Flexible Object Models for Category-Level 3D Object Recognition

Akash Kushal, Cordelia Schmid, Jean Ponce

Flexible Object Models for Category-Level 3D Object Recognition

Akash Kushal[1,3]
Computer Science Department[1]
and Beckman Institute
University of Illinois
Urbana-Champaign, USA

Cordelia Schmid[2]
LEAR Team[2]
INRIA
Montbonnot, France

Jean Ponce[3,1]
WILLOW Team–ENS/INRIA/ENPC[3]
Département d'Informatique
Ecole Normale Supérieure
Paris, France

Abstract

Today's category-level object recognition systems largely focus on fronto-parallel views of objects with characteristic texture patterns. To overcome these limitations, we propose a novel framework for visual object recognition where object classes are represented by assemblies of partial surface models *(PSMs) obeying loose local geometric constraints. The PSMs themselves are formed of dense, locally rigid assemblies of image features. Since our model only enforces* local *geometric consistency, both at the level of model parts and at the level of individual features within the parts, it is robust to viewpoint changes and intra-class variability. The proposed approach has been implemented, and it outperforms the state-of-the-art algorithms for object detection and localization recently compared in [14] on the Pascal 2005 VOC Challenge Cars Test 1 data.*

1. Introduction

Object recognition—or, in a broader sense, scene understanding—is the ultimate scientific challenge of computer vision. After 40 years of research, robustly identifying the familiar objects (chair, person, pet) and scene categories (beach, forest, office) depicted in family pictures or news segments is still far beyond the capabilities of today's vision systems. Despite the limitations of current scene understanding technology, tremendous progress has been accomplished in the past five years, due in part to the formulation of object recognition as a statistical pattern matching problem. The emphasis is in general on the features defining the patterns and the machine learning techniques used to learn and recognize them, rather than on the representation of object and scene categories, or the integrated interpretation of the various scene elements. Modern pattern-matching approaches largely focus on fronto-parallel views of objects with characteristic texture patterns, and they have proven successful in that domain for images with moderate amounts of clutter and occlusion. Most methods represent object classes as assemblies of salient *parts*—that is, (groups of) image features whose appearance remains stable over exemplars. By and large, geometric constraints among parts are either completely ignored (bag-of-parts models [10, 19]), or imposed in a rigid manner (constellation/star models [3, 11, 12]). We believe that, as demonstrated by others in the *specific* object recognition domain [9, 16], geometric constraints are just too powerful to be ignored. For object categories without characteristic textures (e.g., cows, people, etc.), they are also the main image cues available. On the other hand, rigid assemblies of features [3, 11, 12] cannot accommodate the image variability due to significant changes in viewpoint or shape within a category.

In this paper, we propose a novel object model based on the following observation: Even though the geometric relationship between "distant" parts of an object may vary due to intra-class variability and changes in viewpoint, the relative affine transformations among nearby parts are robust to these factors (this is related to the well known fact that arbitrary smooth deformations —including those induced by viewpoint changes for affine cameras or perspectives ones far from the scene relative to its relief— are locally equivalent to affine transformations [8]).

Thus, we represent object parts as *partial surface models* (or *PSMs*) which are *dense, locally rigid* assemblies of texture patches. These PSMs are learned by matching repeating patterns of features across training images of each object class (Section 3). Pairs of PSMs which regularly occur near each other at consistent relative positions are linked by edges whose labels reflect the local geometric relationships between these features. These local connections are used to construct a probabilistic graphical model for the geometry and appearance of the PSMs making up an object (Section 4). In turn, the corresponding *PSM graph* is the basis for an effective algorithm for object detection and localization (Section 5), which outperforms the state-of-the-art methods recently compared in [14] on the Pascal 2005 VOC Challenge Cars Test 1 data (Section 6).

