Automatic Discovery of Groups of Objects for Scene Understanding

Congcong Li
Cornell University
cl758@cornell.edu

Devi Parikh
Toyota Technological Institute (Chicago)
dparikh@ttic.edu

Tsuhan Chen
Cornell University
tsuhan@ece.cornell.edu

Abstract

Objects in scenes interact with each other in complex ways. A key observation is that these interactions manifest themselves as predictable visual patterns in the image. Discovering and detecting these structured patterns is an important step towards deeper scene understanding. It goes beyond using either individual objects or the scene as a whole as the semantic unit. In this work, we promote "groups of objects". They are high-order composites of objects that demonstrate consistent spatial, scale, and viewpoint interactions with each other. These groups of objects are likely to correspond to a specific layout of the scene. They can thus provide cues for the scene category and can also prime the likely locations of other objects in the scene.

It is not feasible to manually generate a list of all possible groupings of objects we find in our visual world. Hence, we propose an algorithm that automatically discovers groups of arbitrary numbers of participating objects from a collection of images labeled with object categories. Our approach builds a 4-dimensional transform space of location, scale and viewpoint, and efficiently identifies all recurring compositions of objects across images. We then model the discovered groups of objects using the deformable parts-based model. Our experiments on a variety of datasets show that using groups of objects can significantly boost the performance of object detection and scene categorization.

1. Introduction

If we were to describe the image shown in Figure 1(a), we would perhaps say it is "an outdoor seating area with three sets of picnic-umbrella, table and chairs". Note that this description demonstrates a natural grouping of objects in the scene. This is in contrast with existing trends in computer vision of treating individual objects (or the entire scene as a whole) as the basic unit of semantics. This is not natural: it is unlikely that we would describe the scene as having "three picnic-umbrellas, three tables and nine chairs". This is because objects in scenes interact with each other in complex ways, and arguably, these interactions are what tell the story of the scene. These interactions may be of various forms such as spatial relationships, physical support, actions being performed by a subject on an object, etc. But the key observation is that all these interactions manifest themselves as predictable visual patterns in the image. While characterizing these different interactions would be valuable, simply discovering and detecting these structured visual patterns themselves is an important step towards deeper scene understanding.

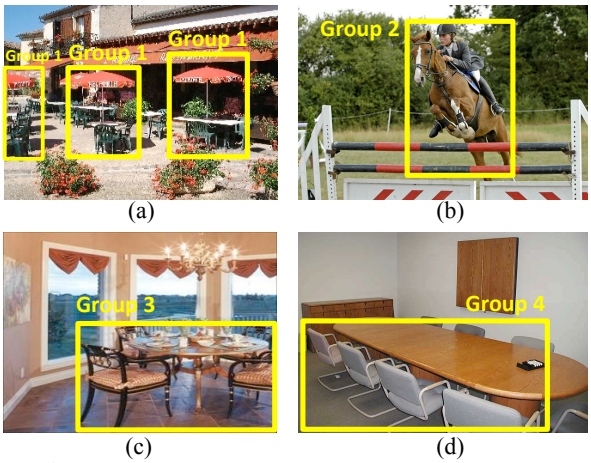

Figure 1. We *automatically* discover and model "groups of objects" which are complex composites of objects with consistent spatial, scale, and viewpoint relationship across images. These groups can aid detection of participating objects (*e.g.* umbrella in (a)) or non-participating objects (*e.g.* fence in (b)) as well as improve scene recognition (*e.g.* dining room vs. meeting room in (c) and (d)).

In this work, we promote "groups of objects". They are complex composites of two or more objects which have consistent spatial, scale, and viewpoint relationships with each other across images. Because of this consistency, they are likely to be more detectable than the participating objects in isolation which may demonstrate more intra-class appearance variance across images. Hence, detecting the groups of objects can help improve detection of the participating objects. For instance, the group shown in Figure 1(a) significantly boosts the performance of an umbrella detector. Even beyond that, groups of objects are likely to correspond to a specific layout of the scene. They can thus provide strong contextual cues for where *other* objects in the scene are likely to be present. For instance, a group captur-

ing a person on a jumping-horse as seen in Figure 1(b) can aid the detection of a fence. Moreover, groups of objects can also better discriminate among scenes that share similar participating objects, but in different configurations such as dining room and meeting room in Figure 1(c) and (d).

Groups of objects clearly have potential for aiding various visual recognition tasks. But where do these groups of objects come from? It is not feasible to manually compile a list of all groups of objects with arbitrary numbers of participating objects that we see in the wide variety of scenes in the visual world around us. On the bright side, what the advent of crowd-sourcing services and visual media on the web has given us is large datasets such as PASCAL [5] and SUN [29] that contain many natural images richly annotated with object categories.

In this work, we *automatically* and efficiently discover a complete and compact set of object groups containing arbitrary numbers of participating objects. We leverage images annotated with object categories to do so. We build a 4-dimensional transform space modeling spatial location, scale and viewpoint of objects. Objects demonstrating consistent interactions along these dimensions across images are mapped to the same region in this transform space. This space is efficiently mined to discover recurring groups of objects. We model these groups of objects via the deformable part-based model, allowing us to detect these groups in novel images. These detections can now be used as contextual cues for participating or non-participating individual object detection, or for scene categorization.

Our contributions are as follows: First, we propose modeling a full spectrum of arbitrarily high-order object interactions for deeper scene understanding. These groups contain objects with consistent spatial, scale and viewpoint relationships between each other. Secondly, we propose an algorithm to automatically discover these groups from images annotated only with object labels. We then model the groups using the existing deformable part-based object models. Finally, we demonstrate on a variety of datasets that group detections can significantly improve object detection and scene recognition performance. We also show that our discovered groups are semantically meaningful.

