W})$;
  for $\ell = 1$ **to** c **do** // *update weights*
    $\mathcal{R}_\ell \leftarrow$ `RepresentativeFocalSet` $(\mathcal{C}_\ell)$;
    $\kappa_\ell \leftarrow$ `Compactness` $(\mathcal{C}_\ell, \mathcal{R}_\ell)$;
    foreach $G_i \in \mathcal{C}_\ell$ **do**
      foreach $F_k \in \mathcal{R}_\ell$ **do**
        $w_{ik} \leftarrow n_\ell \cdot \kappa_\ell \cdot \eta_{\ell k}$;
  for $k = 1$ **to** d **do**
    if $F_k \notin \bigcup_{\ell=1}^{c} \mathcal{R}_\ell$ **then**
      for $i = 1$ **to** n **do**
        $w_{ik} \leftarrow 0$;
until *the overall compactness* $\sum_{\ell=1}^{c} n_\ell\kappa_\ell$ *does not improve*;
return $\{\mathcal{C}_\ell\}_{\ell=1}^{c}$;

Besides the clustering result, we need to identify the representative focals which characterize the clusters. For each cluster $\mathcal{C}_\ell$, we identify a set of representative focals, denoted as $\mathcal{R}_\ell$. We rank the importance of all focals supported by any structural graph in the cluster based on their discriminant score $\eta_{\ell k}$; see Equation (7). The top ranked focal is selected as the representative one. We select the top focals from the list until the i-th one, when there are over $p^{\mathrm{c}} = 80\%$ of the structural graphs in the cluster which support these top i focals simultaneously.

Our ultimate goal is to maximize the compactness of all clusters based on a scene-to-scene similarity emphasizing their representative focal points. The subspace clustering above is based on indicator features, which capture the *occurrence* of the focals but are not sufficiently informative to reflect the actual scene similarity. Directly incorporating focal-centric scene similarity into the subspace clustering is infeasible since the representative focals are unknown before the feature selective clustering is performed. Therefore, we propose an *iteratively reweighted subspace clustering* process to gradually produce more compact clusters where the compactness is measured based on the focal-centric graph kernel (FCGK).

Focal-centric graph kernel. Given a cluster $\mathcal{C}_\ell$, its compactness is defined as the average distance between all pairs of structural graphs belonging to it, measured by the FCGK:

$$\kappa_\ell = \frac{1}{n_\ell^2} \sum_{G_i, G_j \in \mathcal{C}_\ell} k_G^p(G_i, G_j), \tag{9}$$

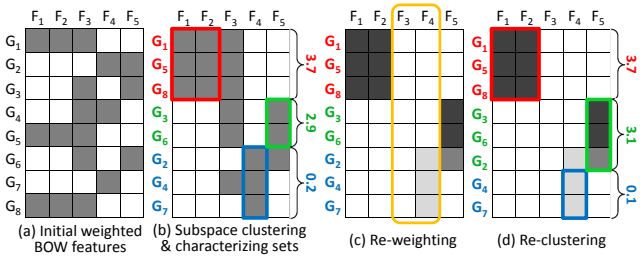

Figure 8: *An mini-experiment on reweighted subspace clustering. The weighted BoW features are shaded in grey level (dark=large; light=small). From the initial BoW features (a), subspace clustering produces three clusters (colored) along with their representative focals (marked in corresponding color). The colored numbers indicate the compactness values of clusters. F_3 is not discriminant as it appears across three clusters (b) so in (c) the corresponding weights are set to 0. The weights for F_4 are decreased due to the low compactness of the blue cluster. The next clustering groups G_2 into the green cluster with F_5 as the representative focal point (d).*

where $k^p_G(\cdot,\cdot)$ is the *weighted* p-th order walk graph kernel:

$$k^p_G(G_i, G_j) = \sum_{r\in G_i, s\in G_j} \lambda_{r,s} k^p_R(G_i, G_j, r, s). \qquad (10)$$

$k^p_R(G_i, G_j, r, s)$ is the p-th order rooted-walk graph kernel [Fisher et al. 2011] which we briefly review below for completeness. It compares nodes r and s, in graphs G_i and G_j, respectively, by comparing all walks of length p whose first node is r against all walks of length p whose first node is s:

$$k^p_R(G_i, G_j, r, s) = \sum_{\substack{(r_1,e_1,\ldots,e_{p-1},r_p)\in W^p_{G_i}(r)\\ (s_1,f_1,\ldots,f_{p-1},s_p)\in W^p_{G_j}(s)}} k_{\mathrm{n}}(r_p, s_p) \prod_{i=1}^{p-1} k_{\mathrm{n}}(r_i, s_i) k_{\mathrm{e}}(e_i, f_i),$$

where $W^p_G(r)$ is the set of all walks of length p originated from r in graph G. The node kernel k_{n} takes both geometry and label comparison into account, similar to [Fisher et al. 2011], except that we used a single label for each object, instead of a series of semantic tags. For edge kernel k_{e}, we use the similarity of spatial arrangement (Equation 2), instead of a binary comparison of edge types.