2. Related Work

Fergus et al. [3] model the joint probability distribution of the relative positions of object parts (which are individual features in their case) as a Gaussian distribution with a full covariance matrix. However, they model each part's location using its x, y image coordinates, making the model highly viewpoint specific. In addition, the learning time is exponential in the number of parts, which limits the number of parts in the model to a maximum of 6 or 7. Loeff et al. [12] propose a "star" model in which an object generates a number of parts, each of which generates features with a certain appearance and location relative to the object center. The appearance of the features is modeled as a multinomial distribution over a codebook of feature types and the relative location is modeled as a Gaussian with a mean and covariance for each part. The appearance model of a single part is rather weak in this case since all the features for a given part are generated with independent locations relative to the part center. In addition, the star model enforces rigid geometric constraints among the different object parts and so is viewpoint dependent. Leibe et al. [11] construct implicit shape models (ISMs) by clustering object features in the training images, and storing for each cluster center the location of the object center and scale relative to the corresponding feature. During recognition, each feature detected on the test image essentially casts probabilistic votes for object centers and scales, and a mean shift procedure is used to find the maxima in this space. The geometric model is again rigid, hence highly viewpoint dependent, and the appearance model is relatively weak since each feature occurrence is considered independently of all others.

Lazebnik et al. [10] propose a bag-of-parts model in which the parts are composed of multiple features linked together in an affinely rigid structure. These parts are quite discriminative and relatively stable against intra-class variations; however, large viewpoint variations result in the affine model no longer holding for the object parts. In this paper, we learn object parts by only enforcing *local* geometric consistency among the features that make up the part. We also augment the "primary" interest point features of [10] with more general "secondary" texture patches to generate dense and highly discriminative PSMs. Using dense models as opposed to just interest points has been shown to improve matching and recognition performance by [5, 9] in the context of specific 3D object recognition. Dense PSM matches provide an extremely stable coordinate frame to compute the relative positions of other PSM matches robustly. Since our PSMs consist of multiple overlapping features, and some features are more discriminative than others we train a logistic regression model to evaluate the quality of a PSM match based on the individual feature matches.

Recently, Thomas et al. [18] have proposed a technique which deals with the viewpoint change problem by combining models learned for different viewpoints. They use a highly supervised dataset that consists of images of multiple motorbike instances, each from a set of up to 16 viewpoints. First, they construct separate viewpoint dependent ISMs [11] for each of the different viewing directions. Then, they use the method of [4] to match the images of the same motorbike instance across different viewpoints and construct region tracks that are later used to transfer the ISM votes from one viewpoint to its neighboring viewpoints. This is an interesting setup, but it requires highly supervised training data and a dense sampling of viewpoints since the ISMs themselves are highly viewpoint dependent. In addition, since each of the ISMs is learned independently, there is no sharing of parts among the different viewpoints. In contrast, we use a single model that shares its parts (PSMs) among different viewpoints.

3. Learning PSMs

We use a *hypothesize and validate* approach to learn PSMs, similar to [10]. A critical difference, however, is that the (relatively) sparse and affinely rigid parts of [10] are replaced by *dense* and *locally* rigid PSMs.

3.1. PSM Formation

The learning process starts by selecting two images at random from the training set and computing appearance-based *primary matches* between pairs of salient image regions. To avoid an excessive reliance on characteristic texture patterns, we use a simple operator (essentially a Hough transform) to detect circles in edge maps [7], and output the smallest squares enclosing them as candidate regions. Candidate matches between these regions are then computed using the SIFT operator [13], and a non-linear refinement process is used to correct the initial alignment of these matches. This process considers the patch in the second image as a deformable parallelogram, and optimizes the affine transformation mapping the first patch onto its match so as to minimize the error between the SIFT descriptors of the matched patches. This is essential for matching images of the same patch viewed from different directions.

• **Initialization:** Once candidate matches have been found, they are partitioned into locally consistent *PSM hypotheses* using a greedy approach: A PSM hypothesis is initialized with a single match, and nearby matches are iteratively added until the corresponding affine transformations are no longer close enough to those associated with nearby matches already in the PSM hypothesis. An unused match is then chosen at random, and a new PSM hypothesis is grown using the leftover candidates. This process partitions the matches into a set of PSM hypotheses. The hypotheses that contain more matches than a given threshold are passed along to the expansion stage. The composition of the large

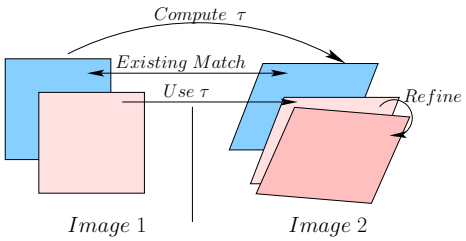


Figure 1. Expansion: match prediction and refinement.