2. Related Work

We compare and contrast our work to several existing works that exploit object interactions, model visual composites of scenes, or discover co-occurring visual patterns.

Object interactions: Many works exploit contextual interactions between objects [1–4, 6, 8–10, 12, 16, 19, 22, 23, 26, 27, 30] for improved recognition. Most of these works only model pair-wise interactions among objects. Even works that go beyond pair-wise interactions (*e.g.* Felzenszwalb et al. [6]) typically rely on individual object detections as the source of context. With groups of objects, we can also capture higher-order contextual interactions. Furthermore, we model the visual appearance of the groups of objects as a whole, resulting in a more reliable contextual signal.

Visual composites: Several works have explored entities that fall between individual objects and scenes. In some works [15, 18, 28] these entities are discovered from unlabeled regions in images. These entities are hence heavily influenced by the particular choice of appearance models used in the discovery process, and are seldom semantically meaningful. We discover groups of objects by exploiting object-level annotations in images. Our groups tend to be semantically meaningful, and are not dependent on the appearance modeling choices that follow. At the other extreme, some works employ a fully supervised approach to learn visual composites. For instance, Xiao et al. [29] label images with 'subscenes'. Sadeghi *et al*. [25] label a subset of the PASCAL dataset with 'visual phrases' which are either objects performing an action (i.e. objects in a certain pose such as person running), or a pair of objects interacting with each other (*e.g*. person riding a horse). They rely on a manual list of 17 visual phrases, and are restricted to groups containing at most two objects from a set of 8 categories. Our work on the other hand automatically discovers groups containing an arbitrary number of objects. As we show in our experiments, we can discover 71 groups containing upto 6 objects from 107 object categories in the SUN dataset that contains images from a wide variety of scene categories.

Finding co-occurring patterns: Several works have looked at the problem of discovering co-occurring patterns across images: be it for discovering hierarchical spatial patterns of visual words in images [20] or discovering segments of foreground objects of interest [11, 14, 24]. In our work, we are interested in finding groupings of objects that consistently co-occur at predictable locations, scales and viewpoints with respect to each other. Zhang *et al*. [31] propose an efficient algorithm for calculating kernels capturing similarity between pairs of images using translation invariant arbitrarily higher-order visual code-word arrangements. We employ a similar Hough-transform like mechanism, but apply to the novel task of discovering groups of objects from images annotated with object categories. We extend their proposed translation based "offset space" to a more complex transform space that also incorporates scale and viewpoint. We also propose a soft voting scheme to be robust to quantization artifacts in this transform space.

3. Approach

We first describe the desirable properties of groups of objects, and then present our approach to discover them.

3.1. Groups of objects

A group of objects contains two or more objects. For objects to belong to the same group, they must co-occur frequently, and have consistent spatial, scale and

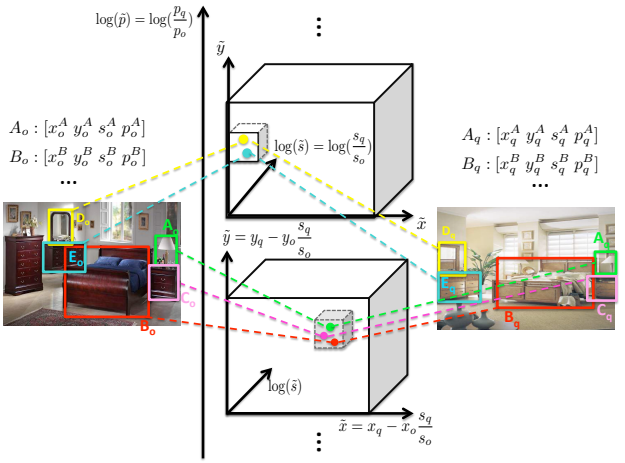

Figure 2. Our algorithm for finding high-order recurring patterns of objects utilizes a 4-D transform space. In this example, we note that the three correspondences of objects $(A_o, A_q), (B_o, B_q), (C_o, C_q)$ fall in the same bin in the transform space, thus the objects (A_o, B_o, C_o) and (A_q, B_q, C_q) form a 3^{rd}-order pattern. Similarly, (D_o, E_o) and (D_q, E_q) form a 2^{nd}-order pattern.

pose/viewpoint relationships across images. Each object category may participate in multiple groups, and multiple instances of the same object category may participate in the same group. For instance, a table with four chairs arranged around the table may form one group, while a table with two chairs and an umbrella may form another group. Hence, any naive clustering of categories based on co-occurrence or location/scale/viewpoint consistency would not suffice for our purposes. We propose the following approach to discovering a complete set of groups from images annotated with object bounding boxes.

3.2. Group Discovery Algorithm

Based on our above definition, the task of discovering groups becomes that of discovering consistently occurring object-layout patterns in a set of images. Our intuition is that if two object-layout patterns belong to the same group, they not only contain the same participating object categories, but the objects share similar transformations (in location, scale and viewpoint) that map them from one pattern to the other. For example, in Figure 2, it is clear that the object-layout pattern $[A, B, C]$ repeats itself in both images. We know this because the displacement in the location of A between the two images, is the same for B and C. All three translate the same amount between the two images. So if we look at a transform space that encodes how much an object translates from one image to the other, A, B and C would fall at the same location in the transform space. This forms the intuition behind our approach, except we deal with not only translation, but also scale and viewpoint changes.