For the walk kernel κ_ℓ to be focal-centric, we set higher weight for those rooted walks which originates from a node in a representative focal of cluster $\mathcal{C}_\ell$:

$$\lambda_{r,s} = \begin{cases} 1+\lambda\cdot\eta_{\ell k} & \text{if } r\in G_i(F_k), s\in G_j(F_k) \text{ and } F_k\in\mathcal{R}_\ell \\ 1 & \text{otherwise} \end{cases}$$

where λ is a scaling factor. In our algorithm, we set $\lambda = 100$ which is fairly high and emphasizes more the role of focals in scene characterization than the overall scene similarity.

Iteratively reweighted subspace clustering. For a structural graph G_i, we weight the individual dimensions of its BoW feature vector by a weight vector $\mathbf{w}_i = (w_{ik})_{k=1}^d$ and solve a weighed subspace clustering which minimizes the error of linear approximation in Equation (8) under a *weighted* Frobenius norm. Specifically, we replace the first term in Equation (8) by:

$$\|\mathbf{XZ}-\mathbf{X}\|^2_{W,F} = \sum_{i=1}^n\sum_{k=1}^d w^2_{ik}[(\mathbf{XZ})_{ik}-\mathbf{X}_{ik}]^2. \qquad (11)$$

The weights allow us to tune the importance of the individual dimensions when seeking subspaces and can be utilized to iteratively shift clustering results. For example, one can increase the weights corresponding to the dimensions spanning the subspace of a cluster obtained in the last round, to reinforce the cluster in the current clustering. In our case, we encourage the reoccurrence of the compact clusters in the next iteration by increasing the weights of the dimensions corresponding to its representative focal points, and deprecate incompact clusters by decreasing their corresponding weights.

Initially, the weights in $\mathbf{w}_i$ are set uniformly to 1. In each iteration, we perform the weighted subspace clustering and then update $\mathbf{w}_i$ based on the compactness of the cluster to which G_i belongs; see Algorithm 3. For each member of a cluster, we compute the weights of the dimensions corresponding to the representative focals of the cluster based on cluster compactness and focal point discriminant score (Line 10). If a focal is not a representative one for any cluster, we set a 0 for the corresponding dimension of the weight vector for all structural graphs (Line 11-14). The stopping criteria for this iterative process is the same as the one used during the interleaving optimization, i.e., the change of overall cluster compactness.

Figure 8 demonstrates the process of reweighted subspace clustering with a mini-experiment on 8 structural graphs with 5 focals. In the experiment, after obtaining the subspace clustering along with the representative focals, the weights corresponding to focal point F_3 and F_4 are decreased, due to low discriminant score and low cluster compactness, respectively. With the updated weights, G_2, which was originally clustered into the blue cluster due to F_4, is now grouped into the green one characterized by F_5. This is because F_5 plays the major role in clustering G_2 after F_4 is deprecated. After reweighting, the weighted feature vector of some structural graphs may decrease to (or close to) $\mathbf{0}$ vector (e.g., G_4 and G_7 in Figure 8). Since the clustering of these structural graphs is quite unpredictable, we choose to leave them out when their weight vector vanishes, to make the iterative clustering converge faster. These structural graphs are later introduced back in the beginning of the next round of interleaving optimization.

4.3 Cluster attachment and focal joining

Cluster attachment. Spectral clustering produces a partition of an input dataset, which does not reflect potential cluster overlapping due to scenes which exist in multiple clusters. In general, a structural graph for an input scene which support multiple focals may belong to multiple clusters that have other different representative focals. We simply attach such clusters with respect to the shared scenes, which can be easily identified, to reveal the overlap.

Focal joining. As subgraph mining is performed on structural graphs whose node connections only capture local proximity, it is unable to return large-scale and non-local substructures. This issue has been observed in the recent work of Xu et al. [2013] which is based on structure group detection over the structural graphs. In our work, frequent substructure detection is coupled with subspace clustering. This enables us to combine the extracted focals to form a larger and non-local substructure, through analyzing the clusters they characterize. Suppose that F_1 and F_2 are both representative focals for some cluster $\mathcal{C}_\ell$. If their supporter sets in $\mathcal{C}_\ell$, denoted as $\mathcal{S}_{\ell 1}$ and $\mathcal{S}_{\ell 2}$, overlap sufficiently, i.e., $|\mathcal{S}_{\ell 1}\cap\mathcal{S}_{\ell 2}| > 0.9\min\{|\mathcal{S}_{\ell 1}|, |\mathcal{S}_{\ell 2}|\}$, we join them, by a union of their nodes, to form a larger substructure F_{12} as a representative focal for $\mathcal{C}_\ell$.

5 Results

We present results obtained by our algorithm for focal point driven analysis of indoor scene collections. For scene retrieval, we com-

Collection	$\#f$	$\#nlf$	f_{min}	f_{avg}	f_{max}	$\%mf$
Stanford	24	4	2	3	6	50.4%
Tsinghua	34	7	2	3	5	46.1%

Table 1: *Statistics for focal point extraction. #f denotes the total number of focals and #nlf that of non-local ones. The minimum, average, and maximum number of objects in an extracted focal is denoted by f_{min}, f_{avg}, and f_{max}. %mf is the percentage of multi-focal scenes over the whole collection.*

pare our results to those obtained from state-of-the-art methods both through precision-recall curves and a preliminary user study, targeted for hybrid scenes. More extensive results and an accompanying video can be found in the supplementary material.

