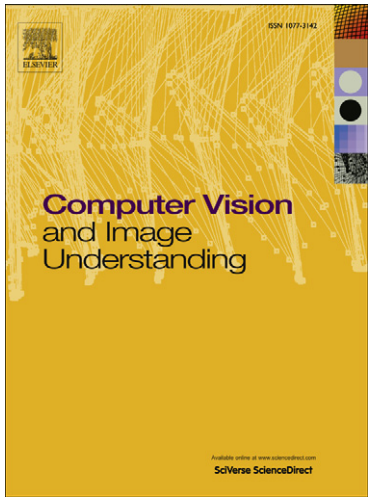


Discovering Hierarchical Object Models from Captioned Images

Michael Jamieson, Yulia Eskin, Afsaneh Fazly, Suzanne Stevenson, Sven J. Dickinson

PII: S1077-3142(12)00049-5
DOI: 10.1016/j.cviu.2012.03.002
Reference: YCVIU 1868

To appear in: *Computer Vision and Image Understanding*

Received Date: 23 April 2011
Revised Date: 17 November 2011
Accepted Date: 5 March 2012

Please cite this article as: M. Jamieson, Y. Eskin, A. Fazly, S. Stevenson, S.J. Dickinson, Discovering Hierarchical Object Models from Captioned Images, *Computer Vision and Image Understanding* (2012), doi: 10.1016/j.cviu.2012.03.002

Discovering Hierarchical Object Models from Captioned Images

Michael Jamieson, Yulia Eskin, Afsaneh Fazly, Suzanne Stevenson, Sven J. Dickinson

University of Toronto

Abstract

We address the problem of automatically learning the recurring associations between the visual structures in images and the words in their associated captions, yielding a set of named object models that can be used for subsequent image annotation. In previous work, we used language to drive the perceptual grouping of local features into configurations that capture small parts (patches) of an object. However, model scope was poor, leading to poor object localization during detection (annotation), and ambiguity was high when part detections were weak. We extend and significantly revise our previous framework by using language to drive the perceptual grouping of parts, each a configuration in the previous framework, into hierarchical configurations that offer greater spatial extent and flexibility. The resulting hierarchical multi-part models remain scale, translation and rotation invariant, but are more reliable detectors and provide better localization. Moreover, unlike typical frameworks for learning object models, our approach requires no bounding boxes around the objects to be learned, can handle heavily cluttered training scenes, and is robust in the face of noisy captions, *i.e.*, where objects in an image may not be named in the caption, and objects named in the caption may not appear in the image. We demonstrate improved precision and recall in annotation over the non-hierarchical technique and also show extended spatial coverage of detected objects.

Keywords: language-vision integration, object recognition, automatic image annotation, learning hierarchical models

Email address: {jamieson, yulia, afsaneh, suzanne, sven}@cs.toronto.edu (Michael Jamieson, Yulia Eskin, Afsaneh Fazly, Suzanne Stevenson, Sven J. Dickinson)

1. Introduction

The automatic learning of visual object models from training images has become a common component of today's object recognition systems. However, such automatic model learning has previously required a high degree of supervision. For example, bounding boxes or bounding contours are typically used to locate the object in a cluttered training image in order to strongly constrain the search for recurring (nonaccidental) features [1, 2]. Alternatively, if the objects are composed of parts, the number of parts is typically specified [3]. In other approaches, the image may be canonically located, oriented, and scaled in the image, and cropped to avoid significant clutter or occlusion [4]. The proposed methods for learning object models thus depend on strong explicit or implicit supervision, in the form of bounding boxes, part constraints, image cropping, or constrained position, scale, or orientation. Such methods are unable to scale to the problem of discovering recurring models from entirely unstructured image collections that contain multiple occluded objects appearing in cluttered scenes.

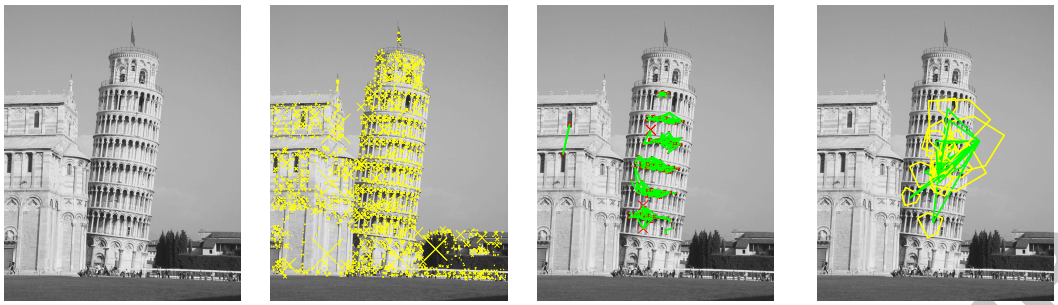
In contrast, a naturally-occurring form of supervision exists in image captions, which often identify objects of interest in the images. Captions may be keywords intended to name objects in the image, as in the case of an annotated photo collection, or full-sentence captions, in the case of a document containing images, which typically contain nouns referring to the depicted objects. However, neither of such captions are explicit, noise-free supervisory signals. For example, some caption words may correspond to objects that do not appear in the image, or, conversely, one or more objects in an image may not be mentioned in the caption. Even if a named object in the caption does appear in the image, the object may appear at any position, orientation, and scale; it may be occluded by other objects; or it may be a small part of a heavily cluttered scene. Still, if a particular image object co-occurs sufficiently often with an appropriate caption word, we can exploit this recurring correspondence to learn both a visual object model as well as its name.

A number of researchers have exploited this observation in work on automatic image annotation [5, 6, 7, 8, 9, 10]. Given cluttered images of multiple objects paired with noisy captions, such systems can learn meaningful correspondences between caption words and appearance models. The learned

visual models and their associated learned names can then be used to annotate uncaptioned scenes by adding a model's name as a keyword to any image containing the model. However, automatic annotation systems have generally been limited in their ability to capture structured object models, instead using appearance models based on colors or textures that are best for structureless materials (*e.g.*, [5]), or appearance models that capture part structure but without spatial information (*e.g.*, [6, 7, 9, 10]).

By contrast, in our previous work [8], we introduced the first framework that used language to drive the iterative grouping of image features into structured appearance models. Given images of cluttered scenes, associated with potentially noisy captions, our previous method can discover spatial configurations of local features that strongly correspond to particular caption words. However, the framework suffered from a number of limitations. First, a learned model tended to capture the structured appearance of only a small patch on an object. While such individual learned parts were often sufficient to indicate the presence of particular exemplar objects, they had limited spatial extent, and the system could not distinguish whether a collection of part detections in an image arose from multiple objects or were multiple parts of a single object. Moreover, the limited scope of the parts also led to poor object localization, since the center of the object would be located at the center of a single patch, which typically was not the center of the actual object. Finally, the small size and scope of the models increased their ambiguity, leading to poor annotation precision when patches were only weakly detected.

One effective strategy for creating more useful representations is to learn a hierarchy of parts in which parts at each level are grouped together into meaningful configurations to form the next higher level [11, 12, 13, 14]. The hierarchical representations are inspired by and intended to reflect the compositional appearance of natural objects and artifacts. For instance, each level of the Leaning Tower of Pisa (Figure 1) appears as a ring of arches while the tower as a whole is composed of a nearly vertical stack of such rings. The broader object recognition literature contains many methods for grouping individual features into meaningful spatial configurations (*e.g.*, [15, 16]), and even for arranging features into hierarchies of parts (*e.g.*, [17, 18, 19, 20, 21, 22]). Some of these methods can learn an appearance model from training images with cluttered backgrounds, sometimes without relying on bounding boxes. However, unlike most automatic annotation work, they are not designed for images containing multiple objects and mul-



(a) pixels (b) local features (c) parts (d) multipart models

Figure 1: Object model detection and learning progresses in stages. Gradient patterns in the original image (a) are grouped into local features (b). Configurations of local features with strong word correspondence are captured as part models (c). Finally, we form multipart models (d) that represent meaningful configurations of part models (shown in yellow) having nonaccidental spatial relationships among them (shown in green).

tiple annotation words.

Here we have integrated this strategy of hierarchical part models into our earlier work on language-driven perceptual grouping [8], producing a system with more accurate image annotation, as well as improved object scope and localization. The new system has two important enhancements over the previous method. Foremost, our new system constructs spatially-configured multipart models, or MPMs, by grouping the local configurations built by our previous method. The local configurations in our previous method thus become the input parts that are grouped by our current method to form higher-level structures. The creation of MPMs, as with the creation of the parts, is driven by the correspondence with the words in the captions. We have also developed a new initialization process that improves the overall distribution of the initially discovered local configurations, optimizing the overall correspondence of each with its associated word. This has benefits both for the initial approach using only local appearance models, as well as for the new MPMs. The resulting MPMs are more robust to occlusion, articulation and changes in perspective than our earlier appearance models. The use of MPMs further reduces false annotations resulting from weak part detections, and provides a better indication of the location and extent of a detected object. Figure 1 illustrates how low-level features are assembled in

stages to form a multipart model for the Leaning Tower of Pisa.[1]

The paper is organized as follows. In Section 2, we review our part representation from [8] and introduce our multipart model representation as a hierarchy of such parts. Section 3 describes our improved strategy for discovering recurring part–noun correspondences in a set of captioned training images. Next, in Section 4, we present our method for building multipart models out of these detected parts. Given a set of named multipart models, we then describe in Section 5 how the models are detected in new images, allowing an uncaptioned image to be annotated with the models present in the image. In Section 6, we evaluate the approach on three different datasets, discussing improvements in performance as well as remaining limitations of the method. We close in Section 7 with our conclusions and future work.

2. Images, Parts and Multipart Models (MPMs)

Our system learns multipart models (MPMs) by detecting recurring configurations of lower-level parts that together appear to have a strong correspondence with a particular caption word. The parts are themselves configurations of image features. Though our overall approach is compatible with a variety of image feature representations, in the system here we use interest points as in [8]. The description below of local features and part models is taken from that work.