Let's first consider a dataset with only two images annotated with object bounding boxes. Let's say the images have a set of objects O and Q respectively. For every object $o \in O$, we are given its category $c(o)$, location $\big(x(o), y(o)\big)$, scale $s(o)$, and viewpoint $p(o)$, where $\big(x(o), y(o)\big)$ is computed as the coordinates of the center of the object bounding-box, $s(o)$ is computed as the square-root of the box area, and $p(o)$ is computed as the aspect ratio of the box. Similarly, for any object $q \in Q$, we have $c(q)$, $\big(x(q), y(q)\big)$, $s(q)$ and $p(q)$.

Now we want to find any co-occurring object-layout patterns between these two images. To do so, we first identify a set of object correspondences $R = \{(o, q) \in O \times Q : c(o) = c(q)\}$. Note that this is a many-to-many mapping: an object in O may correspond to multiple objects in Q, and vice versa. For each correspondence $r \in R$, we construct a transform that describes the location, scale, and viewpoint changes that this correspondence induces: $\mathcal{T}(r) = [\tilde{x}(r), \tilde{y}(r), \tilde{s}(r), \tilde{p}(r)]$. Here $(\tilde{x}(r), \tilde{y}(r))$ denotes the translation of object location, i.e. $\tilde{x}(r) = x(q) - x(o)\frac{s(q)}{s(o)}$ and $\tilde{y}(r) = y(q) - y(o)\frac{s(q)}{s(o)}$, where the factor $\frac{s(q)}{s(o)}$ is used to normalize the translation by the object size[1]. $\tilde{s}(r)$ denotes the scale change, i.e. $\tilde{s}(r) = \frac{s(q)}{s(o)}$ and $\tilde{p}(r)$ denotes the viewpoint change $\tilde{p}(r) = \frac{p(q)}{p(o)}$. This results in a 4-D transform space, as shown in Figure 2. To allow for small variance, we quantize the space into discrete bins.

If we have a set of object correspondences $r_1, \cdots, r_n$ where $r_i = (o_i, q_i)$, that fall in the same bin of the transform space, i.e. share the same transform $\mathcal{T}(r_1) = \cdots = \mathcal{T}(r_n)$, we say that $(o_1, \cdots, o_n)$ and $(q_1, \cdots, q_n)$ form an n^{th}-order object-layout pattern. We represent a pattern via its two instantiations, i.e. $\text{Pa} = \{(o_1, \cdots, o_n), (q_1, \cdots, q_n)\}$.

Note that there may be multiple bins in the transform space that have more than one object in them. For example, in Figure 2, we find that, besides the $[A, B, C]$ pattern, $[D, E]$ is also a repeating pattern. Hence between two images, we may have a set of patterns $\text{Pa}_1, \text{Pa}_2, \cdots, \text{Pa}_K$. Naively, one may consider each pattern to be a group of objects. However, note that multiple patterns may correspond to the same group structure (i.e. the participating objects with the same location, scale, viewpoint relationship). For example, assume we find two patterns between the two images: $\text{Pa}_1 = \{(o_1, o_2, o_3), (q_1, q_2, q_3)\}$ and $\text{Pa}_2 = \{(o_1, o_2, o_3), (q_4, q_5, q_6)\}$, as shown in Figure. 3(a). The repetition of (o_1, o_2, o_3) in both patterns indicates that (o_1, o_2, o_3), (q_1, q_2, q_3) and (q_4, q_5, q_6) are all instantiations of the same group. Hence we employ a straightforward clustering algorithm to cluster all the discovered patterns to generate a set of groups $\mathcal{G}$. Our algorithm is described in Algorithm 1.

To extend the above approach to a dataset with multiple images, we first find all the patterns between every pair of images. We then cluster the patterns based on the transitivity of patterns across images, as shown in Figure 3(b) where Pa_1, Pa_3, Pa_3 should all belong to the same group.

[1] We use the scale of the object as a proxy for estimating the global scale of the scene.

Algorithm 1. Generate groups

$G_1 \leftarrow \text{Pa}_1$, $\mathcal{G} \leftarrow \{G_1\}$
$n_G \leftarrow 1$
for $k = 1 : K$ **do**
  $flag \leftarrow 0$
  for $j = 1 : n_G$ **do**
    if $\text{Pa}_k \cap G_j \neq \emptyset$ **then**
      $G_j \leftarrow G_j \cup \text{Pa}_k$
      $flag \leftarrow 1$
      break
    end if
  end for
  if $flag == 0$ **then**
    $n_G \leftarrow n_G + 1$
    $G_{n_G} \leftarrow \text{Pa}_k$
    $\mathcal{G} \leftarrow \mathcal{G} \cup \{G_{n_G}\}$
  end if
end for

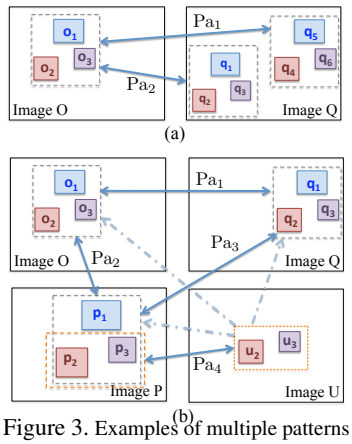

Figure 3. Examples of multiple patterns to be combined into a group.

We utilize the same Algorithm 1 to find the groups.

