Unsupervised Learning of Hierarchical Semantics of Objects (hSOs)

Devi Parikh and Tsuhan Chen
Department of Electrical and Computer Engineering
Carnegie Mellon University, Pittsburgh, PA 15213
{dparikh,tsuhan}@cmu.edu

Abstract

A successful representation of objects in the literature is as a collection of patches, or parts, with a certain appearance and position. The relative locations of the different parts of an object are constrained by the geometry of the object. Going beyond the patches on a single object, consider a collection of images of a particular class of scenes containing multiple (recurring) objects. The parts belonging to different objects are not constrained by such a geometry. However the objects, arguably due to their semantic relationships, themselves demonstrate a pattern in their relative locations, which also propagates to their parts. Analyzing the interactions between the parts across the collection of images would reflect these patterns, and the parts can be grouped accordingly. These groupings are typically hierarchical. We introduce hSO: Hierarchical Semantics of Objects, which is learnt from a collection of images of a particular scene and captures this hierarchical grouping. We propose an approach for the unsupervised learning of the hSO. The hSO simply holds objects, as clusters of patches, at its nodes, but it goes much beyond that and also captures interactions between the objects through its structure. In addition to providing the semantic layout of the scene, learnt hSOs can have several useful applications such as providing context for enhanced object detection and compact scene representation for scene category classification.

1. Introduction

Objects that tend to co-occur in scenes of a particular category are often semantically related. Hence, they demonstrate a characteristic grouping behavior according to their relative positions in the scene. Some groupings are tighter than others, and thus a hierarchy of these groupings among these objects can be observed in a collection of images of similar scenes. It is this hierarchy that we refer to as the Hierarchical Semantics of Objects (hSO). This can be better understood with an example, which is shown in Figure 1 along with the corresponding hSO structure. Along with the structure, the hSO could also store other information such as the relative position of the objects, their co-occurrence counts, etc. as parameters. However, in this paper we focus on unsupervised learning of the hSO structure.

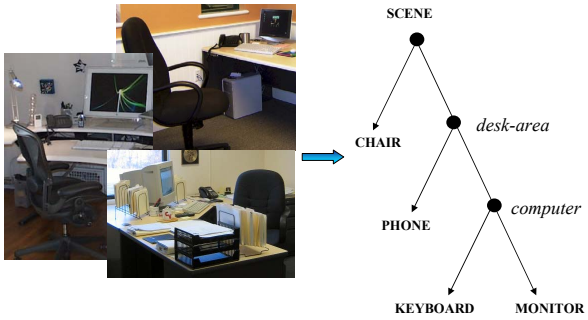

Figure 1. Left: Images for "office" scene from Google image search. There are four commonly occurring objects: chair, phone, monitor and keyboard. The monitor and keyboard occur at similar relative locations across images and hence belong to a common super-object, computer, at a lower level in the hierarchy. The phone is seen within the vicinity of the monitor and keyboard. However, the chair is randomly placed, and hence belongs to a common super-object with other objects only at the highest level in the hierarchy, the entire scene. This pattern in relative locations, often stemming from semantic relationships among the objects, provides contextual information about the scene "office" and is captured by an hSO: hierarchical semantics of objects. A possible corresponding hSO is shown on the right.

Several approaches in text data mining represent the words in a lower dimensional space where words with supposedly similar semantic meanings collapse into the same cluster. This representation is based simply on their occurrence counts in documents. Probabilistic Latent Semantic Analysis [1] is one such approach that has also been applied to images [2–4] for unsupervised clustering of images based on their *topic* and identifying the part of the images that are foreground. Our goal however is a step beyond this towards a higher level understanding of the scene. Apart from simply identifying the *existence* of potential semantic relationships between the parts (parts, features and patches are used interchangeably in the paper), we attempt to characterize these semantic relationships among these parts, and

1-4244-1180-7/07/$25.00 ©2007 IEEE

accordingly cluster them into (super) objects at various levels in the hSO.