An image is represented as a set of local interest points, $I = \{p_m | m = 1 \ldots |I|\}$. These points are detected using [23], which defines each point's spatial coordinates $\mathbf{x}_m$, scale λ_m, and orientation θ_m. A PCA-SIFT [24] feature vector ($\mathbf{f}_m$) describes the portion of the image around each point. In addition, a vector of scale-, translation-, and rotation-invariant spatial relationships r_{mn} is defined between each pair of points, p_m and p_n. This includes the relative distance between the two points (Δx_{mn}) normalized by the scale of the finer point, the relative scale difference ($\Delta\lambda_{mn}$) and the relative bearings in each direction ($\Delta\phi_{mn}$, $\Delta\phi_{nm}$). That is, $r_{mn} = (\Delta x_{mn}, \Delta\lambda_{mn}, \Delta\phi_{mn}, \Delta\phi_{nm})$.

A part appearance model describes the distinctive appearance of an object part as a graph $G = (V, E)$. Each vertex $v_i \in V$ is composed of a continuous

[1] Note that while the system as implemented here uses exemplar-specific SIFT features, the framework we have developed could employ categorical features, such as contours of higher-order shape parts [1].

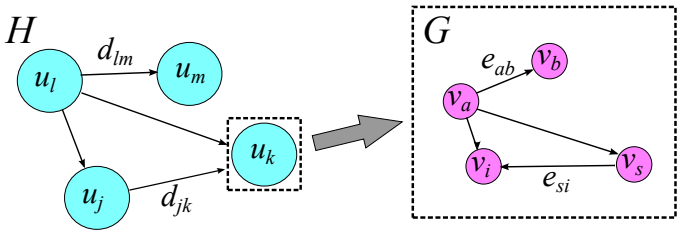


Figure 2: A multipart model H is a graph with parts $u_j \in U$ and spatial relationships $d_{jk} \in D$, where each part is a graph G with local features $v_i \in V$ and spatial relationships $e_{si} \in E$.

feature vector $\mathbf{f}_i$ (describing an interest point), and each edge $e_{ij} \in E$ encodes the expected spatial relationship between two vertices, v_i and v_j. Model detections have a confidence score, $\text{Conf}_{detect}(O, G) \in [0, 1]$, based on the relative likelihood that the part model G generates the observed set of points O and the associated spatial relations, as opposed to them being part of the unstructured background.

We represent multipart models using a similar graph structure, but one in which the vertices are the parts just described (rather than image features), and the edges are spatial relations among those parts. That is, a multipart model is a graph $H = (U, D)$ where vertices $u_j, u_k \in U$ are part appearance model detections, and each edge $d_{jk} \in D$ encodes the spatial relationships between them. We use the same spatial relations as in the part model: $d_{jk} = (\Delta x_{jk}, \Delta\lambda_{jk}, \Delta\phi_{jk}, \Delta\phi_{kj})$. Thus the vertices of the MPM represent a set of local appearances that tend to co-occur in a loose spatial configuration encoded in the graph edges. Figure 2 shows an example MPM and one of its parts.

3. Discovering Parts

Our MPMs use as their parts the same type of individual appearance models as in our earlier work. However, the means for discovering such parts requires some modification, since models trained to maximize stand-alone detection performance are generally not ideal as parts of a larger appearance model. In particular, single-part appearance models, as we used before, need to act as high-precision detectors. In contrast, when used as components of an MPM, the parts need to be more individually ambiguous to allow sharing of such representations across a variety of MPMs. We then rely on

the structure of the MPMs to reduce false-positive detections by imposing co-occurrence and spatial constraints on the parts. Thus when learning parts for use in an MPM, we accept some loss of precision in exchange for better recall and better spatial coverage of the object of interest. We implement this shift toward weaker parts with better coverage by replacing the part initialization process in [8] with an improved approach that makes earlier use of language information, as described below. We also limit the size of learned part models to eight vertices, to avoid the creation of parts that are overly specific.

3.1. Part Model Initialization through Image Pair Sampling

In performing part model initialization, we need to efficiently identify (albeit noisily) recurring structure that is larger than a single local feature, in order to begin building configurations of such points. Our previous system summarized the visual information within each local area of all the images in a dataset as a quantized bag-of-features descriptor called a *neighborhood pattern*. The system then clustered similar neighborhood patterns in order to focus the search for recurring structure within areas of high similarity across images. Next, the system checked for promising co-occurrence patterns between each neighborhood cluster and each word. Finally, the system extracted initial two-vertex appearance models from those clusters with the best correspondences for each word.

This clustering approach to initializing part models has serious limitations. Because the neighborhood patterns are noisy, due to feature quantization and detector errors, a low similarity threshold is needed to reliably group similar appearances. However, the low threshold also incorrectly allows dissimilar neighborhoods to join in a cluster. Especially on large image sets, this can add substantial noise in determining the cluster–word co-occurrences. These noise inclusions have a larger effect on correspondence when the true appearance cluster is small. Therefore recurring visual structure corresponding to rarer object views is often overlooked.

Our new initialization method avoids feature quantization and uses word labels early in the process. Instead of using a neighborhood pattern, we compare visual features directly. Rather than coarsely clustering visual structure across the entire training set, we look for instances of shared appearance between pairs of images with the same word label. That is, for a given word w, the system randomly samples pairs of images I_A and I_B from those with

captions containing w, and identifies neighborhoods in the two images that share visual structure.

We identify shared neighborhoods in three steps. First, the system looks for the best-matching features that are potential anchors for shared neighborhoods. Following [23], we identify matching features that are significantly closer to each other than to either feature's second-best match, *i.e.*, features $\mathbf{f}_m \in I_A$ and $\mathbf{f}_n \in I_B$ that satisfy equations 1 and 2:

$$|\mathbf{f}_m - \mathbf{f}_n|^2 \leq \psi_b |\mathbf{f}_m - \mathbf{f}_k|^2, \forall \mathbf{f}_k \in \{I_B - \mathbf{f}_n\} \quad (1)$$

$$|\mathbf{f}_m - \mathbf{f}_n|^2 \leq \psi_b |\mathbf{f}_l - \mathbf{f}_n|^2, \forall \mathbf{f}_l \in \{I_A - \mathbf{f}_m\} \quad (2)$$

where $\psi_b < 1$ controls the quality of the best anchor matches. This is illustrated in Figure 3(a). For each pair of best-matching features, the system checks for supporting matches in the surrounding neighborhood, as illustrated in Figure 3(b). These supporting matches are not required to be the best; that is, we use use $\psi_s > 1$. For each supporting match pair $f_i \in I_A$ and $f_j \in I_B$, the system then verifies that the point-wise spatial relationships between the best feature and the supporting feature in the two images (r_{mi} and r_{nj}) are consistent. A shared neighborhood has a pair of best matching features and at least two spatially consistent pairs of supporting matches.

Given this evidence of shared visual structure, we construct a set of two-vertex part models, each with one vertex based on the best match and the other on a strong supporting match, as illustrated in Figure 3(c). These two-vertex models represent shared visual structure between two images labeled with word w. To check whether the part models correspond with w, the system detects each model G across the training image set and compares its occurrence pattern with that of w. Below, we explain how we sample image pairs and filter the resulting initial part models to maximize overall coverage of the object.

3.2. Part Coverage Objective

Our earlier system developed the n neighborhood clusters having the best correspondence with w into full appearance models. This approach concentrated parts on the most common views of an object, neglecting less common views and appearances associated with w. Our new method instead selects initial part models so that, *as a group*, they have good coverage of w throughout the training set, as illustrated in Figure 3(d).