PSM hypotheses is almost unaffected by the choice of the match used for initialization.

• **Expansion:** The first image of the training image pair is covered with overlapping square-shaped "secondary" patches, and an expansion step is used to densely cover a region surrounding the PSM hypothesis. Each match in the current PSM hypothesis tries to expand into nearby secondary patches by predicting a match for the secondary patch based on its own affine transformation τ. The "best" predicted match (in terms of SIFT matching score) for each secondary patch is then refined using a non-linear refinement process similar to the one used to align the initial matches. However, now the refinement process also penalizes the deviation of the refined location from the predicted location to prevent the match from drifting too far from its predicted location.[1] Finally, the secondary match is added to the PSM hypothesis if the SIFT matching score exceeds a given threshold. The process is illustrated in Fig. 1. The algorithm continues to expand around the newly added matches until no more secondary matches can be added. The PSM hypothesis consists of all the patches in the first image that were matched successfully.

3.2. PSM Matching

A PSM u can be matched to a target image I using a similar process. First, the circle detector is used to detect candidate regions in I and appearance-based matches are computed between the primary patches in u and the detected regions in I. Next, an expansion step uses these initial matches to hypothesize matches for the nearby unmatched patches of u. The hypothesized matches are first refined (as described before) and are iteratively added to the current matches as long as their matching score exceeds a threshold. Finally, all the feature matches are partitioned into groups of PSM matches in the same way as was done for the initially selected image pair. If the number of patch matches within a PSM match exceeds a certain ratio ($R = 0.5$) of the total number of patches in the corresponding PSM, it is considered correct.

[1]Since match refinement is relatively time consuming, it is done once for the "best" expansion attempt for each secondary patch, instead of doing it for each expansion attempt and choosing the best match later.

Figure 2. Left column: the two base images for a PSM; Right: a few validation matches for the same PSM. The outline of the PSM, defined as the union of matched feature regions, is shown in white.

3.3. PSM Validation and Selection

The PSM hypotheses learned in Section 3.1 are scored by matching them to a set of validation images containing the object as well as a set of background images. Let N_u^+ (resp. N_u^-) be the number of times a PSM candidate u is matched in the validation (resp. background) images. We compute the discriminative power of a PSM u as the ratio $\mathcal{R}_u = N_u^+/N_u^-$ and use it to select PSMs for the object model. The PSM candidates are processed in a decreasing order of $\mathcal{R}_u$ and are selected only if their validation matches do not have a significant overlap with those of previously selected PSMs. This process helps avoid repeated PSMs. Figure 2 shows a PSM found in training images from the PASCAL VOC 2005 dataset. The two images on the left of the figure are the base images used to hypothesize the PSM, and the images on the right are some validation images.

3.4. PSM Appearance Model

We train a logistic regression model to evaluate the quality of individual detections (PSM matches). Concretely, we attach to each match m of a PSM u consisting of n features, a binary *appearance* vector $a = (a_1, \ldots, a_n)$, where a_i is equal to 1 if the corresponding feature has been matched, and -1 otherwise. We also associate a label $\ell \in \{\text{obj}, \text{bkg}\}$ with every match m for a PSM u based on whether it matches the object part corresponding to u or some background texture in the image. Let us define $P_u(a|\ell)$ as the probability that a PSM match of u has appearance vector a given that it has label ℓ. Since a PSM consists of multiple overlapping features, the individual feature matches (the components of a) are not independent, making their probabilistic modeling difficult. Thus, instead of learning $P_u(a|\ell)$ directly, we learn a parametric model for $\frac{P_u(a|\text{obj})}{P_u(a|\text{bkg})}$, which will prove sufficient for object detection.