Datasets. The datasets we experiment on were provided by the Stanford repository [Fisher et al. 2012] and the Tsinghua repository [Xu et al. 2013]. Both datasets contain semantic tags with the objects originally collected from Google (now Trimble) 3D Warehouse. Since the tags from the two datasets are inconsistent, we run our test on each dataset separately. For each scene, we remove the walls and focus only on the interior scene objects. The Stanford collection consists of 132 scenes and 3,461 objects, encompassing 78 object categories and five labeled scene categories. The Tsinghua dataset consists of 792 scenes and 13,365 objects, encompassing 119 object categories and six labeled scene categories. The Tsinghua dataset contains 102 hybrid scenes which is composed of many subscenes, each representing a room.

Parameters and statistics. The key parameters of our algorithm include: the minimum support s_{min} used for frequent substructure mining in the first iteration, and the rooted paths combination weights used in computing graph kernel. All the results reported in the paper were obtained with the same parameter setting: $s_{min} = 40$ for Tsinghua dataset and $s_{min} = 20$ for the Stanford dataset. The parameters for graph kernel use the optimal ones available from the published work of Fisher et al. [2011]. Values for all other parameters are fixed throughout and described in Section 4.

Statistics and timing. Table 1 shows some statistics from focal point extraction and scene clustering. Timing wise, it took 10.5 minutes to process the whole Tsinghua dataset (792 scenes) and 3.2 minutes for the Stanford scene collection (132 scenes). Over an iteration, compactness evaluation (including FCGK computation) takes ~60% of the time, with spectral clustering ~30%, and inexact frequent pattern mining ~5%. Note that the first two parts were both implemented in Matlab and could see significant speed-up if coded in C/C++. Timing is measured on a 4 quad-core 2.80GHz Intel Core CPU with 12GB RAM.

Focal point extraction. Figure 9 shows several clusters and their representative focal points extracted from the Tsinghua collection; the complete set of results for focal extraction can be found in the supplementary material. We can observe hybrid scenes containing multiple focal points, which is fairly typical and results in cluster overlap. Also worth noting is the extraction of non-local focals, which are composed of relatively distant object groups, e.g., {*TV, TV-stand, table, sofa*}, etc. Table 1 gives the number of non-local focals extracted for both datasets. See also the last two rows in Figure 9 for the effect of focal joining.

Iterative clustering. Figure 10 plots how the normalized compactness of the clusters change as the iterative clustering algorithm

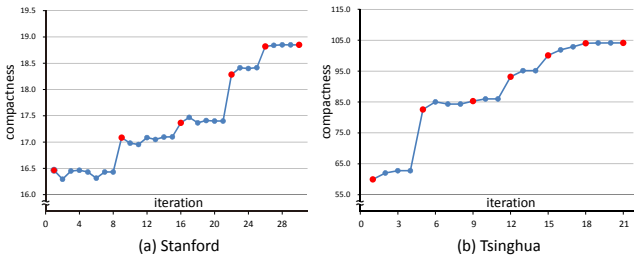

Figure 10: *The plots show the change of the compactness of the clusters obtained as our interleaving optimization progresses, for Stanford and Tsinghua datasets respectively. The red dots represent the switching points from outer loop (mining) to inner loop (clustering). The optimization takes 5 and 6 interleaving iterations to converge on the two datasets, respectively.*

progresses. While the change is not strictly monotone, it is evident that the iteration generally improves cluster quality over time. The final cluster counts for the two sets are 5 and 9, respectively.

Precision-recall on scene retrieval. Figure 11 compares our method to two other methods for scene retrieval:

1. **GK**: Graph kernels of Fisher et al. [2011] to measure similarity between whole scenes. Since we were unable to obtain the authors' code, we coded up our own implementation with two major differences to the original work. First, we use our structural graphs which only encode two types of relationships (support and proximity) and do not consider hierarchical scene graphs. Second, the computation of node and edge kernels are slightly different; see Section 4.2. For both GK and FCGK, the schemes for node and edge kernel estimation and graph kernel normalization, as well as all the parameters, are the same as the original work.

2. **BOW**: A baseline method where we use bag-of-words features on the focal points only as a scene-to-scene similarity.

3. **FCGK (SG)**: On the Tsinghua dataset, we also apply our FCGK similarity on the scenes where as focals, we use the 212 structural groups detected by Xu et al. [2013].

When applying our method, which uses FCGK for scene similarity, we show results in three settings: 1) using the initial set of focals after only one step of frequent pattern mining; 2) using an intermediate set of focals; 3) using the final set of focals extracted.

For the Tsinghua dataset, the ground truth for evaluating scene retrieval is given by the scene labels/categories which come with the dataset. Since this dataset contains many hybrid scenes, we separate it into a subset of simple scenes and the remaining hybrid (complex) scenes and report results on each and their combination. Since the Stanford collection does not come with scene labels, we provide our own labels obtained manually, which, admittedly, could introduce an evaluation bias. A potentially more reliable method, such as voting from multiple users, could be employed.

From the precision-recall curves, we see that our focal-centric similarity based on the final set of focals is the best in all four cases. Moreover, the performance gain is more prominent for hybrid scenes. These results demonstrate not only the merit of utilizing focals for scene comparison but also the merit of our focal extraction scheme, as it seems evident that retrieval performance improves as our iterative algorithm progresses.