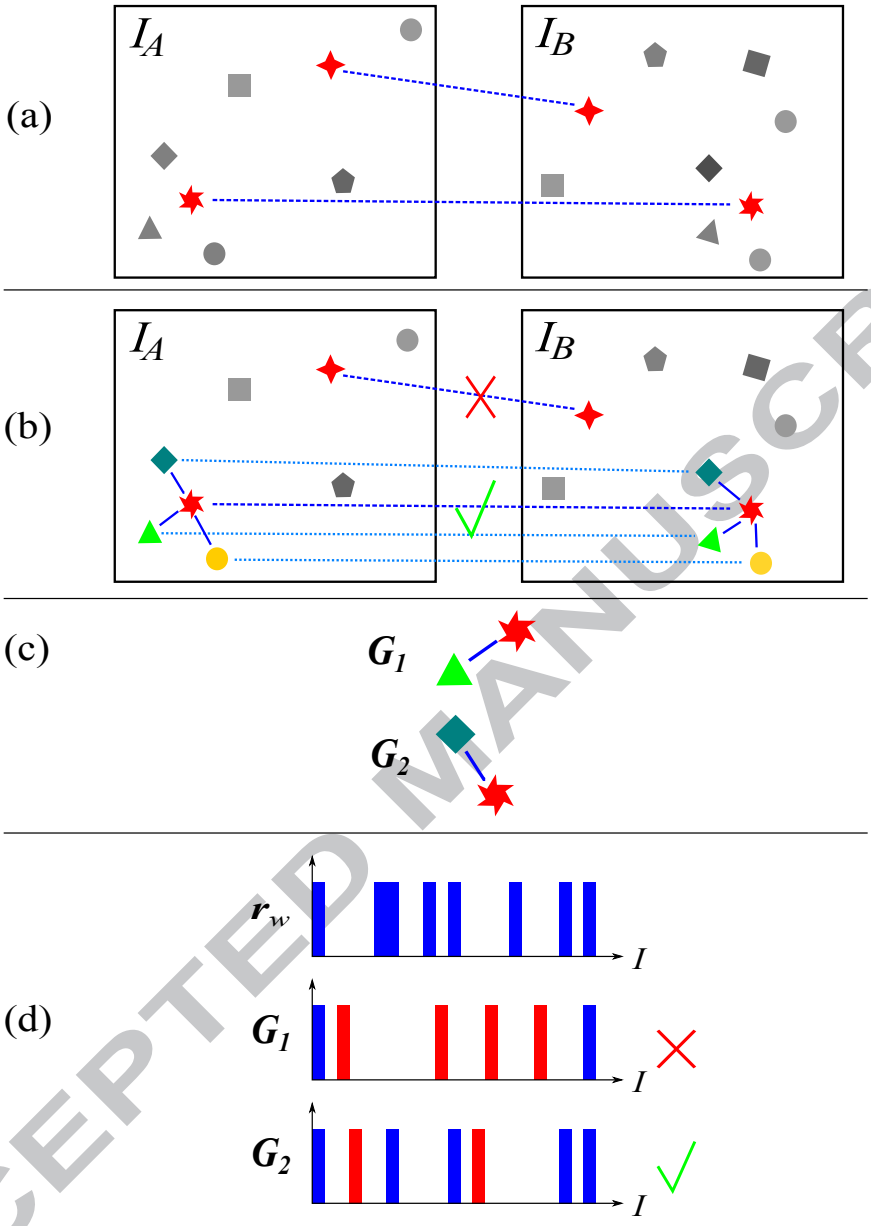


Figure 3: (a) A pair of images, I_A and I_B, associated with caption word w, have best-matching features. (b) One of the matches has supporting features that (c) generate potential initial two-vertex part models G_1 and G_2 (we select the two best-matching potential models). (d) G_2 passes the correspondence threshold with w.

The ideal set of models would have multiple, non-overlapping detections in every training set image annotated with word w, and no detections in other training images. We evaluate how well a given distribution of model detections approaches this ideal using a correspondence measure F between a binary vector $\mathbf{r}_w$, indicating images with w in the caption, and a continuous vector $\mathbf{Q}_w$, whose scores assess the detections of parts in the image. We devise $\mathbf{Q}_w$ (detailed shortly) to range from 0 to 1, with multiple detections of the same part having a lower value than many detections of different parts; good coverage is indicated by $\mathbf{Q}_w$ having values close to 1.

The part initialization process, shown in Figure 4 (and described in more detail below), greedily grows and modifies a collection of non-overlapping two-vertex part models to maximize $F(\mathbf{r}_w, \mathbf{Q}_w)$. At each iteration, we draw a pair of images according to a sampling distribution $\mathbf{s}_w$, and use the image pair to generate potential part models. The algorithm then calculates, for each potential model, the effects on the correspondence score F of adding the model to the current part set, of replacing, in turn, each of the models (parts) in the current part set, and of rejecting the model. The algorithm implements the option which leads to the greatest improvement in correspondence. The process stops once no new models have been accepted in the last N_{pairs} image-pair samples.

Specifically, the initialization process proceeds as follows. Given a set of training images I, $\mathbf{Q}_w = \{Q_{wi} | i = 1 \ldots |I|\}$ is a detection vector representing the distribution of appearance model detections throughout the training set. Given n_i representing the number of distinct models detected in image i, $Q_{wi} = 1 - \nu^{n_i}, \nu < 1$. The vector $\mathbf{Q}_w$ approaches 1 when there are multiple detections per image, but each successive detection has a smaller effect on Q_{wi}. Therefore there is little potential reward (or penalty) for introducing new part detections in images that already have several different parts. The detection vector $\mathbf{Q}_w$ also influences the sample distribution from which we draw the image pairs: $\mathbf{s}_w \sim 1 - \mathbf{r}_w * \mathbf{Q}_w$ (suitably normalized). This focuses the search for new models on images with word w in the caption that do not already contain several model detections.

As the overall objective function $F(\mathbf{r}_w, \mathbf{Q}_w)$, we use F-measure. For a positive real value β, F-measure is a weighted average of precision and recall:

$$F_\beta = \frac{(1 + \beta^2) \cdot \text{precision} \cdot \text{recall}}{(\beta^2 \cdot \text{precision}) + \text{recall}} \tag{3}$$

Later, during individual model improvement, in which precision is more im-

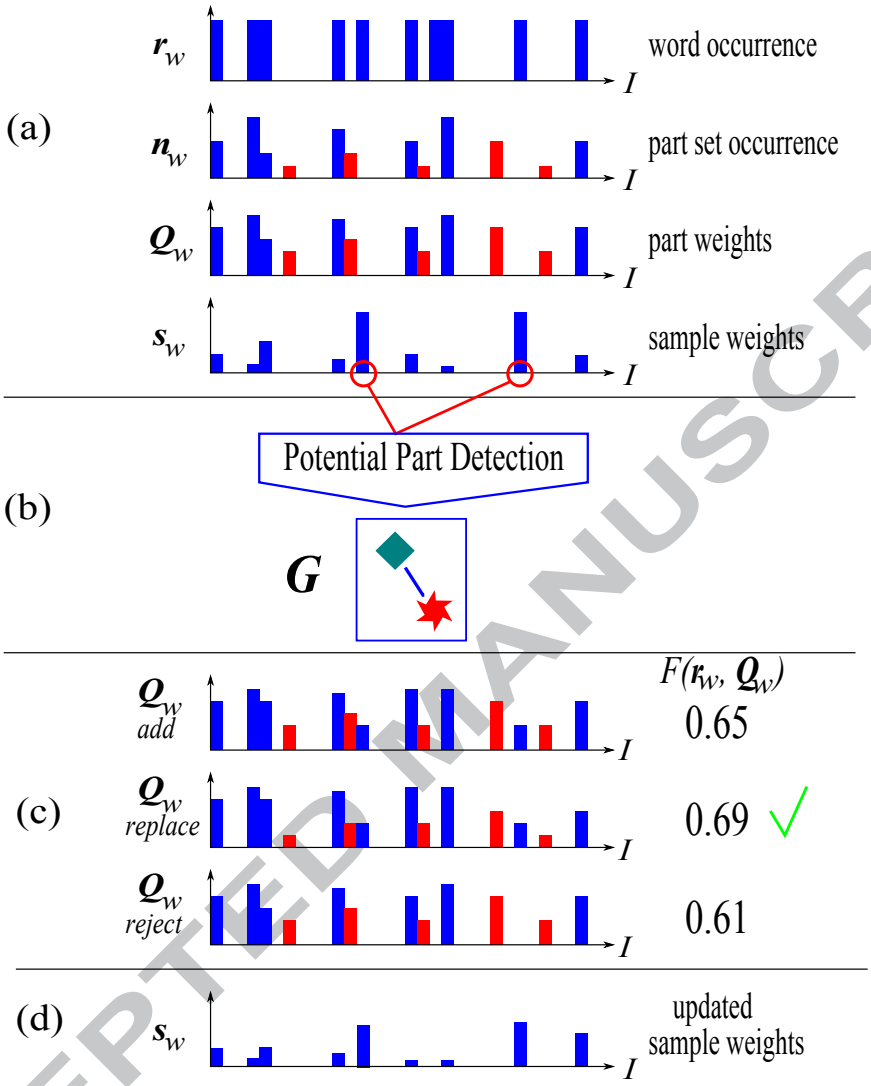


Figure 4: (a) The distribution of word w ($\mathbf{r}_w$) and the distribution of detections of the current part set ($\mathbf{n}_w$, converted to $\mathbf{Q}_w$) determine the image sampling distribution ($\mathbf{s}_w$) for (b) new part detection. (c) For each potential part G, we use the correspondence between $\mathbf{r}_w$ and $\mathbf{Q}_w$ to determine whether G should be added to the current part set, replace spatially overlapping parts, or be rejected. (d) If the part set changes, the image sample weights are updated.

portant, we use the correspondence confidence score, $\text{Conf}_{corr}(G, w)$ from [8] as our objective. To calculate correspondence confidence, we find the pattern of occurrence of a part model G across the training set using the maximum detection confidence ($\text{Conf}_{detect}(O, G)$) per image and evaluate whether G could be a high-precision predictor of the word w. $\text{Conf}_{corr}(G, w)$ is the log-likelihood ratio of this 'reliable indicator' hypothesis and the hypothesis that G and w are independent. Given a per-image detection confidence for multi-part models, the same method is used to evaluat correspondence confidence for MPMs.

Besides optimizing the explicit objective function, the initialization system also avoids redundant models with many overlapping detections. Two models are considered to be redundant when their detections overlap nearly as often as they occur separately. When a new two-vertex model is considered, if selected it must replace any models that it makes redundant. Algorithm 1 summarizes the steps of part initialization.

Algorithm 1 Choose initial part models to maximize overall coverage

FindInitialParts($\mathbf{r}_w$)

1. Start with $n_i = 0, \forall i$, indicating no part models detected in the image set; therefore $\mathbf{Q}_w$ and $\mathbf{s}_w$ are uniform.
2. Draw I_A and I_B (without replacement) from $\mathbf{s}_w$.
3. Find shared neighborhoods and construct the set of potential part models, $\mathcal{G}_{pot}$.
4. Filter $\mathcal{G}_{pot}$ based on $F_{0.25}$ correspondence with w.
5. For each remaining part model $G \in \mathcal{G}_{pot}$:
   - Calculate overlap of G with set of current part models, $\mathcal{G}_{curr}$.
   - If G overlaps with some elements of $\mathcal{G}_{curr}$, calculate $\mathbf{Q}^*_w$ for G replacing overlapping models.
   - Else calculate $\mathbf{Q}^*_w$ for addition and for replacement of each element of $\mathcal{G}_{curr}$ by G
   - Accept best change $\mathbf{Q}^*_w$ if $F_1(\mathbf{r}_w, \mathbf{Q}^*_w) > F_1(\mathbf{r}_w, \mathbf{Q}_w)$.
   - Update $\mathbf{Q}_w$ according to $\mathbf{Q}^*_w$.
6. Update $\mathbf{s}_w$ and go to step 2.
7. If N_{pairs} samples with no change in model set accepted, return.