3.3. Soft Voting

There are still two remaining concerns: (1) The above approach as described is sensitive to the quantization of the transform space; (2) Insisting that all participating objects in a group should be present in every instantiation of the group in images is not realistic. Not only does this reduce the instantiations of groups with many objects, it also results in the clustering algorithm discovering many similar and redundant groups. To address these problems, we propose a soft voting scheme for group discovery.

First, to alleviate the effect of hard quantization, instead of each object correspondence falling in only one bin in the transform space (as described above), we allow it to vote for neighboring bins weighted by a 4-D gaussian filter with a standard-deviation of 1, indicated by the circles surrounding the object correspondences in Figure 4. Note that we show the 2-D transform space $(\tilde{x}, \tilde{y})$ only for ease of illustration. Our implementation uses a 4-D transform space. For each bin, we accumulate the soft votes from all object correspondences, as shown in the heat map in Figure 4. We then use non-maximum suppression to find the locations of the peaks. Each object correspondence is assigned to the peak it contributes to the most. A peak that gets n object assignments corresponds to a n^{th}-order pattern. After finding the co-occuring patterns, we apply the same clustering Algorithm 1 to find groups.

Secondly, to deal with the issue of missing participating objects, we employ a post-processing scheme that allows lower-order group instantiations to be merged with instantiations of corresponding higher-order groups. We do so if only one participating object in the high-order group is missing. For example, in Figure 3(b), pattern Pa_4 is an instantiation of a 2^{nd}-order group and patterns $\text{Pa}_1, \text{Pa}_2, \text{Pa}_3$ are instantiations of a 3^{rd}-order group. Since (p_2, p_3) in the 2^{nd}-order group also participates in (p_1, p_2, p_3) in the 3^{rd}-order group, we absorb (u_2, u_3) into instantiations of the 3^{rd}-order group (but with one participating object missing). Note that (u_2, u_3) is no longer considered to be an instantiation of the 2^{nd}-order group.

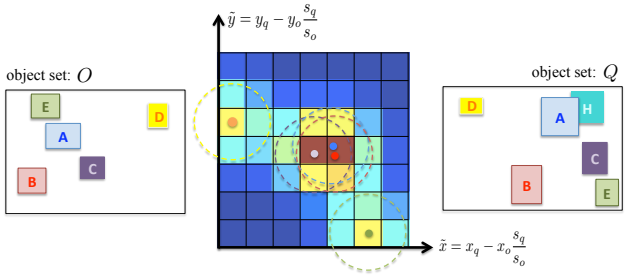

Figure 4. An example of object correspondences voting for neighboring bins in the 2-D transform space. The heat map depicts the accumulated soft votes in each bin.

For a general case, if an n^{th} order group has instantiations where n–1 of the participating objects also participate in an $(n–1)^{th}$ order group, we let the n^{th} order group absorb the instantiations of that $(n–1)^{th}$ order group. We perform this process sequentially on all groups with more than 2 objects, starting with the highest-order groups. We then compute the frequency with which each participating object is present in the group instantiations. We prune objects that participate less than 50% of the time, effectively reducing the order of the group. If the resultant lower-order group already exists, the instantiations are merged. Finally, we only keep groups with more than 30 instantiations in the training data in order to have enough positive samples for training group models as described in Section 3.5.

3.4. Computational Efficiency

Our approach is quite efficient. The computational cost of finding *all* co-occurring patterns of an *arbitrary* order between a pair of images is linear in the number of object correspondences. Note that these can at most be quadratic in the number of objects in both images if *all* objects within both images are the same category. In practice, the number of object correspondences between two images is small and often less than the number of objects in each image. Our matlab implementation takes less than $3ms$ to find all co-occurring patterns in a pair of densely labeled images on a Macintosh machine with 2.66GHz CPU. Finding all patterns from all pairs of 1434 training images in UIUC phrase dataset *sequentially* took about 50 minutes (obviously, this process is highly parallelizable). Clustering these patterns into groups (including soft-voting) took another 20 minutes.

3.5. Group Detection

We model the appearance of groups of objects via object models similar to [15, 18, 25]. This allows us to utilize any off-the-shelf object detector to detect groups. In our experiments we use the deformable part-based model [6]. Specifically, we use the code made available at [7] with default parameter settings to train 4-component group detectors. We now describe how we generate the bounding boxes to train our groups-of-objects detectors.

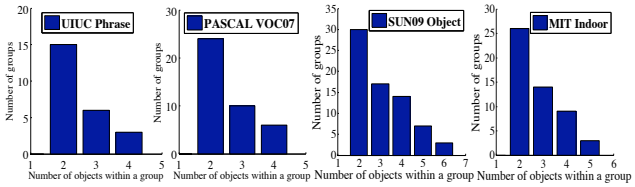

Figure 5. Distribution of the number of objects within our automatically discovered groups of objects using four datasets. We can discover a diverse set of high-order groups.

We generate a bounding box for each instantiation of the groups in the images. If the instantiation has all the participating objects, we generate a bounding box that is the smallest box that encompasses all participating objects. Note that using all instantiations of a group across the dataset, we can estimate the mean location, scale and viewpoint of any participating object with respect to the group. So if any of participating objects are missing we hallucinate the the missing object using these statistics. We then generate a box that encompasses all objects (including the hallucinated one).

4. Experiments and Results

4.1. Auto-Discovery of Object Groups

We perform the group discovery on four datasets: UIUC phrase dataset [25], Pascal VOC 2007 dataset [5], SUN09 object dataset [2] and MIT indoor dataset [21]. Examples of object groups discovered by our algorithm for various datasets are given in Figure 6. Figure 5 also provides the histograms over the object numbers within a group.