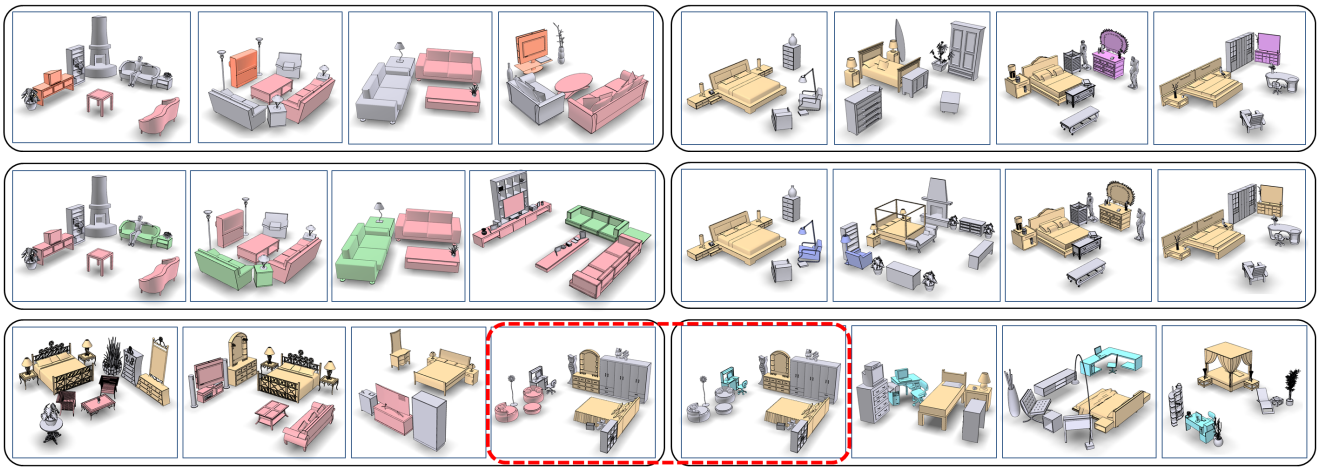

Figure 9: *Several clusters and their representative focals (highlighted in colors) extracted from the Tsinghua scene collection. Top row shows an intermediate result for two clusters and the middle row shows the final result for the relevant clusters. Bottom rows show the final result for other clusters. Note multi-focal hybrid scenes, cluster overlap (marked with the red dashed box), and non-local focal points, such as the combos of {TV, TV-stand, table, sofa} and {bed, nightstands, dresser, mirror} in the last two rows.*

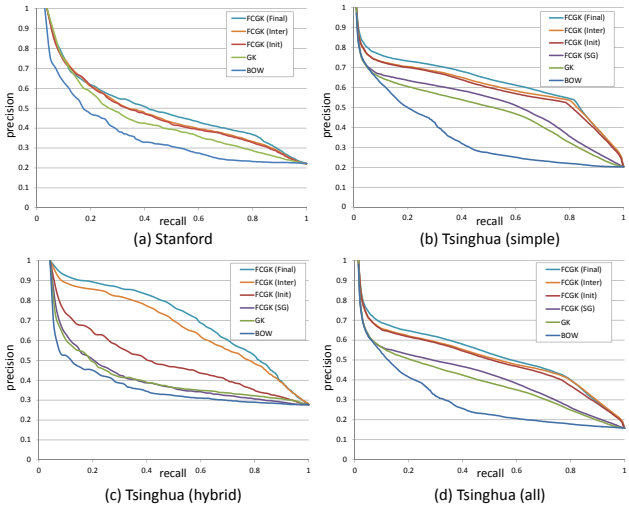

Figure 11: *Precision-recall curves for scene retrieval. (a) Stanford scene collection. (b) Tsinghua collection, simple scenes. (c) Tsinghua, hybrid scenes. (d) Tsinghua, all scenes.*

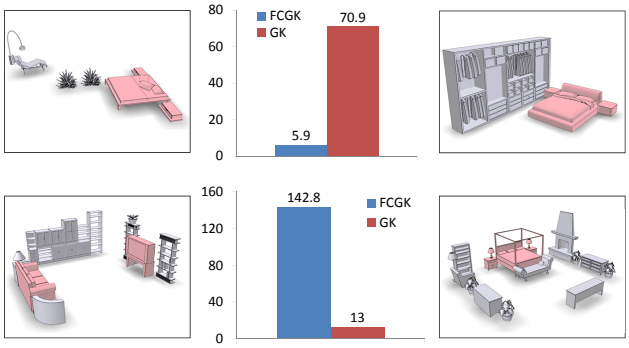

Figure 12: *Comparing GK and FCGK on scene similarity. Top row: two scenes in the same category, but GK returns a large distance between them due to the dissimilar surrounding objects. Bottom row: two scenes belonging to different categories while GK returns a small distance also attributing to surrounding objects, e.g., the nearby bookshelves. In contrast, with a focal-centric view, our method gives more meaningful distances on the two pairs.*

Comparison to GK. Figure 12 shows an explicit comparison between GK and FCGK on scene similarity, attesting to the effectiveness of utilizing focals. In our experiment, we also observed that the matching performance of GK tends to be negatively affected by the presence of many small/trivial objects. For example, when a scene contains a shelf supporting many small objects, GK counts rooted walks from all these objects, which would influence the similarity between more prominent objects. FCGK is more discriminative and trivial objects are less likely to have been chosen as focals.