4. Building Multipart Models

After learning distinctive part models, but before assembling them into multipart models, we perform several stages of preprocessing. Algorithm 2 summarizes both the preprocessing steps and the MPM initialization and assembly process, with reference to the sections of the text that explain the steps of the algorithm.

Algorithm 2 Assemble MPMs from parts associated with word w.

ConstructMPMs(w)

1. For each part G associated with w, find the set $\mathcal{O}_G$ of observations of G in training images.
2. Identify and remove redundant parts (section 4.1).
3. For each G, set the spatial coordinates of each observation $O_G \in \mathcal{O}_G$ (section 4.2):
   - Choose representative vertex v_c to act as the "center" of G (the vertex with minimum graph eccentricity).
   - For each $v_i \in \mathbf{v}_G$, find average spatial relationship, $\bar{\mathbf{r}}_{ic}$, between co-occurrences of $(v_i, v_c) \in \mathcal{O}_G$.
   - For each $O_G \in \mathcal{O}_G$, and each observed vertex $\mathbf{p_i} \in O_G$ calculate expected spatial coordinates ($\mathbf{x}_c$) of the central vertex (v_c) based on ($\bar{\mathbf{r}}_{ic}, \mathbf{x}_i$). The spatial coordinates $\mathbf{x}_G$ of the overall part are the average of the expected center coordinates $\bar{\mathbf{x}}_c$. The scale of $\mathbf{x}_G$ is the average of the expected scales multiplied by the spread of part G.
4. Sort all parts G associated with w by $\text{Conf}_{corr}(G, w)$.
5. For each G:
   - Skip expansion if most $O_G \in \mathcal{O}_G$ are already incorported into existing MPMs (section 4.3).
   - Initialize a new MPM H using G as the seed part.
   - Iteratively expand H using same method as for part models (section 4.4):
     - Search for parts and spatial relationships co-occurring with the detections of H.
     - Expand MPM H to H^* by adding new part or spatial relationship.
     - Detect H^* across the training image set (section 5).
     - If new MPM–word correspondence, $\text{Conf}_{corr}(H^*, w) > \text{Conf}_{corr}(H, w)$, $H \Leftarrow H^*$.
     - Reject H if it cannot be expanded beyond a single part G.
   - If at least N_{MPM} multipart models have been created, return.

4.1. Detecting Duplicate Parts

Our initialization method avoids excessive overlap of initial part models. However, during model refinement, two distinct part models can converge to cover the same portion of an object's appearance. In forming multipart models, near-duplicate parts could be erroneously interpreted as a pair of independent parts that are strongly co-occurring. We prune such parts as follows.

Detecting near-duplicates by searching for partial isomorphisms between part models would be overly complex. Instead we look for groups of parts that tend to be detected in the same images at overlapping locations. If a vertex v_{Ai} in model G_A maps to the same image point as vertex v_{Bj} in model G_B in more than half of detections, then we draw an equivalence between v_{Ai} and v_{Bj}. If more than half of the vertices in either part are equivalent, we remove the part with the weakest model–word correspondence confidence score $\text{Conf}_{corr}(G, w)$.

4.2. Locating Part Detections

Just as a part is comprised of local interest points in certain spatial configurations, so too a multipart model is comprised of parts that have certain spatial relationships among them. Note that a local interest point detector provides that point's scale, orientation and location, which we can use to encode spatial relationships within our parts (see Section 3). However, the part detector does not provide such information for the part itself—we must discover the spatial relationships between the detected parts. We do this by setting the spatial coordinates for each part detection based on the underlying image points in a way that is robust to occlusion and errors in feature detection, as follows.

For each part we select a vertex to serve as the "center" for the part—*i.e.*, the vertex that has minimum eccentricity, equal to the graph radius. (Ties are resolved in favor of vertices that have appeared more often in detections of the part.) Then, for each vertex in the part, we average the spatial relationship between it and the central vertex across all of their cooccurrences in the part detections. Now we can estimate the location of this central vertex even for part detections in which it is not observed, by using the observed vertices to predict its expected location. Figure 5 illustrates this approach. We use the estimated location and orientation of the central vertex as the location and orientation of the part, and multiply the estimated scale of the central vertex

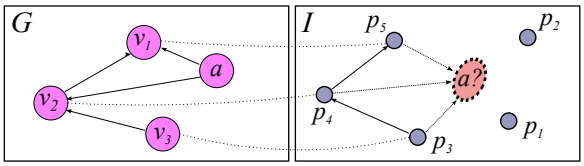


Figure 5: The spatial coordinates of a part detection are based on a central vertex a. We estimate a's coordinates based on observed vertices, even if a itself is not observed.

by a part-specific factor so that the detected part scale reflects the normal spread of the part's vertices.

4.3. Choosing Initial Multipart Models

Our system uses the most promising individual part models as seeds for constructing multipart models. Parts that have good correspondence with a word are likely to co-occur with other parts in stable patterns from which large MPMs with good spatial coverage can be constructed. However, if only the strongest part models are expanded, the resulting MPMs may be overly clustered around only the most popular views of the object. This would neglect views with weaker, more ambiguous individual parts. It is precisely these views where MPMs can be most helpful in improving precision by adding additional constraints.

Therefore initial model selection proceeds as follows. Part models are evaluated in the order of their correspondence with a word w. A model is used as a seed for an MPM if at least half of its 'good' detections (in images labeled with w) have not been incorporated into any of the already-expanded MPMs. Selective expansion continues until the list of part models is exhausted or N_{MPM} distinct multipart models have been constructed for a given word.

4.4. Expansion of Multipart Models

In order to expand the multipart models, we use an approach very similar to that used in our earlier work (and described above) to expand part models—*i.e.*, we use the correspondence strength $\text{Conf}_{corr}(H, w)$ between a multipart model H and a word w to guide its expansion into a larger multipart model. The correspondence score is calculated in the same way for parts and MPMs. It reflects the amount of evidence, available in a set of training images, that a word and a part model are generated from a common underlying source object, as opposed to appearing independently.

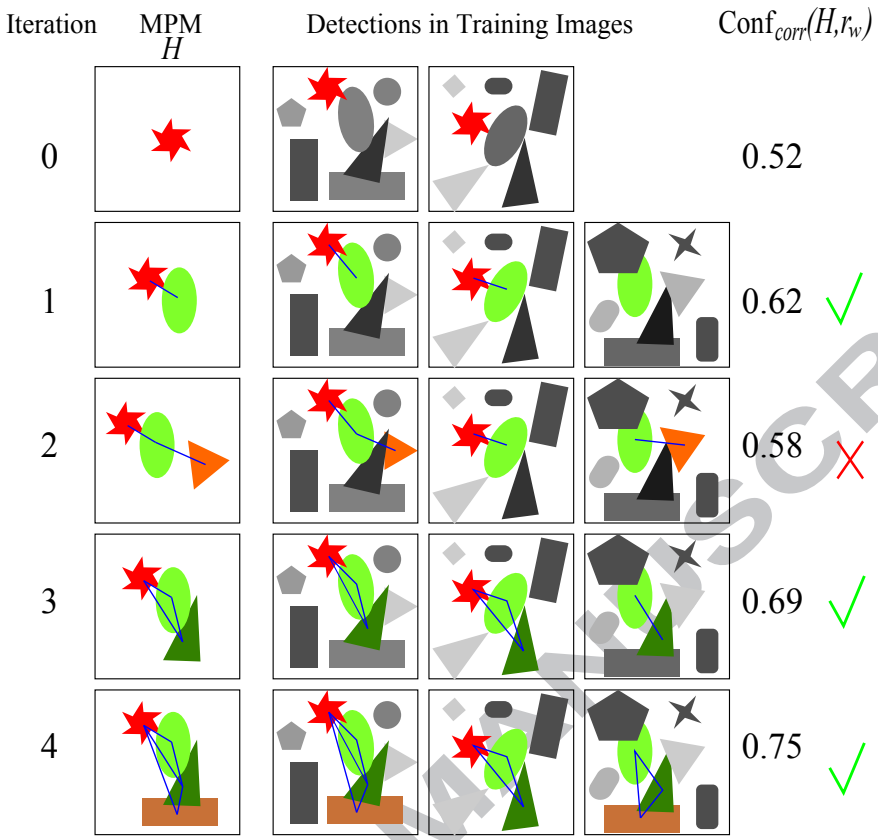


Figure 6: The system identifies part models that tend to co-occur with a consistent spatial relationship with current MPM detections and attempts to add them to the MPM. Changes are accepted if they improve $\text{Conf}_{corr}(H, w)$.

As illustrated in Figure 6, each iteration of the expansion algorithm begins by detecting all instances of the current multipart model in the training set (section 5) and identifying additional parts that tend to co-occur with a particular spatial relationship relative to the multipart model. We expand the multipart model by adding new vertices (part models) and edges (spatial relationships) one at a time from among the candidate parts. An expansion of the multipart model H is accepted if it improves $\text{Conf}_{corr}(H, w)$ (starting a new iteration), and rejected otherwise. The expansion process continues until potential additions to H have been exhausted.

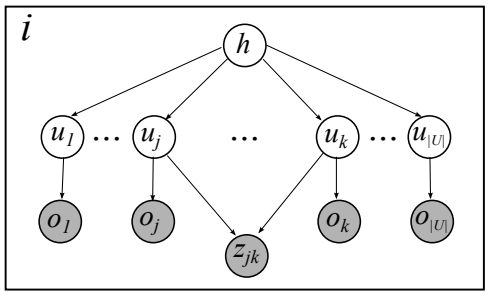


Figure 7: A graphical model of the generative process with multipart model indicator h, part indicators $\mathbf{u}$, part detection confidences $\mathbf{o}$ and observed spatial relations $\mathbf{z}$.