Using hierarchies or dependencies among parts of objects for object recognition has been promoted for decades [5–13]. However we differentiate our work from these, as our goal is not object recognition, but is to characterize the scene by modeling the interactions between multiple objects in a scene. More so, although these works deal with hierarchies per se, they capture philosophically very different phenomena through the hierarchy. For instance, Marr *et al.* [8] and Levinshtein *et al.* [7] capture the shape of articulated objects such as the human body through a hierarchy whose nodes correspond to different parts of the object and links can be attachment links or decomposition links, where as Fidler *et al.* [6] capture varying levels of complexity of features at different levels. Bienenstock *et al.* [10] and Siskind *et al.* [14] learn a hierarchical structure among different parts/regions of an image. This hierarchy is based on rules similar to those that govern the grammar or syntax of language. Siskind *et al.* [14] encode absolute locations of these regions in the images as opposed to the relative interactions among the regions. These various notions of hierarchies are strikingly different from the inter-object, potentially semantic, relationships we wish to capture through a hierarchical structure. We define dependencies based on location as opposed to co-occurrence. Also, several of these approaches [5, 14] cannot effectively deal with background clutter, while we can.

Scenes may contain several objects of interest, and hand labeling these objects would be quite tedious. To avoid this, as well as the bias introduced by the subjectiveness of a human in identifying the objects of interest in a scene, unsupervised learning of hSO is preferred that truly captures the characteristics of the data. The proposed approach, being entirely unsupervised, can detect the parts of the images that belong to the foreground objects, cluster these parts to represent objects, and provide an understanding of the scene by hierarchically clustering these objects in a semantically meaningful way to extract the hSO - all from a collection of unlabeled images of a particular scene category.

It is important to note that, our approach being entirely unsupervised, the presence of multiple objects as well as background clutter makes the task of clustering the foreground parts into hierarchial clusters, while still maintaining the integrity of objects and yet capturing the inter-relationships among them, challenging; and the information coded in the learnt hSO quite rich. It entails more than a mere extension of any of the above works for single-objects to account for multiple objects.

Our approach to unsupervised learning of (the structure) of hSO is outlined in Figure 2 and described in Section 2. Section 3 describes several experimental scenarios where qualitative as well as quantitative results are provided. Section 4 concludes the paper. Although we focus only on the unsupervised learning of hSOs, before we describe the details of the learning algorithm, we first motivate hSOs through a couple of interesting potential areas for their application.

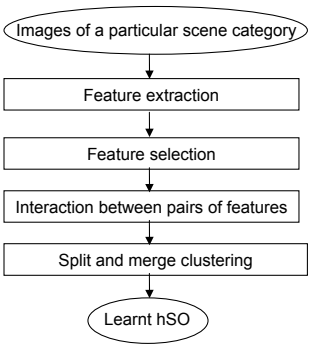

Figure 2. Flow of the proposed algorithm for unsupervised learning of hSOs

At what scale is an object defined? Are the individual keys on a keyboard objects, or the entire keyboard or is the entire computer an object? The definition of an object is blurry, and the hSO exploits this to allow incorporation of semantic information of the scene layout. The leaves of the hSO are a collection of parts and represent the objects, while the various levels in the hSO represent the super-objects at different levels of abstractness, with the entire scene at the highest level. Hence hSOs span the spectrum between specific objects, modeled as a collection of parts, at the lower level and scene categories at the higher level. This provides a rich amount of information at various semantic levels that can be potentially exploited for a variety of applications, ranging from establishing correspondences between parts for object matching, providing context for robust object detection to scene category classification.

1.1. Context

Learning the hSO of scene categories could provide contextual information about the scene and enhance the accuracy of individual detectors. Consider the example shown in Figure 1. Suppose we have independent detectors for a monitor and a keyboard. Consider a particular test image in which a keyboard is detected. However there is little evidence indicating the presence of a monitor - due to occlusion, severe pose change, etc. The learnt hSO (with parameters) for office settings would provide the contextual information indicating the presence of a monitor and also an estimate of its likely position in the image. If the observed bit of evidence in that region of the image supports this hypothesis, a monitor may be detected. However, if the observed evidence is to the contrary, not only is the monitor not detected, but the confidence in the detection of the key-

board is reduced as well. The hSO thus allows for propagation of such information among the independent detectors.

Several works use context for better image understanding. One class of approaches is analyzing individual images for characteristics of the surroundings of the object such as geometric consistency of object hypotheses [15], viewpoint and mean scene depth estimation [16, 17], surface orientations [18], etc. These provide useful information to enhance object detection/recognition. However, our goal is not to extract information about the 3D scene around the object of interest from a single image. Instead, we aim to learn a characteristic representation of the scene category and a more higher level understanding from a collection of images by capturing the semantic interplay among the objects in the scene as demonstrated across the images.