UIUC phrase dataset is a subset of the PASCAL dataset. It contains 2769 images labeled with 8 of the 20 PASCAL categories. In addition to the object category annotations, it contains bounding box annotations for a manually generated list of 17 phrases. 12 of these phrases describe interactions between two objects (e.g. person riding horse) and 5 describe a single object performing an action (e.g. dog running). Since the goal of our work is to model groups of more than one objects, we focus on the 12 phrases. Our algorithm discovers 24 groups in this dataset. Our groups contain 2 to 4 objects, as shown in Figure 5.

We wish to evaluate how well our automatically discovered groups correspond to the hand-generated list of 12 groups containing two objects. To determine if one of our groups 'matches' one of the phrases in the dataset, we compare the bounding boxes automatically generated by our approach for that group, to the hand-annotated bounding boxes for the phrase. If more than 75% of our bounding boxes have more than 50% intersection-over-union overlap with the hand annotated bounding boxes, we assign our group to that phrase. Note that each group can match only one phrase, but multiple groups can match the same phrase. We find that every phrase has at least one matched group. However, if we use a lower dimensional transform space (e.g. $(\tilde{x}, \tilde{y})$ or $(\tilde{x}, \tilde{y}, \tilde{s})$), different phrases (e.g. 'person riding horse' and 'horse and rider jumping') would be grouped together. Apart from phrases, our groups also capture other concepts such as two-people on a sofa. We merge the bounding boxes from all groups that match the same phrase. We now have a one-to-one correspondence between the phrases and the matched groups. In Table 1 we report the percentage of the phrase bounding boxes covered by our groups. We find a large proportion of the hand annotated bounding boxes have been discovered by our automatic approach. We train detectors for detecting the manually labeled phrases, but using our automatically discovered group bounding boxes. As seen in Table 1, the performance is comparable to and sometimes even superior to training a detector using the manually labeled bounding boxes! We use the same test settings as in [25]: roughly 50 positive and 150 negative images. This confirms that our approach can find semantically meaningful groups in an automatic manner.

Phrase Names	Ratio covered by groups	AP (trained by manual labels) [25]	AP (trained by discovered groups)
Person next to bicycle	81.7%	46.6	43.5
Person lying on sofa	72.9%	24.9	25.2
Horse and rider jumping	80.0%	87.0	86.5
Person drinking from bottle	91.7%	27.9	30.3
Person sitting on sofa	69.1%	26.2	24.8
Person riding horse	77.7%	78.7	77.3
Person riding bicycle	82.3%	66.9	66.1
Person next to car	64.2%	44.3	41.2
Dog lying on sofa	85.1%	23.5	25.5
Bicycle next to car	84.0%	44.8	49.6
Person sitting on chair	95.2%	20.1	21.5
Person next to horse	68.2%	35.1	34.5
MEAN	79.3%	43.8	43.8

Table 1. Column 1: the ratio of training examples for a phrase covered by our corresponding groups. Column 2: detection performance of detectors trained using manually labeled phrase bounding boxes in [25]. Column 3: detection performance of detectors trained using our automatically discovered group bounding boxes. Our automatically discovered groups match the manually annotated phrases very well. (APs measured in %.)

PASCAL VOC 2007 dataset contains 9963 images with annotations for 20 object categories. We discover 40 groups containing 2 to 4 objects (Figure 5). We use these groups as contextual cues for improving object detection performance (Section 4.2). **SUN09 object dataset** contains 12059 images annotated with 107 object categories, the largest dataset of its kind. Our algorithm discovers 71 groups containing 2 to 6 objects (Figure 5). Again, we use these groups as contextual cues for improving object detection (Section 4.2). **MIT indoor dataset** contains 15613 images from 67 scene categories and 423 labeled object categories. Since only 2743 images in the dataset have object annotations, we select 15 categories that have more than 50 training images annotated, as listed in Table 5-Row 1. We utilize 152 object categories present more than 20 times in images of the 15 scene categories. We discover 52 groups containing 2 to 6 objects. We use these groups to improve scene recognition performance as described in Section 4.3.

4.2. Object Detection

We use the deformable part-based model [6] to train detectors for all the individual objects of interest (OOI) and the groups. We use the contextual re-scoring scheme used by Felzenszwalb *et al.* [6]. We re-score a candidate OOI

Figure 6. Examples of our automatically discovered groups of objects from four datasets. Instantiations of the same group depict the same objects with consistent spatial, scale, and viewpoint relationships. They often have the same semantic meaning. At the 4^{th} column of the 4^{th} row, we also show a failure case where the instantiations do not have the same semantic meaning. This is because objects interact with each other in complex ways, which may not always be captured by our 4-dimensional transform.

detection using a classifier that incorporates the highest detections of groups of objects in the image. We evaluate the resultant improvements in object detection performance on three datasets: UIUC phrase dataset, PASCAL VOC 2007 dataset, and SUN09 object dataset.

UIUC phrase dataset. Table 2 compares the object detection improvement by using our automatically discovered object groups as contextual cues, as opposed to using detectors for the manually defined phrases [25] as context. We also compare with using other individual object categories as contextual information as in [6]. The same contextual re-scoring scheme is used for all approaches.

