Hierarchical Semantics of Objects (hSOs)

Devi Parikh, Tsuhan Chen
Carnegie Mellon University
{dparikh,tsuhan}@cmu.edu

Abstract

We introduce hSOs: Hierarchical Semantics of Objects. An hSO is learnt from a collection of images taken from a particular scene category. The hSO captures the interactions between the objects that tend to co-occur in the scene, and hence are potentially semantically related. Such relationships are typically hierarchical. For example, in a collection of images taken in a living room scene, the TV, DVD player and coffee-table co-occur frequently. The TV and the DVD player are more closely related to each other than the coffee table, and this can be learnt from the fact that the two are located at similar relative locations across images, while the coffee table is somewhat arbitrarily placed. The goal of this paper is to learn this hierarchy that characterizes the scene. 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 - all from a collection of unlabeled images of a particular scene category. In addition to providing the semantic layout of the scene, learnt hSOs can have several useful applications such as compact scene representation for scene category classification and providing context for enhanced object detection.

1. Introduction

Objects that tend to co-occur in scenes 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 an 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.

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

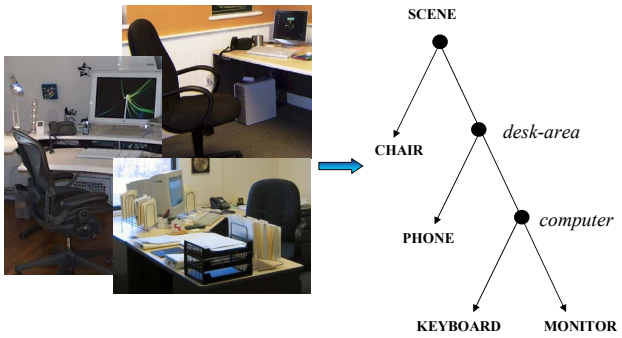


Figure 1. Images for "office" scene from Google image search. There are four commonly occurring objects: chair, phone, monitor and keyboard. The monitor and keyboard occur at similar relative locations across images and hence belong to a common super-object, computer, at a lower level in the hierarchy. 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.

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.

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 occur-

978-1-4244-1631-8/07/$25.00 ©2007 IEEE

rence 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 relationship between the parts, we attempt to characterize these semantic relationships among these parts, and accordingly cluster them into (super) objects at various levels in the hSO. We define dependencies based on relative location as opposed to co-occurrence.

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 say, 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, 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 based on rules on absolute locations of the regions in the images, similar to those that govern the grammar or syntax of language. These various notions of hierarchies are strikingly different from the inter-object, potentially semantic, relationships we wish to capture through a hierarchical structure.

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. 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.

Before we describe the details of the learning algorithm, we first motivate hSOs through a couple of interesting potential areas for their application.

1.1. Context

Learning the hSO of scene categories could provide contextual information about the scene and enhance the accuracy of individual detectors by providing a prior over the likely position of an object, given the position of another object in the scene.

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 surroundings of 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 models dependencies among different parts of an image [19–25] from a collection of images. However, these approaches require hand annotated or labeled 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 global statistics of the image to predict the type of scene which provides context for the location of the object, however their approach is also supervised. Torralba *et al.* [27] learn interactions among the objects in a scene for context however again, their approach is supervised and the different objects in the images are annotated. Marszałek *et al.* [28] also learn relationships among multiple classes of objects, however indirectly through a lexical model learnt on the labels given to images, and hence is a supervised approach. Our approach, is entirely unsupervised - the relevant parts of the images, and their relationships are automatically *discovered* from a corpus of unlabeled images.

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. Singhal *et al.* [29] 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 [30, 31] also categorize scenes but require extensive human labeling. 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 an 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 inter-

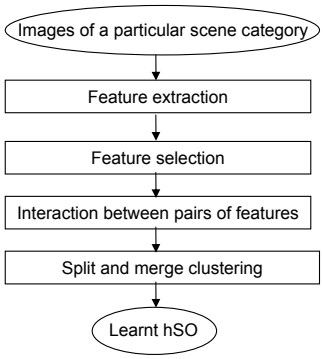


Figure 2. Flow of the proposed algorithm for unsupervised learning of hSOs

actions among the entities in the scene, arguably according to the underlying semantics.