5. Detecting Multipart Models

As in part model detection, multipart detection must be robust to changes in viewpoint, occlusion or lighting that can cause individual part detections to be somewhat out of place or missing entirely. We use a simple generative model illustrated in Figure 7 to explain the pattern of part detections both in images that contain a particular multipart model and those that do not.

Each image i has an independent probability $P(h_i = 1)$ of containing the multipart model H. Given h_i, the presence of each model part is determined independently ($P(u_{ij} = 1|h_i)$). The foreground probability, $P(u_{ij} = 1|h_i = 1)$, is the likelihood of a model part being present when the model is present; it is set to a relatively high value (.95). This reflects our desire for an MPM to represent a recurring configuration of parts rather than possibly a disjunction of stable part configurations. The background probability, $P(u_{ij} = 1|h_i = 0)$, is the likelihood of a model part being present when the model is not; it is equal to the part's normalized frequency across the training image set. If a part is present ($u_{ij} = 1$), it tends to have a higher observed part detection confidence, o_{ij}, than if it is not present ($u_{ij} = 0$). We therefore set $p(o_{ij}|u_{ij} = 1) \propto o_{ij}$, and $p(o_{ij}|u_{ij} = 0) \propto (1 - o_{ij})$. If the multipart model is present ($h_i = 1$) and contains an edge d_{jk}, and the parts u_{ij} and u_{ik} are present, then the observed spatial relationship z_{ijk} between the two parts has a relatively narrow distribution centered at the spatial relationships encoded in the edge, d_{jk}. Otherwise, all spatial relationships follow a broad background distribution.

The MPM detection confidence, $\text{Conf}_{detect}(i, H)$ is the probability of the MPM H being present, given its part detection confidences $\mathbf{o}_i$ and observed spatial relations $\mathbf{z}_i$:

$$\text{Conf}_{detect}(i, H) = P(h_i = 1|\mathbf{o}_i, \mathbf{z}_i) \tag{4}$$

In any given image, there may be many possible assignments between multipart model vertices and observed part detections. We choose assignments in a greedy fashion in order to maximize $P(h_i = 1|\mathbf{o}_i, \mathbf{z}_i)$. First we choose the best-fit assignment of two linked vertices, then one by one we choose the vertex assignment that makes the largest improvement in $P(h_i = 1|\mathbf{o}_i, \mathbf{z}_i)$ and is consistent with existing assignments.

The prior probability $P(h_i = 1)$ depends on the complexity of the MPM, with more complex multipart models having a lower prior probability. Specifically:

$$P(h_i = 1) = \alpha^{|U|} \cdot \beta^{|D|} \tag{5}$$

where $\alpha, \beta < 1$ and $|U|$ and $|D|$ are, respectively, the number of vertices and edges in H. The constants α and β were selected based on detection experiments on random synthetic MPMs with a wide range of sizes. This prior is designed to prevent the detection of large, complete models when only a small subset of vertices is present, helping to ensure that MPMs represent a single view of the named object, rather than a set of loosely connected views.

6. Results on Annotation

Once we have discovered a set of individual part models and learned multipart models from configurations of those parts, we can use these learned structures to annotate new images. For ease of comparison, we run our system on the three image sets used in [8] and described below.[2] In all three cases, the addition of MPM models, in conjunction with our new part initialization method, improves both the precision and the recall of annotation on new images compared to our earlier system. The extent of improvement appears to depend on the scale and degree of articulation of named objects.

[2] We created each of these benchmarks and have made them available to the community, as no such benchmarks exist, *i.e.*, benchmarks consisting of cluttered scenes containing multiple, possibly occluding objects, where objects are not localized by bounding boxes and captions (labels) are noisy (words may or may not refer to objects in scene and objects in scene may or may not be named in caption).

We first describe our annotation method, datasets, and parameter settings, and in subsequent subsections present detailed results on annotating each dataset.

6.1. Annotation Method, Datasets, and System Parameters

To annotate an image, we select all words whose annotation confidence is sufficiently high for that image. We begin by detecting all part models in the image, including those that are relatively weakly detected or have relatively low individual correspondence confidence. Based on these part observations, we then evaluate detection confidence for all learned MPMs, as well as the correspondence confidence between the detected MPMs and words. As in our earlier work, the annotation confidence of both parts and multipart models, for a word and an image, is the product of detection confidence and correspondence confidence; *i.e.*, for MPMs, this is $\text{Conf}_{detect}(i, H) * \text{Conf}_{corr}(H, w)$. Overall annotation confidence for word w in image i is the maximum annotation confidence over w's detected models in i. A word is included in the image annotation if its overall annotation confidence reaches a user-defined threshold. For the results in this paper, we use a threshold of 95%.

We apply this annotation method to three datasets. The TOYS dataset is a small collection of images of groups of children's toys, first introduced in [25]. The much larger HOCKEY dataset was presented in [26] contains images and full-sentence captions of professional hockey teams in action. Finally, in [8], we introduced the LANDMARK dataset, composed of thousands of tourist photos and associated tags of famous structures throughout the world. These three datasets are described in more detail and with annotation results in the following three subsections.

Our system has a number of parameters whose values must be determined. In experimentation on the small TOYS image set, we find that the particular values of our system parameters do not have a substantial effect on our results. The same parameter values chosen based on the TOYS dataset are carried over to the two larger, real-world datasets without modification. We set uniqueness factors $\psi_b = 0.9$ and $\psi_s = 1.2$ (see Section 4.1). $N_{pairs} = 50$ (in Algorithm 1) allows a large number of failed pair samples before ending initial model search. $\nu = 0.75$ allows Q_{wi} to build gradually (see Section 4.2). We set the maximum number of MPMs per word, $N_{MPM} = 25$ (in Algorithm 2), to be more than the number of distinct views available for individual objects in these image collections. Finally, we choose MPM detection parameters $\alpha = 0.25$ and $\beta = 0.33$ (Section 6) based on experiments on synthetic data.

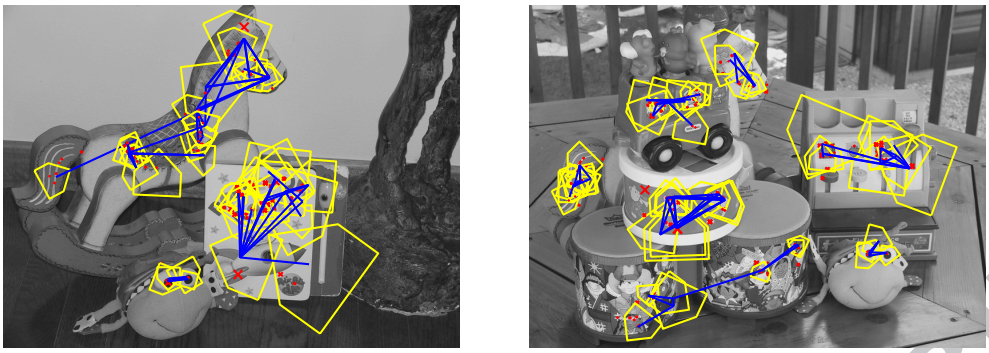
We compare two versions of our new system to the earlier system proposed in [8], referred to here as PARTS. PARTS used the old initialization method and only singleton part models, rather than MPMs. The first version of our new system uses just the new part initialization along with our earlier singleton part models; this is referred to in the results figures as PARTS+. The second one uses both the new part initialization and the MPMs in addition to singletons; this is referred to in the results figures as MPMS+. Testing both of these versions of the system allows us to evaluate the contribution of each change (initialization and MPMs) to the performance of the new system.

Annotation results are evaluated primarily in terms of precision and recall of image labels detected at various confidence levels. Tables and summary results are given for a 95% annotation confidence threshold while precision-recall curves display results for the full range of possible confidence thresholds. For each of the three datasets (in Sections 7.2, 7.3, and 7.4 respectively), we provide a figure comparing the performance of the PARTS, PARTS+, and MPMS+ systems. We also give the performance of the MPMS+ system in a table for each dataset reporting the per-object-name precision and recall. In each of these three tables, the Frequency column shows the number of captions within the test set that contain at least one instance of the given word (the Name). All of the per-object-name precision and recall values we report are based on the occurrence of the name in the captions of the test set; if the system does not detect an object for a word that appears in the caption, that instance is counted as a false negative, even if the named object does not actually appear in the image.

6.2. Experiments on the TOYS Dataset

This section examines the annotation performance of our new system on the TOYS dataset. This dataset contains images of 10 named toys, posed in groups so that no image contains fewer than 3 toys. There are 128 training images and results are evaluated on 100 test images.

Figure 8 displays some example images from the test set along with the highest-confidence MPM for each object and the associated annotations produced by the MPMS+ system. Each MPM part is displayed as a yellow, five-sided figure indicating canonical position, orientation and scale. The underlying interest points for each part are drawn in red and the edges connecting MPM parts are drawn in blue. The detections illustrate how MPMs



(a) Horse, Rocket, Bug (b) Horse, Bus, Drum, Bongos, Cash, Bug

Figure 8: Sample detections of MPMs for objects in the TOYS test set. In these examples, all named objects were correctly detected.

are able to integrate many local patches of distinctive appearance into a single structure. However, the MPM coverage is uneven, with some areas of the objects covered with a large number of overlapping parts while coverage in other areas is relatively sparse.

Although MPMs do integrate local detections, Figure 9 indicates that they have only a minor effect on overall precision and recall in the TOYS dataset. The system with the new initialization only [PARTS+] improves recall by a small amount, with a slight loss of precision at lower recall levels. Adding the multipart models [MPMs+] corrects the precision while retaining most of the gains in recall. We attribute the relatively small impact of the changes in our new system to the already good performance of our earlier system [PARTS] on this dataset. Most of the remaining missed object detections are more difficult cases with a high degree of occlusion. Also, some of the objects (Bug, Bus, and Dino) are small enough that individual part models can cover most of the area of distinctive appearance.