The other class of approaches has been along the lines of modeling dependencies among different parts of an image [19–25] from a collection of images. However, these approaches require hand annotated or labeled images. Our approach, is entirely unsupervised - the relevant parts of the images, and their relationships are automatically *discovered* from a corpus of unlabeled images. Also, [19–21, 24] are interested in pixel labels (image segmentation) and hence do not deal with the notion of *objects*. Torralba *et al.* [26] use the scene category for context, or learn interactions among the objects in a scene [27] for context, however their approach is supervised and the different objects in the images are annotated.

1.2. Compact scene category representation

hSOs provide a compact representation that characterizes the scene category of the images that it has been learnt from. Hence, hSOs can be used for scene category classification e.g. office, home, beach, etc. Having learnt hSOs for several scene categories, given a collection of images from an unknown scene category, the category corresponding to the hSO, among the learnt hSOs, that fits the provided collection best can be identified as the unknown category. Singhal *et al.* [28] learn a set of relationships between different regions in a large collection of images with a goal to characterize the scene category. However, these images are hand segmented, and a set of possible relationships between the different regions are predefined (above, below, etc.). Other works also categorize scenes but based on global statistics (texture like) of the scene [26, 29] and require extensive human labelling [30]. Fei-Fei *et al.* [3] group the low-level features into *themes* and *themes* into scene categories. However, the *themes* need not corresponding to semantically meaningful entities. Also, they do not include any location information, and hence cannot capture the interactions between different parts of the image. They are able to learn a hierarchy that relates the different scenes according to their similarity, however, our goal is to learn an hierarchy for a particular scene that characterizes the interactions among the entities in the scene, arguably according to the underlying semantics.

2. Unsupervised learning of hSOs

The approach we employ for unsupervised learning of hSOs is outlined in Figure 2. Each of the stages are explained in detail below. The input to the algorithm is a collection of images taken from a particular scene category, and the desired output is a learnt hSO. In this paper we focus on learning the structure of the hSO and not on learning its parameters that store additional information such as the relative location of objects with respect to each other or their co-occurrence counts. The underlying intuition behind the approach is that if two parts always lie at the same location with respect to each other, they probably belong to the same rigid object and hence should share the same leaf on the hSO. However if the position of the two parts with respect to each other varies significantly across the input images, they lie on two objects that are found at unpredictable relative locations, and are hence unrelated and should belong to a common super-object only higher up in the hSO. Other part-part, and hence object-object, relationships should lie in a spectrum in between these two extreme conditions. Since object transformations such as scale and rotation could cause even two parts of the same object to seem at different relative locations across images, we incorporate a notion of geometric consistency that ensures that two parts that are found at geometrically consistent (invariant to scale and rotation) locations across images are assigned to the same cluster/object.

2.1. Feature extraction

Given the collection of images taken from a particular scene category, local features describing interest points/parts are extracted in all the images. These features may be appearance based features such as SIFT [31], shape based features such as shape context [32], geometric blur [33], or any such discriminative local descriptors as may be suitable for the objects under consideration. In our current implementation, we use the Derivative of Gaussian interest point detector, and SIFT features as our local descriptors.

2.2. Correspondences

Having extracted features from all images, correspondences between these local parts are to be identified across images. For a given pair of images, potential correspondences are identified by finding k nearest neighbors of each feature point from one image in the other image according to an appropriate distance metric. We use Euclidean distance between the SIFT descriptors to determine the nearest neighbors. The geometric consistency between every pair

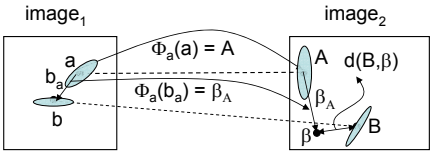

Figure 3. An illustration of the geometric consistency metric used to retain *good* correspondences.

of correspondences is computed to build a geometric consistent adjacency matrix.