We see that using our automatically discovered groups outperforms the other two methods in 5 out of 8 categories and performs comparably for the remaining 3 categories. There are two main reasons for our approach having better performance: (1) Unlike the phrases [25], our groups contain more than just 2 objects. For instance, we have a group composed of two horses and two persons, and another group containing four persons. (2) Our groups explicitly model the spatial and scale relationship between objects, thus resulting in more robust appearance models themselves. For instance, in Figure 6, we have two groups both containing a person and a bottle, but with different spatial interactions. These are lumped together in [25] as "person drinking bottle" resulting in large intra-class variance in appearance.

PASCAL VOC 2007 dataset. Table 3 shows the results of using different types of contextual information to improve the detection performance of objects. We compare our method with the baseline method without using context, the method of using other objects as context [6] and the state-of-the-art method of using contextual-meta-objects (CMO) recently proposed by Li *et al*. [15] as context. CMOs are contextually relevant regions for each object category that are automatically discovered by using that object as an anchor point and exploiting surrounding unlabeled regions. Our groups on the other hand have access to more annotations but are unable to leverage unlabeled regions that may be consistent. Hence CMOs and our groups can be viewed as being complementary sources of context. Overall, using the discovered groups as context achieves better average performance across the 20 object categories over the other two methods. Furthermore, combining these various contextual cues further boosts performance.

	bike	bottle	car	chair	dog	horse	pers	sofa	MEAN
Base w/o context [6]	57.0	7.0	25.8	11.1	5.6	49.3	25.7	14.1	24.5
Object context ([6])	58.8	9.3	**33.1**	13.4	5.0	53.7	27.9	19.8	27.6
Phrase context ([25])	60.0	9.3	32.6	**13.6**	**8.0**	53.5	28.8	22.5	28.5
Group context	**63.5**	**10.7**	32.5	13.2	**8.0**	**54.6**	**30.6**	**24.9**	**29.8**

Table 2. Average precision (AP) for all 8 categories in UIUC phrase dataset, mean AP across all categories. Methods: Baseline without context; Object context (rescoring using other objects); Phrase context (rescoring using the manually defined phrases); Group context (rescoring using our automatically discovered object groups).

SUN09 object dataset. SUN09 is a very challenging recent dataset containing complex scenes with many object categories and large within-class variance. Table 4 gives the mean average precision across the 107 object categories. We compare five different methods: (1) Base w/o context: the individual object detector trained using the deformable

	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbik	pers	plant	sheep	sofa	train	tv	MEAN
Base w/o context [6]	28.9	59.5	10.0	15.2	25.5	49.6	57.9	19.3	22.4	25.2	23.3	11.1	56.8	48.7	41.9	12.2	17.8	33.6	45.1	41.6	32.3
20OOI ([6])	**31.2**	61.5	**11.9**	**17.4**	27.0	49.1	59.6	23.1	23.0	26.3	24.9	12.9	60.1	**51.0**	43.2	**13.4**	18.8	36.2	**49.1**	43.0	34.1
CMO [15]	30.5	60.1	11.2	17.0	26.7	**49.7**	59.1	23.3	23.4	26.9	29.3	**13.2**	59.7	49.3	43.0	**13.4**	20.4	37.8	46.8	**43.3**	34.2
Group	29.5	**62.4**	10.8	16.4	**28.3**	**49.7**	**60.7**	**23.8**	**24.5**	**27.2**	**31.3**	**13.2**	**61.0**	49.2	**43.5**	12.7	**20.9**	**38.8**	45.3	42.6	**34.6**
Groups+OOIs+CMOs	**31.5**	**63.0**	**12.6**	**18.1**	**29.0**	**51.7**	**61.4**	**25.0**	**24.9**	**28.0**	**31.4**	**14.1**	**61.5**	**51.4**	**44.0**	**14.6**	**21.2**	**39.4**	**49.1**	**44.3**	**35.8**

Table 3. AP (%) for 20 categories in PASCAL VOC 2007 and the mean AP across 20 categories. Our proposed groups of objects outperform and are complementary to existing sources of context for object detection. Best single-context performance and best overall performance are in bold.

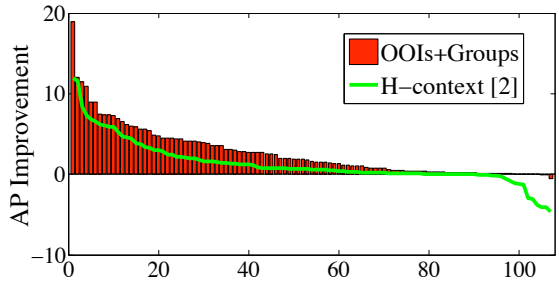

Figure 7. Improvement of different context methods over the baseline detectors on SUN09 object dataset. Object categories are sorted by the improvement in average precision (AP). (AP measured in %)

	mean AP
Base w/o context [6]	7.06
H-context [2]	8.37
OOIs	8.34
Groups	9.06
OOIs+Groups	9.75

Table 4. The mean average precision (AP) across 107 object categories in SUN09 object dataset using different methods. (APs measured in %)

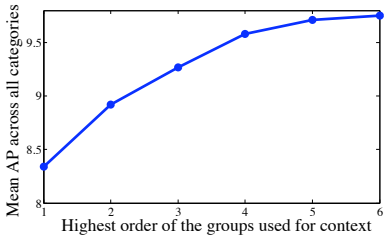

Figure 8. The mean APs (%) across the 107 object categories in SUN09 dataset as higher-order groups are included. Order = 1 indicates using the individual objects as context. Clearly, higher order groups provide useful contextual information for object detection.