Table 1 shows the per-object precision and recall values of the MPMs+ system. Overall, our system achieves about a 3% improvement in recall on the TOYS set over our previous approach in the PARTS system. As in past evaluations, the two books (Franklin and Rocket), which have large, detailed planar surfaces, were easiest to detect. The two most difficult objects (Dino and Ernie) are notable for their curved surfaces and lack of distinctive fine-scale texture. That said, most of the recall improvement was due to a roughly three-fold increase in recall for the Dino object.

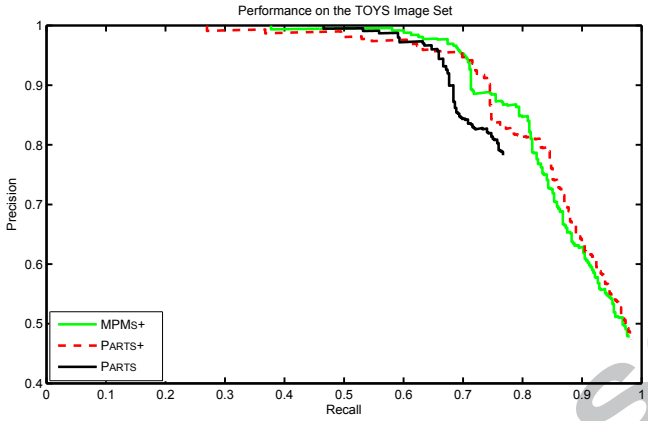
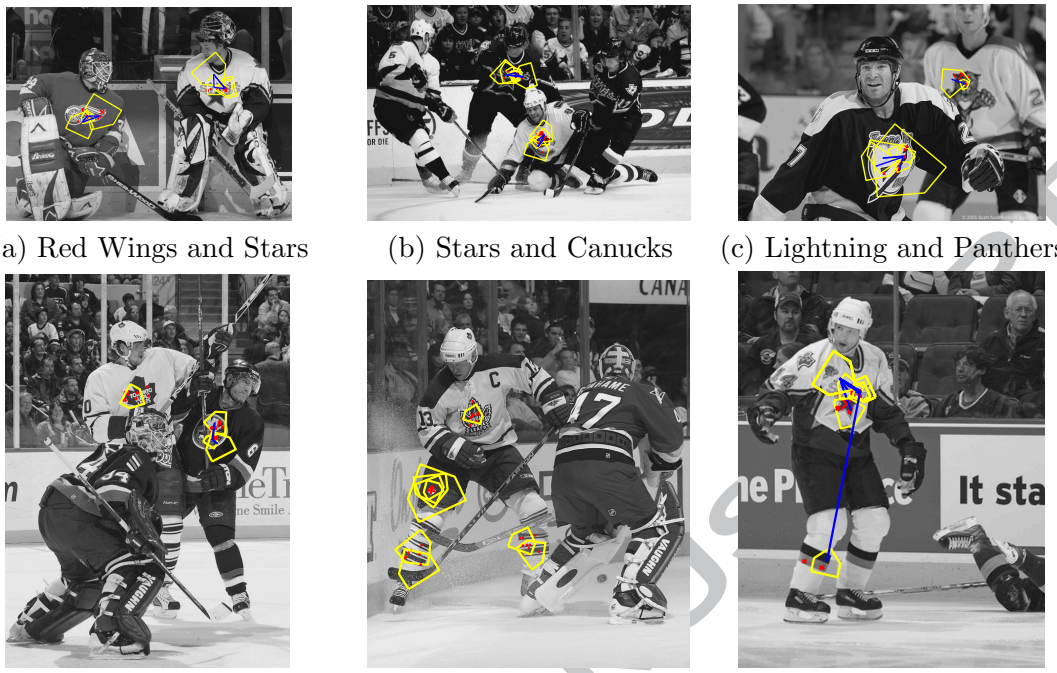


Figure 9: A comparison of precision–recall curves over the TOYS test set, for three systems: the new initialization method using singleton models only, PARTS+; the new initialization method with multipart models, MPMS+; and the system described in [8], PARTS. The new initialization method [PARTS+] improves recall somewhat over the earlier system [PARTS], while the addition of MPMs [MPMS+] corrects a slight drop in precision shown by the PARTS+ system.

Name	Precision	Recall	Frequency
Franklin	1.00	0.88	33
Rocket	1.00	0.80	44
Drum	1.00	0.69	32
Bus	1.00	0.51	57
Bongos	1.00	0.50	36
Bug	1.00	0.49	51
Dino	1.00	0.38	42
Ernie	1.00	0.28	39
Cash	0.97	0.78	46
Horse	0.96	0.86	28

Table 1: Per-object-name precision and recall of the MPMS+ system on the TOYS test set; mean precision = 99%, mean recall = 60%.



(a) Red Wings and Stars (b) Stars and Canucks (c) Lightning and Panthers

(d) Maple Leafs and Islanders (e) Maple Leafs (f) Stars

Figure 10: Sample detections of objects in the HOCKEY test set.

6.3. Experiments on the HOCKEY Dataset

The HOCKEY set includes 2526 images of National Hockey League (NHL) players and games, with associated captions, downloaded from a variety of sports websites. It contains examples of all 30 NHL teams, and is divided into 2026 training and 500 test image–caption pairs. About two-thirds of the captions are full sentence descriptions, whereas the remainder simply name the two teams involved in the game.

Figure 10 shows sample multipart model detections on test-set images and the associated team names. Compared to MPMs in the TOY and LANDMARK sets, most MPMs in the HOCKEY set are relatively simple. They typically consist of 2 to 4 parts clustered around the team's chest logo. Since the chest logos are already reasonably well covered by individual part models, there is little reward for developing extensive MPMs. In principle, MPMs could tie together parts that describe other sections of the uniform (socks, pants, shoulder insignia) like those shown in Figure 10(e), but this type of MPM (seen in Figure 10(f)) is quite rare. There may be too much articulation and

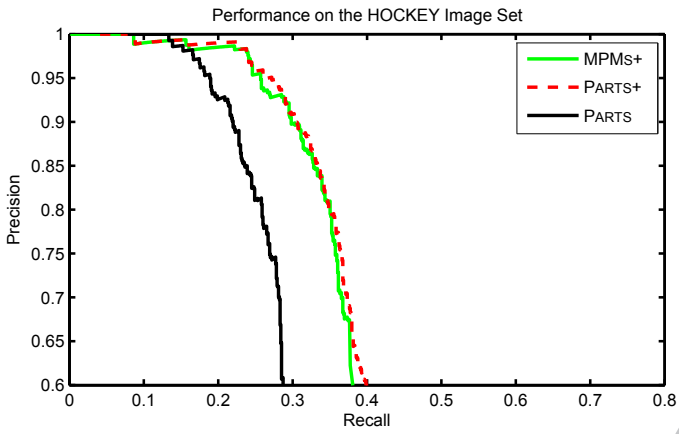


Figure 11: A comparison of precision–recall curves over the HOCKEY test set, for three systems: the new initialization method using singleton models only, PARTS+; the new initialization method with multipart models, MPMs+; and the system described in [8], PARTS. The new initialization method alone, in PARTS+, substantially improves overall recall. However, the addition of MPMs in MPMs+ has little effect. This is probably because the distinctive portions of a player's appearance are of limited size and do not tend to co-occur in repeating patterns.

(perhaps more importantly) too few instances of co-occurrence of these parts in the training set to support such MPMs.

Figure 11 indicates that our new approach for initializing part models leads to an improvement in recall of about 10-12 percentage points. Considering the barriers to achieving high recall on the HOCKEY set (discussed in [8]), this represents a substantial gain. Our system with the new initialization [PARTS+] is better able to identify regions of distinctive appearance than the PARTS system. For instance, one of the best-recognized NHL teams using our new method was completely undetected in our earlier work. On the other hand, the addition of MPMs in the MPMs+ system does not improve annotation performance at all. This is probably due to the relatively small size of distinctive regions in the HOCKEY images combined with a degree of articulation and occlusion that make larger models unreliable.

Table 2 shows the annotation performance of the MPMs+ system with respect to individual team names. The system has high-confidence detections for 27 of the 30 teams, 4 more than with the PARTS system. At 95% precision, overall recall is 26%, 12% higher than the previous method. For example, the Washington Capitals are one of the better-recognized teams whereas the

Name	Precision	Recall	Frequency
Tampa Bay Lightning	1.00	0.61	49
Pittsburgh Penguins	1.00	0.45	29
Minnesota Wild	1.00	0.37	35
Washington Capitals	1.00	0.35	17
Los Angeles Kings	1.00	0.31	36
Dallas Stars	1.00	0.29	42
Detroit Red Wings	1.00	0.26	42
San Jose Sharks	1.00	0.26	23
Buffalo Sabres	1.00	0.25	32
Calgary Flames	1.00	0.23	26
Columbus Blue Jackets	1.00	0.18	11
Philadelphia Flyers	1.00	0.17	46
Carolina Hurricane	1.00	0.17	30
New York Rangers	1.00	0.14	42
Montreal Canadiens	1.00	0.13	23
Colorado Avalanche	1.00	0.09	23
Anaheim Ducks	1.00	0.08	27
Vancouver Canucks	1.00	0.05	40
New York Islanders	0.96	0.45	60
Toronto Maple Leafs	0.92	0.33	73
New Jersey Devils	0.89	0.29	59
Florida Panthers	0.88	0.28	25
Ottawa Senators	0.88	0.12	58
Chicago Blackhawks	0.86	0.34	35
Nashville Predators	0.83	0.25	20
Atlanta Thrashers	0.80	0.23	35
Boston Bruins	0.75	0.18	17

Table 2: Per-object-name precision and recall of the MPMs+ system on 27 (of 30) team names detected with high confidence in the HOCKEY test set; mean precision = 95%, mean recall = 26%.

earlier system had not detected them in the test set at all. This may be due to the new system's focus in initialization on images sharing particular caption words.