To simplify the learning task, we assume that the training data is generated from a joint distribution $P_u(a, \ell)$ as follows: First, the label is generated from $P(\ell = \text{obj}) = P(\ell = \text{bkg}) = 0.5$ and then the appearance a is generated given the label from the probability distribution $P_u(a|\ell)$. We begin by training a logistic regression classifier on this data to construct the probability distribution $P_u(\ell|a)$. This

now allows us to compute the desired probability ratio as

$$\frac{P_u(a|\text{obj})}{P_u(a|\text{bkg})} = \frac{P_u(\text{obj}|a)P_u(a)}{P_u(\text{obj})} \times \frac{P_u(\text{bkg})}{P_u(\text{bkg}|a)P_u(a)} = \frac{P_u(\text{obj}|a)}{P_u(\text{bkg}|a)}.$$

We now describe the generation of the training data samples (a, ℓ) and the classifier training procedure.

• **Generating data from the joint distribution:** Every match m of the PSM u provides us with a data point, labeled as obj or bkg depending on whether it has been found in a validation or background image. However, since the number of background instances available is typically much smaller than the number of object instances, we repeatedly sample the background matches as necessary to create as many data points labeled bkg as there are points labeled obj. We assume that the matches observed in the validation images correctly match the corresponding object parts and that the observed appearances a of the matches in the validation (resp. background) images are random samples from $P_u(a|\text{obj})$ (resp. $P_u(a|\text{bkg})$).

• **Training the classifier:** We train the logistic regression model to output the probability $P_u(\ell|a)$ that a data point has label ℓ given that it has appearance a. The binary features used by the logistic regression model are simply the components of a. The weights associated with the features of the model are learned so as to maximize the log-likelihood of labels observed on the training data. This is a convex optimization problem and can be solved efficiently. Since the amount of training data is limited we regularize the maximum likelihood parameter learning process to avoid overfitting by adding a penalty proportional to the squared norm of the weight vector of the logistic regression model [17]. Once the PSMs and the corresponding logistic regression models have been learned, the matching process can essentially be thought of as running a set of PSM detectors that fire at certain locations in an image and provide an estimate of the quality of the match based on its appearance. As argued before, since these detectors enforce only local geometric consistency, they are robust to viewpoint changes and intra-class variability.

4. Learning Object and Background Models

4.1. The PSM Graph

We can associate with any instance u' of a PSM u detected in an image the 2D affine transformation which is the "mean" of the affine transformations mapping the individual patches of u in the base image roughly to their detected matches u'. Intuitively, this transformation represents the affine deformation of the PSM from its base image to its matched location. We denote by $\mathcal{A}_u$, the random variable associated with the affine deformation corresponding to PSM u, and we model the joint probability distribution of the variables $\mathcal{A}_u$ associated with all PSMs in an object model as a MRF (Markov Random Field) structure dubbed the *PSM graph*. The vertices of this graph are identified with the random variables $\mathcal{A}_u$. Nearby PSMs are linked by edges that enforce local consistency between them. An edge joins two PSMs when they co-occur within a specified range from each other in a sufficient number of validation images. Figure 3 shows the PSM graph model for a car learned from the PASCAL VOC 2005 dataset using the technique described in the rest of this section.

The Hammersley-Clifford theorem [1] allows us to write the joint probability distribution of the variables $\mathcal{A}_u$ as a product of functions over maximal cliques in the graph. For efficiency reasons, we ignore the cliques of size greater than two while modeling the intra-PSM relations. The pairwise consistency constraints between adjacent PSMs are modeled using normal distributions on the relative affine transformations. Concretely, let $\mathcal{R}_{u:v} \equiv {\mathcal{A}_u}^{-1}\mathcal{A}_v$ denote the affine transformation between the patches of v and u, or equivalently the vector of $\mathbb{R}^6$ representing the location, scale, skew and orientation of v in the coordinate frame of u, and let $\mu_{u:v} \in \mathbb{R}^6$ and $\sigma_{u:v} \in \mathbb{R}^{6\times 6}$ denote the corresponding mean vector and covariance matrix. We model the joint distribution of the PSM positions as

$$p_g(\mathcal{A}_1, \mathcal{A}_2 \ldots \mathcal{A}_N) = \frac{1}{Z} \prod_{(u,v)\in E} \mathcal{N}(\mathcal{R}_{u:v}; \mu_{u:v}, \sigma_{u:v}),$$

where $\mathcal{N}(\mathcal{R}; \mu, \sigma)$ is the normal distribution with mean μ and covariance matrix σ, and Z is a normalization constant. We assume a diagonal form for the covariance matrix $\sigma_{u:v}$.