User evaluation on retrieval. For a hybrid scene, it may be difficult to assign an unambiguous category label. The ground truth used for retrieval on such scenes may be unreliable. Thus instead of relying on scene categories as ground truth, we let human users judge scene similarity based on their prior knowledge. In this second comparative study on scene retrieval, we focus exclusively on retrieval where the query is a hybrid scene. We present a user with 10 queries. For each query, the top return from the three compared methods (GK, BOW and FCGK) are presented to the user and the user is asked to choose which of the three is most similar to the query. We repeat this for a total of 102 queries for the hybrid scenes in the Tsinghua dataset. Against GK, we obtain a winning percentage of **70.2%** and against BOW, we obtain **73.9%**. The results are statistically significant (with $p = 0.01$). In the studies, each scene has been rendered in three random bird's eye views and the images were presented randomly. Among the 43 participants, 80% are computer science researchers, with ages 20 to 50. The rest are frequent computer users with varying backgrounds.

6 Applications

Our scene organization allows classical scene queries and is thus suitable for any application which utilized retrieval results as before, e.g., [Fisher et al. 2012; Xu et al. 2013]. In this section, we

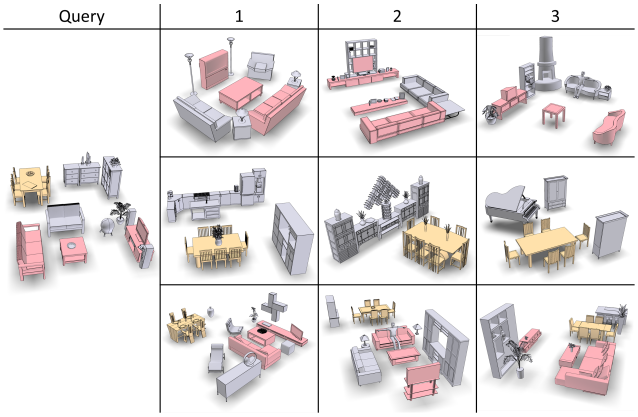

Figure 13: *Comprehensive retrieval takes a query scene and returns scenes grouped by well-matched focals with the query. In each group, the returns are ranked by FCGK based on the corresponding focal. In this example, the query has two focals (colored yellow and red) matched from the scene organization. Three ranked lists of returns corresponding to the two focals (first two rows) and to the* joined *focal (bottom row) are shown.*

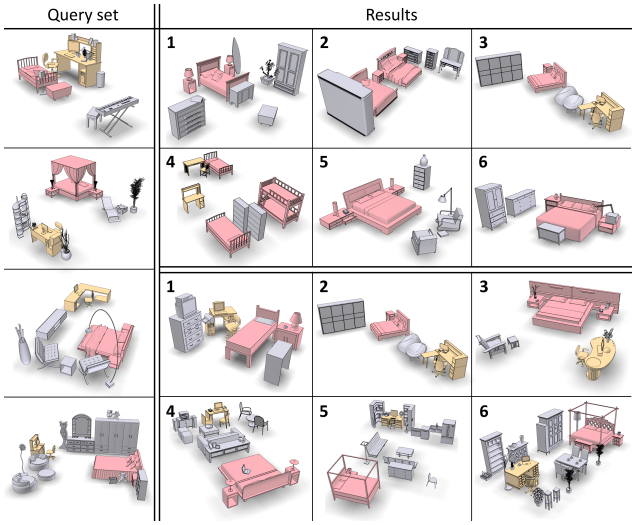

Figure 14: *Multi-query retrieval takes a query set (left) and returns a ranked list of scenes (bottom-right) via focal-based scene comparison. FCGK similarity is used and measured based on focals (colored red and yellow) that well-match frequent substructures in the query set. Returns based on global scene similarity computed by GK are also shown (top-right). To not introduce a bias by coloring of the focals, in the GK returns, we also color any object whose tag matches that of an object in one of the focals.*

discuss several new capabilities afforded by our focal-based data organization for scene retrieval and exploration.

Comprehensive retrieval. In classical retrieval, a single query would fetch a single ranked list of data items. With our focal-centric similarity and pre-computed set of focals, our scene organization supports such classical queries. It also supports part-in-whole type of queries, where the user specifies a region of interest (ROI) in the query scene. This is demonstrated with the exploration tool which we describe below. The interesting new feature enabled by our scene organization is what we call *comprehensive retrieval*. Here the query does not have a specified focal. However, the available focals in the organization are matched with the query scene. Instead of returning a single ranked list of scenes, the comprehensive retrieval returns *multiple* ranked lists, each of which corresponds to a well-matched focal. Figure 13 shows such a result. Note that the vertical order in the table has no clear meaning since the three (horizontal) lists are retrieved based on different sets of focals. If putting all the results together, however, one can expect that those retrieved with multiple focals should be ranked higher since they have more focal substructures receiving higher weights; refer to Equation (9).