Suppose we wish to compute the geometric consistency between a pair of correspondences shown in Figure 3 involving interest regions a and b in $image_1$ and A and B in $image_2$. All interest regions have a scale and orientation associated with them. Let ϕ_a be the similarity transform that transforms a to A. β_A is the transformed b_a, the relative location of b with respect to a in $image_1$, using ϕ_a. β is thus the estimated location of B in the $image_2$ based on ϕ_a. If a and A, as well as b and B are geometrically consistent under rotation and scale, $d(B, \beta)$ would be small. A score that decreases exponentially with increasing $d(B, \beta)$ is used to quantify the geometric consistency of the pair of correspondences. To make the score symmetric, a is similarly mapped to α using the transform ϕ_b that maps b to B, and the score is based on $max(d(B, \beta), d(A, \alpha))$. This metric provides us with invariance to scale and rotation, however does not allow for affine transforms, but the assumption is that the distortion due to affine transform in realistic scenarios is minimal among local features that are closely located on the same object.

Having computed the geometric consistency score between all possible pairs of correspondences, a spectral technique is applied to the geometric consistency adjacency matrix to retain only the geometrically consistent correspondences [34]. This helps eliminate most of the background clutter. This also enables us to deal with incorrect low-level correspondences among the SIFT features that can not be reliably matched, for instance at various corners and edges found in an office setting. To deal with multiple objects in the scene, an iterative form of [34] is used.

2.3. Feature selection

Only the feature points that find geometrically consistent corresponding points in most other images are retained. This post processing step helps to eliminate the remaining background features. Since we do not require a feature to be observed in all the images in order to be retained, occlusions, severe view point changes, even missing objects in some images can be handled. Also, this enables us to deal with different number of objects in the scene across images - the assumption being that the objects that are present in most images are the objects of interest (foreground), while

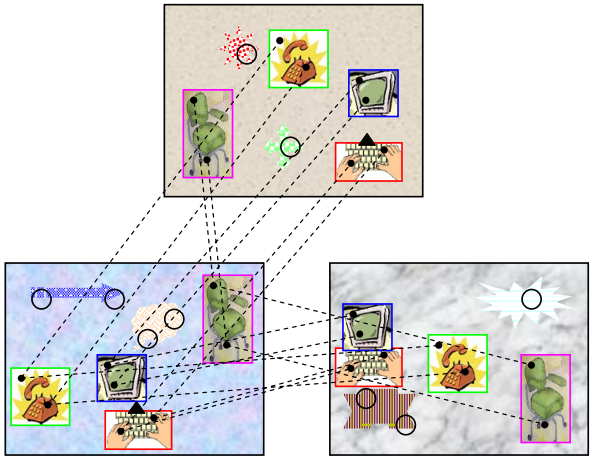

Figure 4. An illustration of the correspondences and features retained during learning of hSOs. The images contain four foreground objects, and some background. This example is only an illustration and not a real output.

those that are present in a few images are part of the background clutter. This proportion can be varied to suit the scenario at hand.

We now have a reliable set of foreground feature points and a set of correspondences among all images. An illustration can be seen in Figure 4 where the only a subset of the detected features and their correspondences are retained.

It should be noted that the approach being unsupervised, there is no notion of an object yet. We only have a cloud of patches in each image and correspondences among them. The goal is to now separate these patches into different clusters (each cluster corresponding to a foreground object in the image), and also learn the hierarchy among these objects that will be represented as an hSO that will characterize the entire collection of images and hence the scene.

2.4. Interaction between pairs of features

In order to separate the cloud of retained feature points into clusters, a graph is built over the feature points, where the weights on the edge between the nodes represents the interaction between the pair of features across the images. The metric used to capture the interaction between the pairs of features is what we loosely refer to as the correlation of the location of the two feature points across the input images. Let us assume, for simplicity of notation, that the same number of features have been retained in all input images. We have the correspondences among these features between every pair of images. Let F be the number of fea-

tures retained in each of the N input images. Suppose $\boldsymbol{M}$ is the $F \times F$ correlation adjacency matrix, then $\boldsymbol{M}_{ij}$ holds the interaction between the i^{th} and j^{th} features as

$$\boldsymbol{M}_{ij} = R(x_i x_j) + R(y_i y_j), \tag{1}$$

where, $R(x_i x_j) = \frac{C(x_i x_j)}{\sqrt{C(x_i x_i) C(x_j x_j)}}$, where, $C(x_i x_j)$ is the covariance between x_i and x_j across the input images, and $x_i = \{x_{in}\}, y_i = \{y_{in}\}$, (x_{in}, y_{in}) is the location of the i^{th} feature point in the n^{th} image, $i, j = 1, \ldots, F$, $n = 1, \ldots, N$. In addition to $R(x_i x_j)$ and $R(y_i y_j)$ in Equation 1, $R(x_i y_j)$ and $R(y_i x_j)$ could also be included. Using correlation to model the interaction between pairs of features implicitly assumes a Gaussian distribution of the location of one features conditioned on the other, similar to those made by traditional constellation models [35].