part-based model [6] trained on the same additional dataset as the state-of-the-art [2]. (2) H-context: the tree based hierarchical contextual model proposed in [2] which models the inter-object contextual interactions as well as the global scene context. (3) OOIs: using all object categories to provide contextual information through re-scoring [6]. (4) Groups: using the detected groups to provide contextual information for object detection using the same re-scoring. (5) OOIs+Groups: using both objects and groups for re-scoring. Table 4 shows that even with such a simple re-scoring algorithm, using the groups as context outperforms the state-of-the-art algorithm. Although the state-of-the-art algorithm also models the spatial and scale relationship between objects, it relies on the performance of the individual object detectors. Some contextually relevant categories such as flowers and vases are both difficult to detect in isolation and can not benefit each other. However, the appearance of the flowers-vase group is more consistent and can be reliably detected. We find that our simple re-scoring method using objects and groups as context outperforms the state-of-the-art algorithm on 88 among the 107 categories on this challenging dataset, and performs comparably on the remaining. Figure 7 shows the improvement in average precision for each object category sorted by the improvement over the baseline. We note that our method rarely hurts the baseline (i.e. falls below zero) while the state-of-the-art does so to several categories. Our method also achieves larger improvement in a large number of categories.

Figure 8 shows that by increasing the highest order of groups to be used for providing contextual information, the mean performance over all object categories increases. This confirms the usefulness of high-order groups and hence the need for an automatic approach to discover these groups.

4.3. Scene Categorization

Intuitively the object groups stand as more structured components in a scene than individual objects. In this section, we make use of the object groups to improve the scene categorization performance. We consider 15 of the 67 categories in MIT indoor scene dataset [21] as described earlier.

To analyze the usefulness of object groups to represent a scene as opposed to just individual objects, we conduct an experiment of scene classification based on the groundtruth annotated objects in the images (automatic experiments follow next). We compare three image descriptors which are classified by an RBF-kernal SVM. The first is a 152-D vector indicating the occurrence of each object, the second is an analogous 52-D vector for our groups, and the third is a concatenation of both. The average accuracy for the 15 class scene categorization is respectively 81.5%(object), 84.5%(group), and 89.0%(object+group). This demonstrates the benefit of using groups of objects, as well as the complementary nature of objects as groups.

In the following experiments, we use the same training / testing split as in [21], where each scene category has 80 training images and 20 testing images. We compare different approaches for scene categorization: (1) **GIST-color** [17]: features are computed by concatenating the three 320-dimensitional GIST descriptor of the RGB channels of the image, followed by the one-vs-all SVM classifiers with RBF kernel. (2) **Spatial Pyramids (SP)** [13]: We compute the spatial pyramid features with the implementation provided by [13]. We use a vocabulary of size 200 and three levels in the pyramid, followed by one-vs-all SVM classifiers with the histogram intersection kernel. (3) **Deformable Part-based Model (DPM)**: Pandey *et al*. [18] recently proposed the use of a deformable part-based model for a scene categorization, which implicitly captures concurrent regions within a scene. (4) **Objects (OBJ)**: we represent an image with the detected individual objects. We note that due to the partial object labeling of the training images and the large variance of the object appearance, it

	airportIn	artstudio	bakery	bar	bath_rm	bed_rm	bookstore	class_rm	corridor	dine_rm	kitchen	living_rm	mtg_rm	office	warehouse	MEAN
GIST-color	10.0	10.0	26.3	16.7	72.2	47.6	15.0	**83.3**	66.7	22.2	61.9	5.0	22.7	14.3	33.3	33.8
SP	30.0	10.0	**57.9**	**50.0**	55.6	71.4	50.0	55.6	**76.2**	11.1	42.9	0.0	27.3	0.0	**61.9**	40.0
DPM[18]	25.0	40.0	47.4	33.3	**83.3**	9.5	**65.0**	**83.3**	**76.2**	27.8	61.9	25.0	**81.8**	38.1	47.6	49.7
Objects (OBJ)	**40.0**	45.0	47.4	38.9	61.1	61.9	55.0	50.0	66.7	27.8	42.9	25.0	27.3	19.1	42.9	43.4
Groups (GRP)	35.0	**50.0**	47.4	33.3	77.8	**76.2**	60.0	**83.3**	66.7	**33.3**	**66.7**	**30.0**	63.6	**33.3**	47.6	**53.6**
GIST-SP-DPM-OBJ	**40.0**	**55.0**	63.2	**44.4**	83.3	33.3	60.0	83.3	**81.0**	50.0	66.7	20.0	77.3	33.3	47.6	55.9
GIST-SP-DPM-OBJ-GRP	**40.0**	**55.0**	**68.4**	38.9	**88.9**	**61.9**	**70.0**	**88.9**	**81.0**	**61.1**	**71.4**	**25.0**	**81.8**	**42.9**	**52.4**	**61.8**

Table 5. Classification rates (%) for the 15 scene categories in MIT Indoor dataset and the mean classification rate (%) across 15 categories. Best single-approach performance and best combined performance are in bold. Our proposed groups of objects significantly boost scene recognition performance.

is difficult to train a robust object detector with the limited number of positive samples. To boost this baseline, we use 152 object detectors trained with an additional annotated dataset in [2] to detect objects. We apply the object detectors on all images. For each image, we form a 152-dimensional feature vector with each dimension indicating the highest score among the detections of each object category on the image. [2] RBF-kernel SVM classifiers are utilized for classification. (5) **Groups (GRP)**: We train detectors for our groups with the very limited positive training samples. We apply the group detectors on all images. Each image is represented as a 52-dimensional feature vector with each dimension indicating the highest score among the detections of each group on the image. Again, one-vs-all SVM classifiers with RBF kernel are utilized for classification. (6) **GIST+SP+DPM+OBJ**: We combine all the above methods except the groups by multiplying the softmax-transformed outputs of the SVMs from each method similar to [18].(7) **GIST+SP+DPM+OBJ+GRP**: We combine all the above methods including the groups.