For focal-to-scene matching, we utilize the efficient subgraph matching approach described in [Riesen et al. 2010], by which the focal subgraphs are pre-compiled into a hierarchical representation to accelerate the online matching. The average query time is 960ms for the Tsinghua collection and 140ms for the Stanford set.

Multi-query retrieval. In applications such as example-based scene synthesis [Fisher et al. 2012], one may form queries consisting of multiple *semantically related* scenes and wish to retrieve more scenes "of the same". Such *multi-query* retrievals are well-supported by our scene organization. Indeed, since the query scenes are related, they likely share meaningful substructures, making them suitable for focal-based scene comparisons.

Given a query set, we extract frequent substructures from the set and match them against the extracted focals in the scene organization. We then retrieve scenes from the organization using FCGK based on the matched focals. Figure 14 shows one such result with a query set of four hybrid scenes. For comparison, we also show a ranked list of returns based on GK similarity measured against any scene in the query set. As one would expect, the focal-based retrieval produces more discernable results, and more useful results. If the user selected four query scenes all containing a bed-nightstand combo and a desk-chair combo, then it is likely that he/she was seeking scenes that contain similar substructures.

Scene exploration. We develop an exploration tool, based on the extracted focals, which enables a user to browse through a heterogeneous scene collection. Focal points are the primary means for search and navigation. Figure 15 shows the GUI of our tool. The user can select a few focals from the focal point list panel (bottom), and our tool automatically selects a set of scenes sharing similar focals and lists them in the scene list panel (right). The user can browse the list and view the scenes in the main viewer (middle). At any time, the user can click on a selected focal to view its embedding in the current scene. In terms of navigation, as shown in Figure 3, the user can traverse from one scene to another, and one scene cluster to another, through focals which interlink them. The accompanying video contains full sessions of interactive exploration.

In addition, we provide an interface for the user to paint a region of interest (ROI) and search for scenes which contain sub-scenes that are similar to the surroundings of the ROI. When the user selects an ROI in a scene, our system first finds a focal point in the scene which overlaps most with the ROI and adds the focal to the selected list. It then retrieves a new list of scenes based on the updated list of selected focals. Exploring the database with focal points around an ROI, instead of with only the ROI, can provide more relevant results. For example, if the user selects only a chair model as ROI, naive partial matching would simply return all scenes containing a chair. In contrast, our tool searches for scenes sharing the same focal around the chair, returning results that are more context-aware.

Note that the rooted walk graph kernels of Fisher et al. [2011] could also support contextual part-in-whole queries. However, performing subgraph search is likely too time consuming for online retrieval. With pre-analysis resulting a focal-based scene organiza-

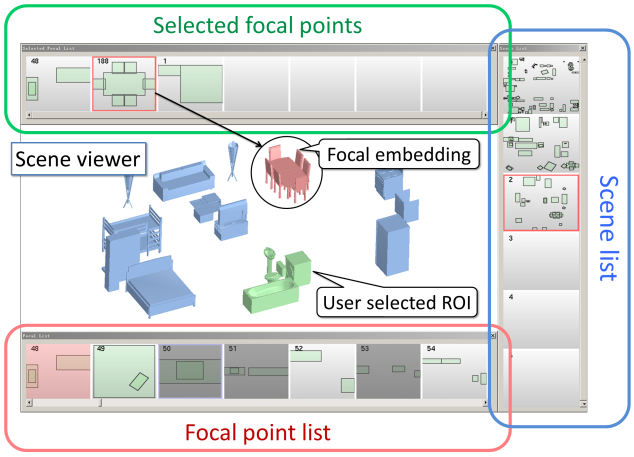

Figure 15: *GUI for our exploration tool is composed of four parts: the focal point list panel (red box), the selected focal list panel (green), the scene list panel (blue), and the main scene viewer. The user can pick a selected focal to view its embedding in the current scene. She can also select a region of interest (ROI) in the viewer to explore more scenes via the focals around the ROI.*

tion, our tool can support efficient context-aware partial matching over a large heterogeneous scene collection.

7 Discussion and future work

At the core of the data organization problem is the mechanism for comparing data. Traditional approaches rely on holistic data views and unique distances defined between data items for grouping or clustering. However, when the data become complex and multi-faceted, a fixed and global view on data similarity can hardly express the rich characteristics in the data.

We advocate the use of focal points for comparing and organizing complex and heterogeneous data and use 3D indoor scenes as a prototype to demonstrate its feasibility and performance gains, e.g., in retrieval. The new approach seems particularly apt at dealing with complex and hybrid scenes. Perhaps its most compelling feature is the ability to process large and heterogeneous collections of scenes and to organize them into an interlinked and well-connected cluster formation, which facilitates scene exploration.

FCGK vs. GK. While our retrieval experiment showed superior performance of FCGK over GK, one should realize that a direct comparison between the two is not exactly fair. GK is a standalone graph similarity measure, where only two graphs to compare are needed. FCGK-based comparison comes with a higher cost as it requires a set of graphs and a co-analysis for focal extraction. That said, if a scene collection is available, we would still suggest using FCGK for its better performance and modest processing costs.