If the correlation between the location of two feature points from Equation 1 is high, they appear at similar relative locations across images. On the other hand, if the correlation between the location of two feature points is low, they occur at unpredictable locations with respect to each other across the images. An illustration of the correlation adjacency matrix can be seen in Figure 5. Again, there is no concept of an object yet. The features in Figure 5 are arranged in an order that correspond to the objects, and each object is shown to have only two features (consistent with example in Figure 4), only for illustration purposes.

2.5. Split and merge clustering

Having built the graph capturing the interaction between all pairs of features across images, recursive clustering is performed on this graph. At each step, the graph is clustered into two clusters. The properties of each cluster is analyzed, and one or both of the clusters are further separated into two clusters, and so on. If the variance in the correlation adjacency matrix corresponding to a certain cluster (subgraph) is very low but with a high mean, it is assumed to contain parts from a single object, and is hence not divided further. Every stage in this recursive clustering adds to the structure of the hSO being learnt. It can be verified for the example shown in Figure 5, where the hSO learnt would be the one shown in Figure 1. Since the statistics of each of the clusters formed are analyzed to determine if it should be further clustered or not, the number of foreground objects need not be known *a priori*. We use normalized cuts [36] to perform the clustering. The code provided at [37] was used. This is the splitting step.

As stated earlier, transformations of objects in the scene could lead the correlation values for a pair of features to be small, even if the corresponding objects are the same or are at similar locations in images (except under different transformations). This could lead us to cluster a cloud of features corresponding to the same object into multiple clusters during the splitting step. If the transformations are significant, such as rotation of a relatively large object, the correlation among the parts of the object that are far may seem lower than the correlation among parts that lie on two different objects that are at similar locations across images. Thus, early on in the above splitting stage (at higher levels in the hSO), a single object may have been broken down into multiple clusters, even before two different objects in the scene have been separated. Although in the subsequent stages in splitting, the parts on different objects would be separated into different clusters, the different clusters formed from the same object early on can not be re-combined. To rectify this, the geometric consistency score computed in Section 2.2 (averaged across all images containing these features) is now re-considered. All pairs of clusters formed at the end of the splitting stage are examined and those that are in fact geometrically consistent are merged together, since they are likely to lie on the same object. This is repeated till no two clusters are geometrically consistent. For every merge, among the levels at which these individual clusters were placed in the hSO before merging, the merged cluster is placed at the lowest level, since correlation was underestimated, and redundant nodes are removed. This gives us the final hSO structure. The merging step attempts to ensure that the final clusters of features do in-fact correspond to objects in the scene. This split and merge approach to clustering is similar in philosophy to that used in the image segmentation literature.

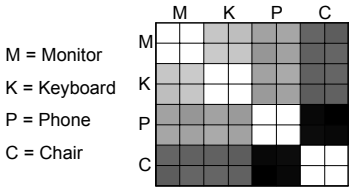

Figure 5. An illustration of the correlation adjacency matrix of the graph that would be built for the illustration in Figure 4 in a scenario depicted in Figure 1.

3. Experimental results

It should be noted that the goal of this work is not improved object recognition in the sense of better feature extraction or matching. We focus our efforts at learning the hSO that codes the different interactions among objects in the scene by using well matched parts of objects, and not on the actual matching of parts. This work is complementary to the recent advances in object recognition that enable us to deal with object categories and not just specific objects. These advances indicate the feasibility to learn hSO even among objects categories. However, in our experiments we use specific objects with SIFT features to demonstrate our proposed algorithm. However SIFT is not an integral part

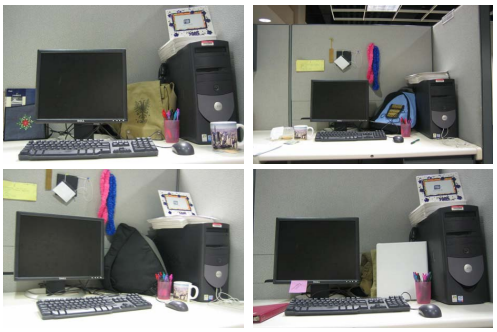

Figure 6. A subset of images provided as input to learn the corresponding hSO.