Results. We summarize the results for the different methods in Table 5. We see that our proposed groups of objects outperform all methods. Many scenes (*e.g.* meeting room and dining room) may contain similar objects and can be confusing. On the other hand, global appearance of scenes may vary (*e.g.* the locations of cabinets or table-chair sets in dining rooms). Groups of objects (*e.g.* configuration of tables and chairs) seem to hit the right balance between the generalization and discriminative power. We also note that combining the object group results with those from the other approaches significantly improves performance.

5. Conclusion

In this work, we propose to model group of objects, which are high-order composites of objects with consistent spatial, scale, and viewpoint relationships with respect to each other across images. Manually listing all possible groups of objects is not feasible for groups containing arbitrary number of diverse object categories in a wide variety of scenes. We propose a novel Hough-transform based approach to efficiently discover the groups of objects from images annotated with object categories. We model groups of objects via deformable part-based models. Our extensive experiments on 4 challenging datasets show that the detection of groups can significantly improve both object detection and scene categorization, and outperform multiple state-of-the-art methods.

[2] We also tried using the histogram of detected objects as input, but achieved lower performance.

References

[1] M. Blaschko and C. Lampert. Object localization with global and local context kernels. In *BMVC*, 2009.
[2] M. J. Choi, J. J. Lim, A. Torralba, and A. S. Willsky. Exploiting hierarchical context on a large database of object categories. In *CVPR*, 2010.
[3] C. Desai, D. Ramanan, and C. Fowlkes. Discriminative models for multi-class object layout. In *ICCV*, 2009.
[4] S. Divvala, D. Hoiem, J. Hays, A. Efros, and M. Hebert. An empirical study of context in object detection. In *CVPR*, 2009.
[5] M. Everingham, L. V. Gool, C. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.
[6] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *PAMI, IEEE Transactions on*, 32(9):1627 –1645, sep. 2010.
[7] P. F. Felzenszwalb, R. B. Girshick, and D. McAllester. Discriminatively trained deformable part models, release 4. http://people.cs.uchicago.edu/ pff/latent-release4/.
[8] S. Fidler, M. Boben, and A. Leonardis. Evaluating multiclass learning strategies in a generative hierarchical framework for object detection. In *NIPS*, 2009.
[9] C. Galleguillos, B. McFee, S. Belongie, and G. Lanckriet. Multi-class object localization by combining local contextual interactions. In *CVPR*, 2010.
[10] C. Galleguillos, A. Rabinovich, and S. Belongie. Object categorization using co-occurrence, location and appearance. In *CVPR*, 2008.
[11] K. Grauman and T. Darrell. Unsupervised learning of categories from sets of partially matching image features. In *CVPR*, 2006.
[12] S. Kumar and M. Hebert. A hierarchical field framework for unified context-based classification. In *ICCV*, 2005.
[13] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006.
[14] Y. J. Lee and K. Grauman. Foreground focus: Unsupervised learning from partially matching images. In *IJCV*, 2009.
[15] C. Li, D. Parikh, and T. Chen. Exploiting regions void of labels to extract adaptive contextual cues. In *ICCV*, 2011.
[16] M. Marszalek and C. Schmid. Semantic hierarchies for visual object recognition. In *CVPR*, 2007.
[17] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. In *IJCV*, 2001.
[18] M. Pandey and S. Lazebnik. Scene recognition and weakly supervised object localization with deformable part-based models. In *ICCV*, 2011.
[19] D. Parikh, C. Zitnick, and T. Chen. From appearance to context-based recognition: Dense labeling in small images. In *CVPR*, 2008.
[20] D. Parikh, C. Zitnick, and T. Chen. Unsupervised learning of hierarchical spatial structures in images. In *CVPR*, 2009.
[21] A. Quattoni and A. Torralba. Recognizing indoor scenes. In *CVPR*, 2009.
[22] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie. Objects in context. In *ICCV*, 2007.
[23] N. Rasiwasia and N. Vasconcelos. Holistic context modeling using semantic co-occurrences. In *CVPR*, 2009.
[24] B. C. Russell, A. A. Efros, J. Sivic, W. T. Freeman, and A. Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In *CVPR*, 2006.
[25] M. A. Sadeghi and A. Farhadi. Recognition using visual phrases. In *CVPR*, 2011.
[26] R. Salakhutdinov, A. Torralba, and J. Tenenbaum. Learning to share visual appearance for multiclass object detection. In *CVPR*, 2011.
[27] J. Sivic, B. C. Russell, A. Zisserman, W. T. Freeman, and A. A. Efros. Unsupervised discovery of visual object class hierarchies. In *CVPR*, 2008.
[28] J. Uijlings, K. van de Sande, A. Smeulders, T. Gevers, N. Sebe, and C. Snoek. The most telling windows for image categorization. http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2011/workshop/uva.pdf.
[29] J. Xiao, J. Hays, K. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. ieee conference on computer vision and pattern recognition. In *CVPR*, 2010.
[30] B. Yao and L. Fei-Fei. Modeling mutual context of object and human pose in human-object interaction activities. In *CVPR*, 2010.
[31] Y. Zhang and T. Chen. Efficient kernels for identifying unbounded-order spatial features. In *CVPR*, 2009.