Comparison to structural groups. In our work, a focal point consists of a group of scene objects and it is derived via structural scene analysis. By name alone, this suggests similarity to the structural groups computed by Xu et al. [2013]. There are however major differences. First, their structural groups are *category* groups, while our focals are object groups. More importantly, their group extraction involves only frequent pattern mining through local proximity based search. The latter implies that their method is unlikely to return non-local structural groups. This is in part evidenced by the much higher number of groups (212) they obtain vs. the 34 focals we obtain, on the same scene collection (Tsinghua, 792 scenes). The retrieval results in Figure 11 seem to suggest that non-local focals extracted via mining and clustering provide the better perspectives for meaningful scene comparison.

Non-unique distance. The retrieval experiment using FCGK seems to suggest that our method assigns a unique distance between any two scenes. This is true once the set of focals is fixed and FCGK is to be computed based on those focals and the clustering result. However, the non-uniqueness of focal-centric distances is well utilized in other settings including comprehensive retrieval, multi-query retrieval, and ROI-driven scene exploration, where the relevant focals in the query scenes are all determined on-demand.

Limitations. Our current algorithm depends on semantic labeling of scene objects. It remains to be seen whether it works effectively with noisy or incomplete labels, based on pure geometry analysis. For example, it is interesting to test our method on inputs with various levels of label noise. However, it would be hard to quantitatively evaluate the robustness against noisy labels since it may be difficult to reproduce realistic labeling noise introduced by humans. Nevertheless, the two datasets we used do contain some incorrect labels, which did not seem to affect the overall performance. There are perhaps more than a desirable number of parameters in the algorithm, whose values were determined experimentally. From a technical stand point, improvements are possible in various components of the algorithm. For example, our layout similarity operates on OBBs only, which may be unsuitable for objects with complex geometry and spatial arrangements. The structural graphs model the scenes only as flat arrangements of objects. Hierarchical organization may be potentially advantageous.

Future work. One obvious pursuit is to apply our focal-driven approach to other datasets, e.g., large and heterogeneous collections of annotated images. An interesting technical question is whether our scene organization can be updated with an additional set of scenes without recomputing everything. Also, rather than replacing one object at a time for scene synthesis like in previous works, our scene organization and focal-based partial scene retrieval, may allow for substituting sub-scenes for the synthesis task.

We conclude the paper with a question: "what is the best way to compare complex scenes?" This work, along with others before it, assume that comparing attributed graphs defined by semantic tags and object arrangements is the best way. However, we observe that visually, many retrieval results do not look so compelling even with the best method to date. If one takes away the colorings in Figure 14, then the contrast between GK and FCGK would not be as salient. Hence, the focal-centric view we advocate offers a perspective worth considering. The general question, also one that is attributed to complex data beyond those of indoor scenes, should perhaps be answered with user and application intent in mind.

Acknowledgments. We thank all the reviewers for their comments and feedback. We are grateful to the authors of [Fisher et al. 2011] and [Xu et al. 2013] for providing their datasets. This work was supported in part by NSFC (61202333, 61379090 and 61272327), NSERC Canada (611370), National 863 Program of China (2012AA011802), Shenzhen Innovation Program (CXB201104220029A, KQCX20120807104901791, JSGG20130624154940238, JCYJ20120617114842361), CPSF China (2012M520392), and Israel Science Foundation.

References

BIBERMAN, Y. 1994. A context similarity measure. *Machine Learning 784*, 49–63.

CHENG, M.-M., MITRA, N. J., HUANG, X., AND HU, S.-M. 2014. SalientShape: group saliency in image collections. *The Visual Computer 30*, 4, 443–453.

DOERSCH, C., SINGH, S., GUPTA, A., SIVIC, J., AND EFROS, A. A. 2012. What makes paris look like Paris? *ACM Trans. on Graph (Proc. of SIGGRAPH) 31*, 4, 101:1–9.

FISHER, M., AND HANRAHAN, P. 2010. Context-based search for 3D models. *ACM Trans. on Graph (Proc. of SIGGRAPH Asia) 29*, 6, 182:1–10.

FISHER, M., SAVVA, M., AND HANRAHAN, P. 2011. Characterizing structural relationships in scenes using graph kernels. *ACM Trans. on Graph (Proc. of SIGGRAPH) 30*, 4, 34:1–11.

FISHER, M., RITCHIE, D., SAVVA, M., FUNKHOUSER, T., AND HANRAHAN, P. 2012. Example-based synthesis of 3D object arrangements. *ACM Trans. on Graph 31*, 6, 135:1–11.

HAN, J., CHENG, H., XIN, D., AND YAN, X. 2007. Frequent pattern mining: current status and future directions. *Data Mining and Knowledge Discovery 15*, 1, 55–86.

HUANG, Q., ZHANG, G., GAO, L., HU, S., BUSTCHER, A., AND GUIBAS, L. 2012. An optimization approach for extracting and encoding consistent maps in a shape collection. *ACM Trans. on Graph (Proc. of SIGGRAPH Asia) 31*, 6, 167:1–11.

HUANG, Q., SU, H., AND GUIBAS, L. 2013. Fine-grained semi-supervised labeling of large shape collections. *ACM Trans. on Graph (Proc. of SIGGRAPH Asia) 32*, 6, 190:1–10.