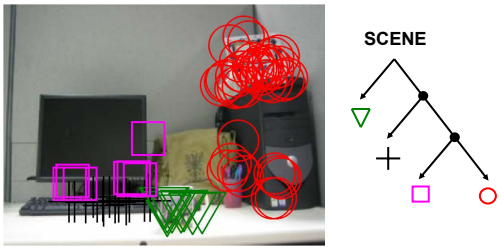

Figure 7. Results of the hSO learning algorithm. Left: The cloud of features clustered into groups. Each group corresponds to an object in the foreground. Right: The corresponding learnt hSO which captures meaningful relationships between the objects.

of our approach. This can easily be replaced with patches, shape features, etc. with appropriate matching techniques as may be appropriate for the scenario at hand - specific objects or object categories. Future work includes experiments in such varied scenarios.

Several different experimental scenarios were used to learn the hSOs. Due to lack of standard datasets where interactions between multiple objects can be modeled, we use our own collection of images.

3.1. Scene semantic analysis

Consider a surveillance type scenario where a camera is monitoring, say an office desk. The camera takes a picture of the desk every few hours. The hSO characterizing this desk, learnt from this collection of images could be used for robust object detection in this scene, in the presence of occlusion due to the person present, or other extraneous objects on the desk. Also, if the objects on the desk are later found in an arrangement that cannot be explained by the hSO, it can be detected as an anomaly. Thirty images simulating such a scenario were taken. Examples of these can be seen in Figure 6. Note the occlusions, presence of background clutter, change in scale and viewpoint, etc. The corresponding hSO as learnt from these images is depicted in Figure 7.

Several different interesting observations can be made. First, the background features are mostly eliminated. The features on the right-side of the bag next to the CPU are retained while the rest of the bag is not. This is because due to several occlusions in the images, most of the bag is occluded in images. However, the right-side of the bag resting on the CPU is present in most images (not all), and hence is interpreted to be foreground. The monitor, keyboard, CPU and mug are selected to be the objects of interest (although the mug is absent in some images). The hSO indicates that the mug is found at most unpredictable locations in the image, while the monitor and the CPU are clustered together till the very last stage in the hSO. This matches our semantic understanding of the scene. Also, since the photo frame, the right-side of the bag and the CPU are always found at the same location with respect to each other across images (they are stationary), they are clustered together as the same object. Ours being an unsupervised approach, this artifact is expected as there is no way for the algorithm to segment these into separate objects.

3.2. Photo grouping

We consider an example application where the goal is to obtain the semantic hierarchy among photographs. We present users with 6 photos of which 3 are outdoor (2 beaches, 1 garden) and 3 indoor (2 of a person in an office, 1 empty office). These photos can be seen in Figure 8. The users were instructed to group these photos such that the ones that are similar are close by. The number of groups to be formed was not specified. Some users made two groups (indoor vs. outdoor), while some made four groups by further separating these two groups into two each. We took images of 20 such arrangements. Example images are shown in Figure 9. We use these images to learn the hSO. The results obtained are shown in Figure 10. We can see that the hSO can capture the semantic relationships among the images, including the general (indoor vs. outdoor) as well as more specific ones (beaches vs. garden) through the hierarchical structure. It should be noted that the content of the images was not utilized to compute the similarity between images and group them accordingly - this is based purely on the user arrangement. In fact, it may be argued that although this grouping seems very intuitive to us, it may be very challenging to obtain this grouping through low level features extracted from the photos. Such an hSO on a larger number of images can hence be used to empower a content based digital image retrieval system with the users semantic knowledge. In such a case a user-interface, similar to [38], may be provided to users and merely the position of each image can be noted to learn the underlying hSO without requiring feature extraction and image matching. In [38], although user preferences are incorporated, a hierarchial notion of interactions is not employed which provides much

Figure 8. The six photos that users arranged.

Figure 9. A subset of images of the different arrangements of photos that users provided for which the corresponding hSO was learnt.