Our approach to unsupervised learning of (the structure) of hSO is outlined in Figure 2 and described in Section 2. Section 3 presents experiments and results on the unsupervised learning of hSO as well as using an hSO to provide context for robust object detection. Section 4 concludes the paper.

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 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 [32], shape based features such as shape context [33], geometric blur

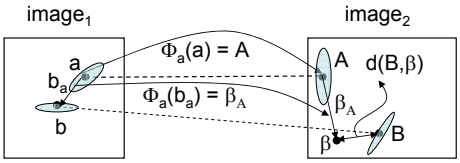


Figure 3. An illustration of the geometric consistency metric used to retain *good* correspondences.

[34], 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 of correspondences is computed to build a geometric consistent adjacency matrix, $\boldsymbol{M}_G$. This is done as follows.

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 [35]. 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 [35] is used.

2.3. Feature selection

Only the feature points that find geometrically consistent corresponding points in most other images, and not just a pair of 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, etc. can be handled. Using multiple images gives us the ability to fully take advantage of the fact that erroneous matches are random, while true matches are mostly consistent. We now have a reliable set of foreground feature points and a set of correspondences among all images. 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 features retained in each of the N input images. Suppose $\boldsymbol{M}_R$ is the $F \times F$ correlation adjacency matrix we wish to fill, then $\boldsymbol{M}_R(i,j)$ holds the interaction between the i^{th} and j^{th} features as

$$\boldsymbol{M}_R(i,j) = R(x_i x_j) + R(y_i y_j), \tag{1}$$

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 traditional constellation models [36].

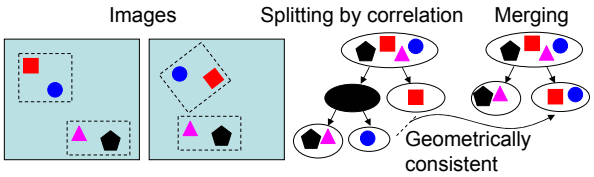


Figure 4. Illustration of the split and merge clustering algorithm

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 represented via $\boldsymbol{M}_R$. At each step, the graph is clustered into two clusters. The properties of each cluster are 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. 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 [37] to perform the clustering. The code provided at [38] 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 (but only under different transformations). If the transformations are significant, such as rotation of a relatively large object, the correlation among the parts of the object that are further apart may seem lower than the correlation among parts that lie on two different objects however are at similar locations across images. This is illustrated in Figure 4. 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 are separated in subsequent steps, and these can not be recombined. To rectify this, the geometric consistency score computed in Section 2.2 $\boldsymbol{M}_G$ is now reconsidered. The score is averaged across all images containing these selected foreground features. Due to this accumulation of statistics across images, the noise is significantly suppressed than while considering only a pair of images as was done in Section 2.2. All pairs of clusters formed at the end of the splitting stage are examined and those that are in fact geometrically consistent according to these accumulated statistics, are merged together, since they are likely to lie on the same object. This

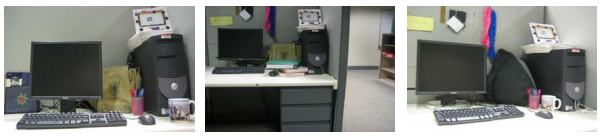


Figure 5. A subset of images provided as input to learn the corresponding hSO.

Figure 6. 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.

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 under-estimated), and redundant states are removed. This is illustrated in Figure 4. 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.