HUANG, S.-S., SHAMIR, A., SHEN, C.-H., ZHANG, H., SHEFFER, A., HU, S.-M., AND COHEN-OR, D. 2013. Qualitative organization of collections of shapes via quartet analysis. *ACM Trans. on Graph (Proc. of SIGGRAPH) 32*, 4, 71:1–10.

JAIN, A., THORMÄHLEN, T., RITSCHEL, T., AND SEIDEL, H.-P. 2012. Exploring shape variations by 3D-model decomposition and part-based recombination. *Computer Graphics Forum (Special Issue of Eurographics) 31*, 2, 631–640.

JEH, G., AND WIDOM, J. 2002. SimRank: a measure of structural-context similarity. In *Proc. of ACM SIGKDD*, 538–543.

JUNEJA, M., VEDALDI, A., JAWAHAR, C. V., AND ZISSERMAN, A. 2013. Blocks that shout: Distinctive parts for scene classification. In *Proc. IEEE Conf. on Comp. Vis. and Pat. Rec.*, 923–930.

KIM, V. G., LI, W., MITRA, N., DIVERDI, S., AND FUNKHOUSER, T. 2012. Exploring collections of 3D models using fuzzy correspondences. *ACM Trans. on Graph (Proc. of SIGGRAPH) 31*, 54:1–11.

OVSJANIKOV, M., LI, W., GUIBAS, L., AND MITRA, N. J. 2011. Exploration of continuous variability in collections of 3D shapes. *ACM Trans. on Graph (Proc. of SIGGRAPH) 30*, 4, 33:1–10.

QUATTONI, A., AND TORRALBA, A. 2009. Recognizing indoor scenes. In *Proc. IEEE Conf. on Comp. Vis. and Pat. Rec.*, 413–420.

RASIWASIA, N., AND VASCONCELOS, N. 2008. Scene classification with low-dimensional semantic spaces and weak supervision. In *Proc. IEEE Conf. on Comp. Vis. and Pat. Rec.*, 1–6.

RIESEN, K., JIANG, X., AND BUNKE, H. 2010. Exact and inexact graph matching: Methodology and applications. *Managing and Mining Graph Data 40*, 217–247.

ROSCH, E. 1975. Cognitive reference points. *Cognitive Psychology 7*, 4, 532–547.

SHAPIRA, L., SHALOM, S., SHAMIR, A., COHEN-OR, D., AND ZHANG, H. 2009. Contextual part analogies in 3D objects. *Int. J. Comp. Vis. 89*, 2-3, 309–326.

SHILANE, P., AND FUNKHOUSER, T. 2007. Distinctive regions of 3D surfaces. *ACM Trans. on Graph 26*, 2, 7:1–15.

SINGH, S., GUPTA, A., AND EFROS, A. 2012. Unsupervised discovery of mid-level discriminative patches. In *Proc. Euro. Conf. on Comp. Vis.*, 73–86.

TSUDA, K., AND KUDO, T. 2006. Clustering graphs by weighted substructure mining. In *Proc. Intl Conf on Machine Learning (ICML)*, 953–960.

TVERSKY, A. 1977. Features of similarity. *Psychological Review 84*, 4, 327–352.

VAN KAICK, O., XU, K., ZHANG, H., WANG, Y., SUN, S., SHAMIR, A., AND COHEN-OR, D. 2013. Co-hierarchical analysis of shape structures. *ACM Trans. on Graph (Proc. of SIGGRAPH) 32*, 4, 69:1–10.

VIDAL, R. 2011. Subspace clustering. *IEEE Signal Processing Magazine 28*, 3, 52–68.

WANG, S., YUAN, X., YAO, T., YAN, S., AND SHEN, J. 2011. Efficient subspace segmentation via quadratic programming. In *AAAI*, 519–524.

WANG, Y., XU, K., LI, J., ZHANG, H., SHAMIR, A., LIU, L., CHENG, Z., AND XIONG, Y. 2011. Symmetry hierarchy of man-made objects. *Computer Graphics Forum (Special Issue of Eurographics) 30*, 2, 287–296.

WITTGENSTEIN, L. 1953. *Philosophical investigations*. New York: Macmillan.

XU, K., ZHANG, H., COHEN-OR, D., AND CHEN, B. 2012. Fit and diverse: Set evolution for inspiring 3D shape galleries. *ACM Trans. on Graph (Proc. of SIGGRAPH) 31*, 4, 57:1–10.

XU, K., CHEN, K., FU, H., SUN, W.-L., AND HU, S.-M. 2013. Sketch2Scene: Sketch-based co-retrieval and co-placement of 3D models. *ACM Transactions on Graphics 32*, 4, 123:1–10.

YAN, X., AND HAN, J. 2002. gSpan: graph-based substructure pattern mining. In *Proc. Int. Conf. on Data Mining*, 721–724.

ZELNIK-MANOR, L., AND PERONA, P. 2004. Self-tuning spectral clustering. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, vol. 17, 1601–1608.

ZHAO, X., WANG, H., AND KOMURA, T. 2014. Indexing 3d scenes using the interaction bisector surface. *ACM Trans. on Graph*, to appear.

ZHENG, Y., COHEN-OR, D., AND MITRA, N. J. 2013. Smart variations: Functional substructures for part compatibility. *Computer Graphics Forum (Special Issue of Eurographics) 32*, 2, 195–204.