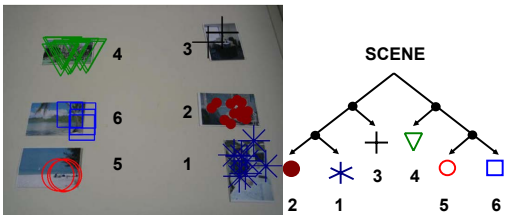

Figure 10. Results of the hSO learning algorithm. Left: The cloud of features clustered into groups. Each group corresponds to a photograph. Right: The corresponding learnt hSO which captures the appropriate semantic relationships among the photos. Each cluster and photograph is tagged with a number that matches those shown in Figure 8 for clarity.

richer information.

3.3. Quantitative results

In order to better quantify the performance of the proposed algorithm, a hierarchy among objects was staged i.e. the ground truth hSO is known. As shown in the example

Figure 11. A subset of images provided as input to learn the corresponding hSO.

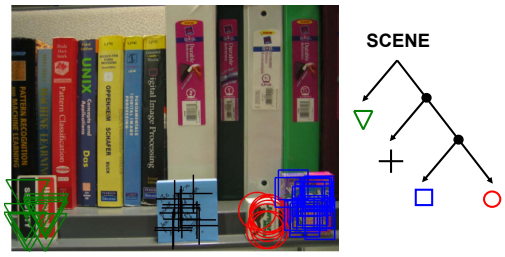

Figure 12. Results of the hSO learning algorithm. Left: The cloud of features clustered into groups. Each group corresponds to an object in the foreground. Right: The corresponding learnt hSO which matches the ground truth hSO.

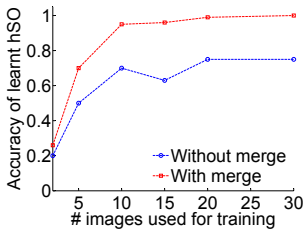

Figure 13. The accuracy of the learnt hSO as more input images are provided. Also, the need for a merging step after the splitting stage in clustering is illustrated.

images in Figure 11, two candy boxes are placed mostly next to each other, a post-it-note around them, and an entry card is tossed randomly. Thirty such images were captured against varying cluttered backgrounds. Note the rotation and change in view point of the objects, as well as varying lighting conditions. These were hand-labeled so that the ground truth assignments of the feature points to different nodes in the hSO are known and accuracies can be computed. The corresponding hSO was learnt from these (unlabeled) images. The results obtained are as seen in Figure 12. The feature points have been clustered appropriately, and the learnt hSO matches the description of the ground truth

hSO above. The clutter in the background has been successfully eliminated. Quantitative results reporting the accuracy of the learnt hSO, measured as the proportion of features assigned to the correct level in the hSO, with varying number of images used for learning are shown in Figure 13. It can be seen that with significantly few images a meaningful hSO can be learnt. Also, the accuracy of the hSO learnt if the merge step as described in Section 2.5 is not incorporated after the splitting stage is reported. It should be noted that this accuracy simply reports the percentage of features detected as foreground that were assigned to the right levels in the accuracy. While it penalizes background features considered as foreground, it does not penalize dropping foreground features as background and hence not considering them in the hSO. Visual quality of results indicate that such a metric suffices. In less textured objects the accuracy metric would need to be reconsidered.

4. Conclusion

We introduced hSOs: hierarchical semantics of objects that capture potentially semantic relationships among multiple objects in a scene as observed by their relative positions in a collection of images. The underlying entity is a patch, however the hSO goes beyond patches and represents the scene at various levels of abstractness - ranging from patches on individual objects to objects and groups of objects in a scene. An unsupervised hSO learning algorithm has been proposed. The algorithm can identify the relevant parts of the images (foreground), and discover the relationships between these parts and the objects they belong to - automatically and entirely unsupervised. Potential directions of future work include learning the parameters of the hSO and applying it to provide context for enhancing the performance of independent object detectors.

Acknowledgments

We thank Andrew Stein and Dhruv Batra for discussion and code to compute geometrically compatible correspondences among images.

References

[1] T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 2001.

[2] J. Sivic, B. Russell, A. Efros, A. Zisserman, W. Freeman. Discovering objects and their location in images. *ICCV*, 2005.

[3] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. *CVPR*, 2005.

[4] P. Quelhas, F. Monay, J.M. Odobez, D. Gatica-Perez, T. Tuytelaars, and L. Van Gool. Modeling scenes with local descriptors and latent aspects. *ICCV*, 2005. 1

[5] G. Bouchard, W. Triggs. Hierarchical part-based visual object categorization. *CVPR*, 2005.