It is important to note that this model is *not* equivalent to a joint Gaussian model on all the random variables $\mathcal{A}_u$. This would indeed be true if the model was constraining the PSM positions instead of relative affine transformations between the PSMs. In that case, the model would be equivalent to a Gaussian model on the random variables with a specific structure imposed on the inverse covariance matrix (the only non-zero entries would be the ones corresponding to the edges in the graph). However, since we impose Gaussian constraints on the *relative* transformations of PSMs, the model can no longer be written as a joint Gaussian density on the $\mathcal{A}_u$. However, as discussed in the introduction, constraining only the relative affine transformations (and not the relative x, y locations) is important to make the model robust to viewpoint and intra-class variations.

4.2. Learning the PSM Graph Parameters

Both $\mu_{u:v}$ and $\sigma_{u:v}$ are estimated from the training images in which u and v are seen together. We initialize $\mu_{u:v}$ using the mean of the observed relative affine transforms and $\sigma_{u:v}$ as a diagonal matrix containing the observed variance of each of the entries in $\mu_{u:v}$. However, this estimate is biased and the variances estimated are extremely small. Intuitively, the observed variance between two adjacent nodes

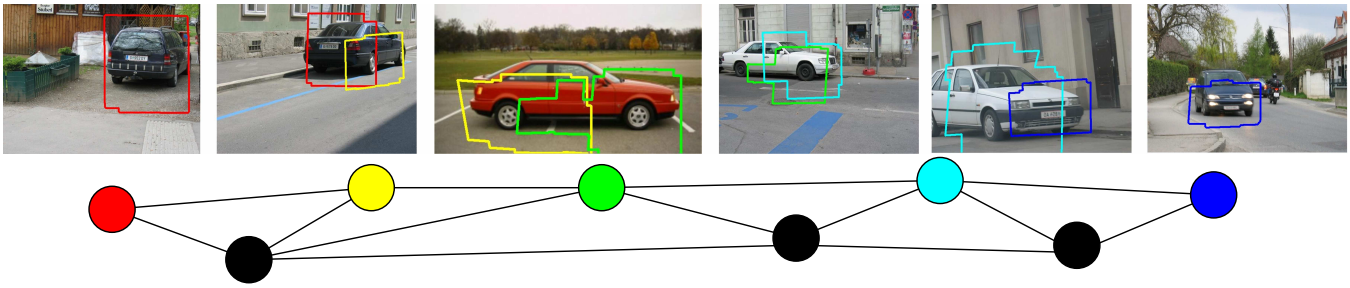


Figure 3. An example of learned PSM graph. The top row shows the outlines of the PSM instances corresponding to nodes with the same color in the PSM graph below it. The black nodes represent other nodes in the PSM graph. Please view in color.

occurs as a result of all the constraints in the graph connecting the 2 nodes acting together and attributing the variance to a single constraint makes it too tight. Hence, we use a gradient ascent procedure to optimize the variance parameters $\sigma_{u:v}$ so that the log likelihood of the PSM matches observed in the validation set is maximized. Let $\{i_1 \dots i_p\}$ be the indices of PSMs observed in some validation image I and let $\{j_1 \dots j_q\}$ be the indices of the PSMs that are not observed in I. The likelihood of the geometrical configuration observed in the ith validation image can be computed by integrating out the unobserved PSMs:

$$p_g(\mathcal{A}_{i_1}, \dots \mathcal{A}_{i_p}) = \int_{j_1 \dots j_q} p_g(\mathcal{A}_1, \mathcal{A}_2 \dots \mathcal{A}_N).$$

We use loopy belief propagation [15] to *approximately* compute the log-likelihood for the observed configurations in the validation images. The total log-likelihood of the validation set is just the sum of the log-likelihoods for the individual validation images since the images are assumed to be independent. Computing the "exact" gradient of the log-likelihood on the validation images is not computationally feasible since it requires running a belief propagation step for each validation image for every variance parameter on every edge after each iteration. Hence, we compute an efficient approximation to this gradient and use it to optimize the log-likelihood. The details of the gradient ascent procedure are described in Algorithm 1. This algorithm runs a single belief propagation on every validation image after every iteration. Each belief propagation is initialized using the state at the end of the previous iteration and hence converges extremely quickly. In practice, the algorithm converges in less than 10 iterations and the entire optimization process takes less than 10 minutes on a desktop machine for a model with about 30 PSMs and 50 edges. Also, it is usually sufficient to optimize just a single scaling ratio for the entire covariance $\sigma_{u:v}$ on each edge instead of optimizing the 6 parameters independently.