3. Experimental results

It should be noted that the goal of this work is not improved object recognition through 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. In our experiments we use specific objects with SIFT features to demonstrate our proposed algorithm, however SIFT is not an integral part of our approach. It can be replaced with patches, shape features, etc. with relevant matching techniques as may be appropriate for the scenario at hand - specific objects or object categories. Recent advances in object recognition indicate the feasibility to learn hSO even among objects 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 5. Note the occlusions, background clutter, change in scale and viewpoint, etc. The corresponding hSO as learnt from these images is depicted in Figure 6.

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, 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 keyboard 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, and natural even, since there is in fact no evidence indicating these entities to be separate objects.

3.2. Photo grouping

We consider an example application where the goal is to learn the semantic hierarchy among photographs. This experiment is to demonstrate the capability of the proposed algorithm to truly capture the semantic relationships, by bringing users in the loop, since semantic relationships are not a very tangible notion. We present users with 6 photos: 3 outdoor (2 beaches, 1 garden) and 3 indoor (2 with a person in an office, 1 empty office). These photos can be seen in Figure 7. 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 pictures that capture 20 such arrangements. Example images are shown in Figure 8. We use these images to learn the hSO. The results obtained are shown in Figure 9. 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

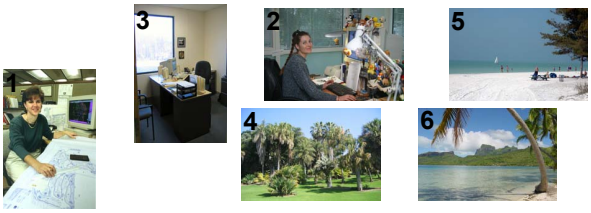


Figure 7. The six photos that users arranged.

Figure 8. A subset of images of the arrangements of photos that users provided for which the corresponding hSO was learnt.

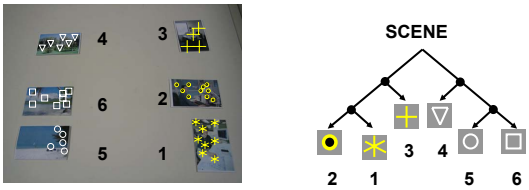


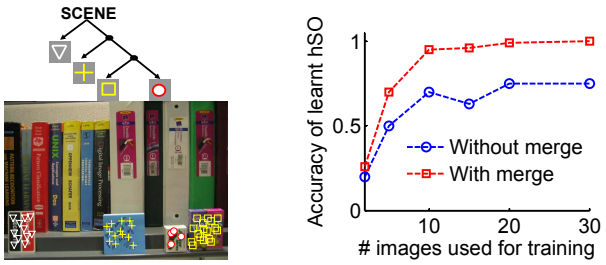
Figure 9. Results of the hSO learning algorithm. Left: The cloud of features clustered into groups. Each group corresponds to a photograph. Right: The corresponding learnt hSO which captures the appropriate semantic relationships among the photos. Each cluster and photograph is tagged with a number that matches those shown in Figure 7 for clarity.

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 [39], 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 [39], although user preferences are incorporated, a hierarchial notion of interactions is not employed which provides much richer information.

3.3. Quantitative results

In order to better quantify the performance of the proposed learning algorithm, a hierarchy among objects was staged i.e. the ground truth hSO is known. As shown in the example images in Figure 10, 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

Figure 10. A subset of images provided as input to learn the corresponding hSO.



(a) (b)

Figure 11. (a) 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. (b) 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.

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 the unlabeled images. The results obtained are as seen in Figure 11(a). 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 11(b). 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.

3.4. Context for robust object detection

Consider the hSO learnt for the office scene in Section 3.1 as shown in Figure 12. Consider an image of the same scene (not part of the learning data) as shown in Figure 13 which has significant occlusions (real on the keyboard, and synthetic on the CPU and mug). We wish to detect the four foreground objects.

The leaves of the hSO hold the clouds of features (along with their locations) for the corresponding objects. To detect the objects, these are matched with features in the test

Figure 12. The information flow used within hSO for context. Solid bi-directional arrows indicate exchange of context. Dotted directional arrows indicate flow of (refined) detection information. The image on the left is shown for reference for what objects the symbols correspond to.