6.4. Experiments on the LANDMARK Dataset

The LANDMARK dataset includes images of 27 famous buildings and locations with some associated tags downloaded from the Flickr website, and randomly divided into 2172 training and 1086 test image–caption pairs. Like the NHL logos, each landmark appears in a variety of perspectives and scales. Compared to the hockey logos, the landmarks usually cover more of the image and have more textured regions in a more stable configuration. On the other hand, the appearance of the landmarks can vary greatly with viewpoint and lighting and many of the landmarks feature interior as well as exterior views.

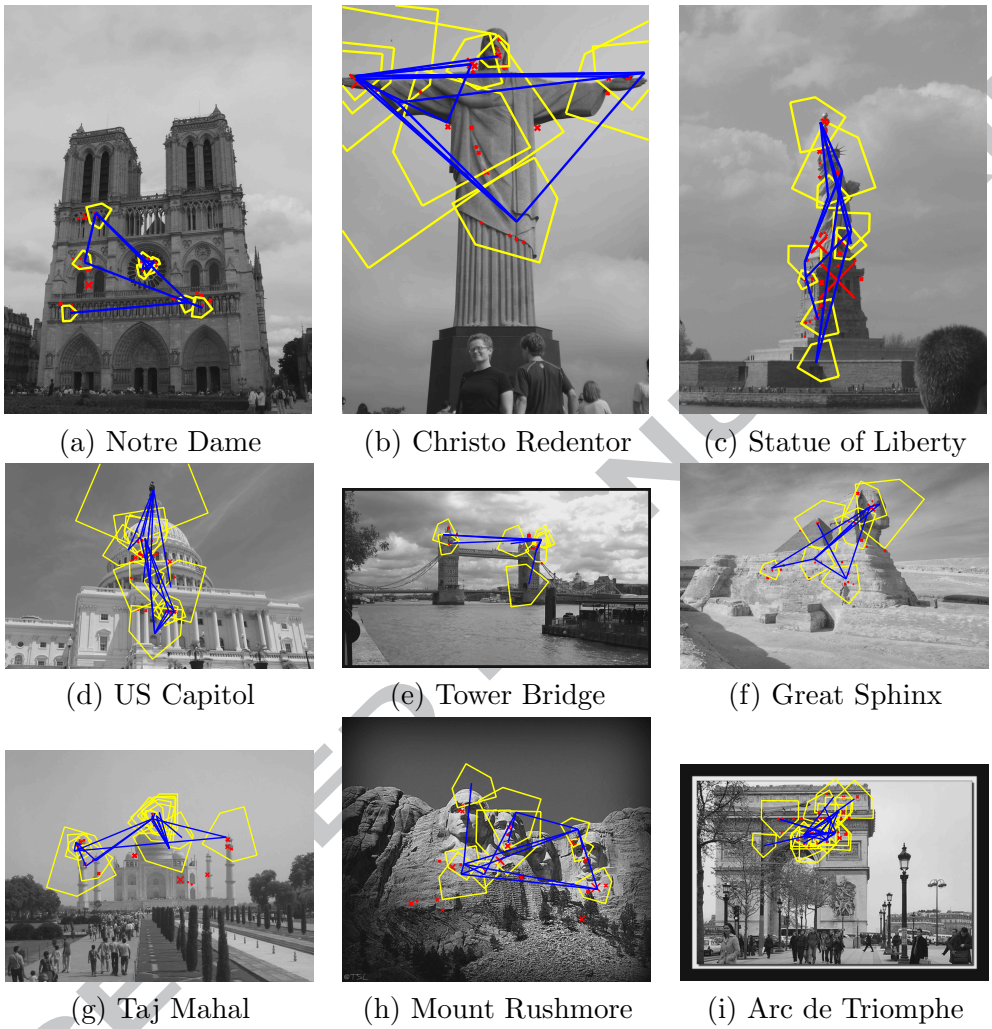
Figure 12 provides some sample detections of multipart models in the LANDMARK test set. In this dataset, MPMs appear in many cases to capture some of the distinctive overall part structure of the objects. The MPMs can integrate widely-separated part detections, thereby significantly improving detection confidence and localization. Taken together, individually uncertain part detections often form an unambiguous whole.

The detailed results also indicate potential areas for further improvement. Some of the models display a high degree of part overlap, especially on objects such as the Arc de Triomphe with a dense underlying array of distinctive features. With MPMs, coverage of the object is much better than that of individual parts, but coverage is not always complete. For instance, the system detects many more parts on the western face of Notre Dame than are incorporated into the displayed MPM. In the future, we could address this by modifying the MPM training routine to explicitly reward spatial coverage improvements. Finally, MPMs often seem to have one or two key parts with a large number of long-range edges. Changes to encourage a more local connection structure could further improve robustness to occlusion.

As indicated by Figure 13, MPMs can significantly improve annotation precision. The new initialization system in PARTS+ improves overall recall by about 10%, and the addition of MPMs in MPMS+ lifts the precision of the curve towards the 100% boundary. In contrast to the performance on the other datasets, the MPMS+ system shows a very marked improvement over the PARTS+ system with new initialization alone. We suggest that this is because a view of a landmark is better represented as a configuration of parts, rather than independent elements. Architectural elements may be shared across many buildings, but the ensemble is more distinctive.

Table 3 breaks the results down by landmark. The structures on which the MPMS+ system achieved the poorest results were St. Peter's Basilica, Chichen Itza, and the Sydney Opera House. (All three reach 100% precision but have very low recall.) The first two of these suffer from a multiplicity of viewpoints, with training and test sets dominated by a variety of interior viewpoints and zoomed images of different parts of the structure. The Sydney Opera House's expressionist design has relatively little texture and is therefore harder to recognize using local appearance features.

Figure 14 illustrates the effect of training set size on the ability of the system to learn landmark appearance. The complete training set has 2172 image–caption pairs, or about 80 images per landmark, distributed among multiple views. Reducing the training set size to 1400 reduces overall recall



(a) Notre Dame (b) Christo Redentor (c) Statue of Liberty

(d) US Capitol (e) Tower Bridge (f) Great Sphinx

(g) Taj Mahal (h) Mount Rushmore (i) Arc de Triomphe

Figure 12: Sample detections of objects in the LANDMARKS test set.

Name	Precision	Recall	Frequency
Leaning Tower	1.00	0.86	43
US Capitol	1.00	0.71	45
Golden Gate Bridge	1.00	0.67	45
Mount Rushmore	1.00	0.66	35
Notre Dame Cathedral	1.00	0.58	40
Great Sphinx	1.00	0.58	40
St. Paul's Cathedral	1.00	0.56	48
Statue of Liberty	1.00	0.56	36
Reichstag	1.00	0.49	45
Empire State Building	1.00	0.47	38
Burj Al Arab	1.00	0.44	43
Sagrada Familia	1.00	0.29	35
Colosseum	1.00	0.28	39
CN Tower	1.00	0.24	34
Parthenon	1.00	0.23	35
St. Peter's Basilica	1.00	0.15	41
Sydney Opera House	1.00	0.12	42
Chichen Itza	1.00	0.05	37
Arc de Triomphe	0.97	0.74	42
White House	0.97	0.67	45
Big Ben	0.97	0.64	44
Tower Bridge	0.97	0.55	47
Stonehenge	0.96	0.60	42
St. Basil's Cathedral	0.96	0.69	35
Taj Mahal	0.95	0.58	33
Eiffel Tower	0.89	0.48	33
Christo Redentor	0.84	0.61	44

Table 3: Per-object-name precision and recall of the MPMs+ system on the 30 structure names in the LANDMARK test set; mean precision = 98%, mean recall = 51%.

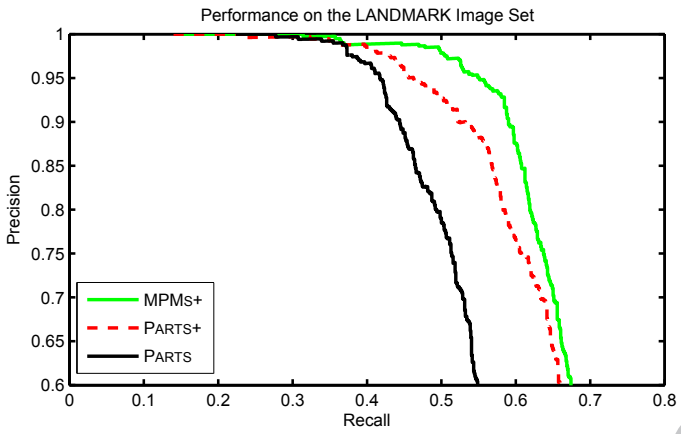


Figure 13: A comparison of precision–recall curves over the LANDMARKS test set, for three systems: the new initialization method using singleton models only, PARTS+; the new initialization method with multipart models, MPMS+; and the system described in [8], PARTS. The new initialization method [PARTS+] substantially improves overall recall. In this case, the addition of MPMs [MPMS+] improves the precision of the new detections. Distinctive portions of landmarks sometimes have stable relationships between one another.

by about 5%. Further reducing training size to 700 image–caption pairs (about 26 images per landmark) reduces recall by an additional 10%. This indicates that at this level, there are many landmark views that have too little representation in the training set to be effectively learned. On the other hand, a larger training set would provide enough examples of even relatively rare views, increasing overall recall.

Though the images of the LANDMARK dataset exhibit a large variation in viewpoint and time of day, the associated captions contain relatively little noise. Figures 15 and 16 indicate that the performance of the system decays gracefully when random caption noise is added. Adding 30% false captions or removing 30% of the true captions reduces recall by $5 - 8\%$. The MPM learning stage is not dependent on reliable labels.