[6] S. Fidler, G. Berginc, A. Leonardis. Hierarchical statistical learning of generic parts of object structure. *CVPR*, 2006.

[7] A. Levinshtein, C. Sminchisescu, S. Dickinson. Learning hierarchical shape models from examples. *EMMCVPR*, 2005.

[8] D. Marr, H. Nishihara. Representation and recognition of the spatial organization of three dimensional structure. *Proceedings of the Royal Society of London B*, 1978.

[9] I. Biederman. Human image understanding: recent research and a theory. *Computer Vision, Graphics and Image Processing*, 1985.

[10] E. Bienenstock, S. Geman, D. Potter. Compositionality, MDL priors, and object recognition. *NIPS*, 1997.

[11] E. Sudderth, A. Torralba, W. Freeman, A. Wilsky. Learning hierarchical models of scenes, objects, and parts. In ICCV, 2005.

[12] G. Wang, Y. Zhang, L. Fei-Fei. Using dependent regions for object categorization in a generative framework. *CVPR*, 2006.

[13] Y. Jin, S. Geman. Context and hierarchy in a probabilistic image model. *CVPR*, 2006.

[14] J. Siskind, J. Sherman, I. Pollak, M. Harper, C. Bouman. Spatial random tree grammars for modeling hierarchal structure in images with regions of arbitrary shape. *PAMI*, to appear.

[15] D. Forsyth, J. Mundy, A. Zisserman, C. Rothwell. Using global consistency to recognise euclidean objects with an uncalibrated camera. *CVPR*, 1994.

[16] A. Torralba, A. Oliva. Depth estimation from image structure. *PAMI*, 2002.

[17] A. Torralba, P. Sinha. Statistical context priming for object detection. *ICCV*, 2001.

[18] D. Hoiem, A. Efros, M. Hebert. Putting objects in perspective. *CVPR*, 2006.

[19] A. Storkey, C. Williams. Image modelling with position encoding dynamic trees. *PAMI*, 2003.

[20] C. Williams, N. Adams. DTs: Dynamic trees. *NIPS*, 1999.

[21] G. Hinton, Z. Ghahramani, Y. Teh. Learning to parse images. *NIPS*, 2000.

[22] Z. Tu, X. Chen, A. Yuille, S. Zhu. Image parsing: unifying segmentation, detection, and recognition. *IJCV*, 2005.

[23] K. Murphy, A. Torralba, W. Freeman. Using the forest to see the trees: a graphical model relating features, objects, and scenes. *NIPS*. 2003.

[24] X. He, R. Zemel, M. Carreira-Perpinan. Multiscale conditional random fields for image labeling. *CVPR*, 2004.

[25] S. Kumar, M. Hebert. A hierarchical field framework for unified context-based classification. *ICCV*, 2005.

[26] A. Torralba, K. Murphy, W. Freeman, M. Rubin. Context-based vision system for place and object recognition. *ICCV*, 2003.

[27] A. Torralba, K. Murphy, W. Freeman. Contextual models for object detection using boosted random fields. *NIPS*, 2005.

[28] A. Singhal, J. Luo, W. Zhu. Probabilistic spatial context models for scene content understanding. *CVPR*, 2003.

[29] A. Oliva, A. Torralba. Modeling the shape of the scene: a holistic representation of the spatial envelope. *IJCV*, 2001.

[30] J. Vogel, B. Schiele. A semantic typicality measure for natural scene categorization. *Pattern Recognition Symposium, DAGM*, 2004.

[31] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*,2004.

[32] S. Belongie, J. Malik, J. Puzicha. Shape context: a new descriptor for shape matching and object recognition. *NIPS*, 2000.

[33] A. Berg, J. Malik. Geometric blur for template matching. *CVPR*, 2001.

[34] M. Leordeanu and M. Hebert. A spectral technique for correspondence problems using pairwise constraints. *ICCV*, 2005.

[35] M. Weber, M. Welling, P. Perona. Unsupervised learning of models for recognition. *ECCV*, 2000.

[36] J. Shi, J. Malik. Normalized cuts and image segmentation. *PAMI*, 2000

[37] J. Shi. http://www.cis.upenn.edu/ jshi/software/

[38] M. Nakazato, L. Manola, T. Huang, ImageGrouper: search, annotate and organize image by groups. *VISual*, 2002.