Figure 13. Test image in which the four objects of interest are to be detected. Significant occlusions are present.

Figure 14. Left: candidate detections of keyboard, along with the max score (incorrect) detection. Middle: context prior provided by detected monitor. Right: detections of keyboard after applying context from monitor along with the max score (correct) detection. The centers of the candidate detections are shown.

Figure 15. Detections of the 4 objects without context (left) - 3 of 4 are incorrect due to significant occlusions. Detections with context (right) - all 4 are correct.

image through geometrically consistent correspondences similar to that in Section 2.2. Multiple candidate detections along with their corresponding scores are retained, as seen in Figure 14 (left). The detection with the highest score is determined to be the final detection. Due to significant occlusions, background may find candidate detections with higher scores and hence the object would be miss-detected, as seen in Figure 15 (left), where three of the four objects are incorrectly localized.

Instead we use the context provided by other objects in the scene for robust detection. The structure of the hSO indicates that the *siblings* i.e. the entities (objects or super-objects) sharing the same parent node in the hSO structure are the most informative for each other to predict their location. Hence, during learning, we learn the parameters of the relative location of an entity only with respect to its sibling in the hSO; as compared to learning the interaction among all objects (a flat network structure instead of hierarchy) where all possible combinations of objects would need to be considered entailing learning of a large number of parameters, which for a large number of objects and limited training data could be prohibitive. Each entity is treated as a point with a scale and orientation computed from the distribution of features or objects that compose it. The relative locations (normalized for scale and orientation) are modeled as Gaussian distributions that provide the context prior. Hence, in addition to the cloud of features at the leaves, each node in the hierarchy holds the mean and variance of the location of its sibling relative to its own position.

The flow of information used to incorporate the context is shown in Figure 12. In the test image, candidate detections of the foreground objects at the lowest level (L_0) in the hSO structure are first determined. The context prior provided by each of these (two) objects is applied to the other object and these detections are pruned/refined as shown in Figure 14. The distribution in Figure 14 (middle) is strongly peaked because it indicates the relative location of the keyboard with respect to the monitor, which is quite predictable. However, the distribution of the absolute location of the keyboard across the training images as shown in Figure 5 is significantly less peaked. The hSO allows us to condition on the appropriate objects and obtain such peaked contextual distributions. This refined detection information is passed on to the next higher level (L_1) in the hSO, which constitutes the detection information of the super-object containing these two objects, which in turn provides context for refining the detection of the other object at L_1, and so on.

In the presence of occlusion, even if a background match has a higher score, it will most likely be pruned out by the context prior. The detection results obtained by using context is shown in Figure 15 (right) which correctly localizes all four objects. The objects, although significantly occluded, are easily recognizable to us. So the context is not hallucinating the objects entirely, but is amplifying the available (little) evidence at hand, while enabling us to not be distracted by the false background matches.

Ongoing work involves a more theoretical treatment of the hSO, formalizing the independence assumptions made, avoiding making hard decisions at each level in the hSO to

determine the final detection, as well as iteratively flowing information from leaves to the root as well as the root to the leaves of the hSO till convergence.

4. Conclusion

We introduced hSOs: Hierarchical Semantics of Objects that capture potentially semantic relationships among objects in a scene as observed by their relative positions in a collection of images. An unsupervised hSO learning algorithm has been proposed. Given a collection of images of a scene, the algorithm can identify the foreground parts of the images, discover the relationships between these parts and the objects they belong to, learn the appearance models of these objects as well as relative location models for related objects and use these to provide context for robust object detection even with significant occlusions in a new test image - automatically and entirely unsupervised. This, we believe, takes us a step closer to true image understanding.

Acknowledgments

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

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, 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.

[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. *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. *AI Memo, MIT*, 2003.

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

[28] M.Marszałek, C.Schmid. Semantic hierarchies for visual object recognition. *CVPR*, 2007.

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

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

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

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

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

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

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

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

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

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

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