7. Conclusions

Our initialization method and multipart models are designed to work together to improve annotation accuracy and object localization over our earlier approach in [8]. Our initialization mechanism boosts recall and part coverage by detecting potential parts that would have been overlooked by the

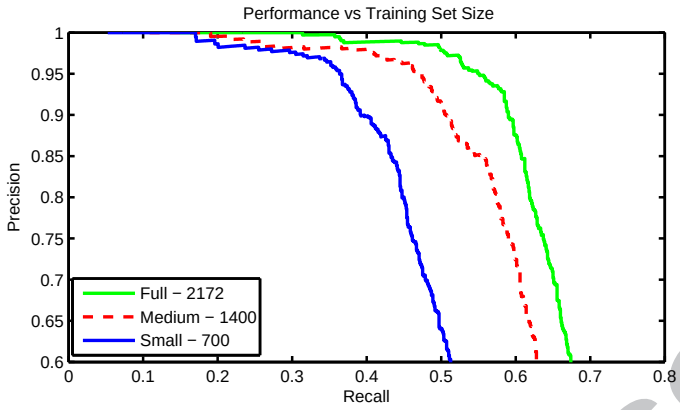


Figure 14: A comparison of precision–recall curves over the LANDMARKS test set, for three different training set sizes: the 'full' set of 2172 training image–caption pairs, a 'medium' subset of 1400 pairs and a 'small' set of 700 pairs.

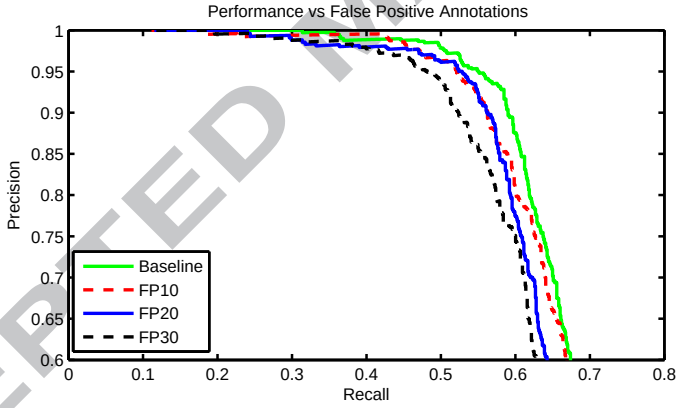


Figure 15: A comparison of precision–recall curves over the LANDMARKS test set, for four different levels of false positive caption noise. The 'baseline' results on trained on captions with relatively few, naturally-occurring false positive labels. The 'FP10', 'FP20' and 'FP30' results are trained on data where respectively 10, 20 and 30% of the captions have an extra random false-positive label inserted.

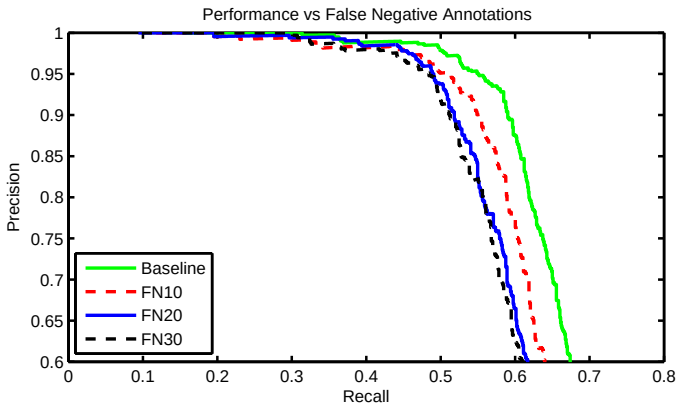


Figure 16: A comparison of precision–recall curves over the LANDMARKS test set, for four different levels of false negative caption noise. The 'baseline' results on trained on captions without added false negative labels. The 'FN10', 'FN20' and 'FN30' results are trained on data where respectively 10, 20 and 30% of the captions have the true image label removed.

previous system, thereby identifying more individually ambiguous parts and providing for a better distribution of parts over the image set. The MPMs boost precision and localization by integrating parts that may be individually ambiguous into models that can cover an entire view of an object.

Together, these two enhancements substantially improve annotation accuracy over previous results on the experimental datasets. Our improvements to part initialization and training have increased recall considerably, though sometimes at the expense of precision. For objects with recurring patterns of distinctive parts, the use of MPMs can filter out bad detections, resulting in a substantially improved precision–recall curve.

The annotation performance improvement due to MPMs depends strongly on the properties of the dataset. MPMs are most useful for combining multiple part models that, individually, may be too weak or ambiguous to be informative, but whose co-occurrence in repeatable spatial configurations provides strong evidence of some object. We speculate that MPMs do not improve results on the TOYS and HOCKEY sets because either the objects (or a sufficiently distinctive subregion) can be described effectively by a single part, or parts do not recur often enough in stable configurations (*e.g.*, hockey chest, shoulder and sock patterns). Landmarks, being larger and often with more ambiguous local structure, benefit more from MPMs. We expect that the use of MPMs will be even more important as individual parts are more

ambiguous (such as parts for describing object classes).

Our new initialization mechanism and the development of multipart models also improve object localization. The initialization approach ensures that parts have less spatial overlap than before, cover portions of the object that are less individually distinctive, and are better distributed across object views. The MPMs tie together recurring patterns of these parts, allowing us to distinguish between the presence of multiple parts and multiple objects. Future work could further improve localization by ensuring that MPMs use more of the available parts to maximize spatial coverage and are themselves well-distributed across object views.

References

[1] V. Ferrari, L. Fevrier, F. Jurie, C. Schmid, Groups of adjacent contour segments for object detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (2008) 36–51.

[2] M. Stark, M. Goesele, B. Schiele, A shape-based object class model for knowledge transfer, in: Twelfth IEEE International Conference on Computer Vision (ICCV), Kyoto, Japan, 2009.

[3] P. Felzenszwalb, D. McAllester, D. Ramanan, A discriminatively trained, multiscale, deformable part model, in: CVPR, 2008.

[4] R. Fergus, P. Perona, A. Zisserman, Object class recognition by unsupervised scale-invariant learning, in: CVPR, 2003.

[5] K. Barnard, P. Duygulu, D. Forsyth, N. de Freitas, D. Blei, M. Jordan, Matching words and pictures, Journal of Machine Learning Research 3 (2003) 1107–1135.

[6] G. Carneiro, A. Chan, P. Moreno, N. Vasconcelos, Supervised learning of semantic classes for image annotation and retrieval, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (3) (2007) 394–410.

[7] P. Carbonetto, N. de Freitas, K. Barnard, A statistical model for general contextual object recognition, in: ECCV, 2004.

[8] M. Jamieson, A. Fazly, S. Dickinson, S. Stevenson, S. Wachsmuth, Using language to learn structured appearance models for image annotation,

IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (1) (2010) 148–164.

[9] F. Monay, D. Gatica-Perez, Modeling semantic aspects for cross-media image indexing, IEEE Trans. Pattern Analysis and Machine Intelligence 29 (10) (2007) 1802–1817.

[10] A. Quattoni, M. Collins, T. Darrell, Learning visual representations using images with captions, in: CVPR, 2007.

[11] R. Brooks, Model-based 3-D interpretations of 2-D images, IEEE Transactions on Pattern Analysis and Machine Intelligence 5 (2) (1983) 140–150.

[12] S. Dickinson, A. Pentland, A. Rosenfeld, 3-D shape recovery using distributed aspect matching, IEEE Transactions on Pattern Analysis and Machine Intelligence 14 (2) (1992) 174–198.

[13] K. Siddiqi, A. Shokoufandeh, S. J. Dickinson, S. W. Zucker, Shock graphs and shape matching, International Journal of Computer Vision 35 (1) (1999) 13–32.

[14] A. Shokoufandeh, L. Bretzner, D. Macrini, M. Demirci, C. Jonsson, S. Dickinson, The representation and matching of categorical shape, Computer Vision and Image Understanding 103 (2006) 139–154.

[15] R. Fergus, L. Fei-Fei, P. Perona, A. Zisserman, Learning object categories from Google's image search, in: CVPR, 2005.

[16] D. J. Crandall, D. P. Huttenlocher, Weakly supervised learning of part-based spatial models for visual object recognition, in: ECCV, 2006.

[17] I. Kokkinos, A. Yuille, HOP: Hierarchical object parsing, in: CVPR, 2009.

[18] L. Zhu, C. Lin, H. Huang, Y. Chen, A. Yuille, Unsupervised structure learning: Hierarchical recursive composition, suspicious coincidence and competitive exclusion, in: ECCV, 2008.

[19] G. Bouchard, B. Triggs, Hierarchical part-based visual object categorization, in: CVPR, 2005.

[20] S. Fidler, M. Boben, A. Leonardis, Similarity-based cross-layered hierarchical representation for object categorization, in: CVPR, 2008.

[21] B. Epshtein, S. Ullman, Feature hierarchies for object classification, in: ICCV, 2005.

[22] B. Ommer, J. Buhmann, Learning the compositional nature of visual object categories for recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (3) (2010) 501–516.

[23] D. G. Lowe, Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision 60 (2) (2004) 91–110.

[24] Y. Ke, R. Sukthankar, PCA-SIFT: A more distinctive representation for local image descriptors, in: CVPR, 2004.

[25] M. Jamieson, S. Dickinson, S. Stevenson, S. Wachsmuth, Using language to drive the perceptual grouping of local image features, in: CVPR, 2006.

[26] M. Jamieson, A. Fazly, S. Dickinson, S. Stevenson, S. Wachsmuth, Learning structured appearance models from captioned images of cluttered scenes, in: ICCV, 2007.

- We learn to recognize exemplars from unstructured collections of captioned images.
- Using language, we perceptually grouping local features into meaningful parts.
- We further group discovered parts into flexible hierarchical configurations.
- Learned visual structures are scale, translation and rotation invariant.
- Learning is robust to distractors, clutter, ambiguous and incomplete captions.