Algorithm 1 Optimization of Variance Parameters.

Input: A set of validation images with detected PSMs, initial estimate of the PSM graph and step length τ.
Output: Optimized PSM Graph.

for all validation images I **do**
- Initialize the observed PSMs in I;
- Run BP on I and store the state of messages;

end for
repeat
- Set gradient $G_{u:v}$ of $\sigma_{u:v}$ on all edges (u, v) to 0;
- **for all** validation images I **do**
  - **for all** observed nodes v in I **do**
    - Assume v is not observed and compute the belief at v using the incoming messages;
    - Compute gradient $\partial L / \partial M_{u:v}$ of the log-likelihood of the observed v w.r.t. the incoming messages $M_{u:v}$;
    - Compute gradient $\partial L / \partial \sigma_{u:v}$ by multiplying $\partial L / \partial M_{u:v}$ and the Jacobian $J(M_{u:v}, \sigma_{u:v})$;
    - Add gradient to $G_{u:v}$ for edges (u, v) incident on v;
  - **end for**
- **end for**
- Update $\sigma_{u:v} \leftarrow \sigma_{u:v} + \tau \cdot G_{u:v}$;
- Run BP and update the messages on all validation images using the updated $\sigma_{u:v}$;

until convergence.

4.3. Object and Background Models

Our object model is generative: First, a PSM graph instance, with an affine transformation $\mathcal{A}_u$ for every component PSM u, is chosen from the above distribution. Next, each PSM independently chooses its occlusion state, with probability $P_{\text{vis}}(u)$ to be visible, and probably $P_{\text{occ}}(u) = 1 - P_{\text{vis}}(u)$ to be hidden. If a PSM u is visible it then generates a PSM match m at the location $\mathcal{A}_u$. Finally, the appearance of m is chosen independently from the distribution $p(m|\text{obj})$ for each visible PSM. $P_{\text{vis}}, P_{\text{occ}}$ are learned by just measuring the statistics of the PSM on the validation data.

The background model generates matches for each model PSM u from a Poisson distribution $P_{poiss}(n|K_u)$ with mean K_u. The location for each PSM match is selected from a uniform pdf over the size of the image (for position) and a reasonable range of scale, orientation and skew parameters. These ranges are estimated from the background dataset. Finally, the appearance of the PSM match is chosen from the distribution $p(m|\text{bkg})$. The means K_u of the

Poisson distribution are learned using the statistics observed on the background data. Note that the appearance models $P(m|\text{obj})$ and $P(m|\text{bkg})$ are not learned explicitly. Instead, the logistic regression model associated with every PSM is used to predict their ratio $\frac{P(m|\text{obj})}{P(m|\text{bkg})}$ for the PSM. Finally, we assume that the background is present in every image and the number of objects present in it follows a Poisson distribution with mean $\lambda_{\mathcal{O}}$. Again, this mean $\lambda_{\mathcal{O}}$ may be learned from the training data.

5. Object Detection/Localization

5.1. Object Detection

We start by matching all the PSMs in the object model to the test image independently to obtain a set of PSM matches. Each PSM match m is assigned a probability ratio $\frac{P(a|\text{obj})}{P(a|\text{bkg})}$ based on its appearance by the logistic regression model for the corresponding PSM. Let $\mathcal{D}$ represent the set of all the PSM matches detected in the test image. Every match in $\mathcal{D}$ is generated either by the corresponding PSM from an object or by the background. We denote by $\mathcal{O}_i$ the subset of $\mathcal{D}$ corresponding to instance number i of the object in the image (there may be no such instances, or several ones). Even though $\mathcal{D}$ may contain multiple matches for a single PSM, each object instance contains at most one match per PSM. Let Φ denote the set of all the PSMs in the object model, and let $\Phi_{\mathcal{S}} \subset \Phi$ for any set of PSM matches $\mathcal{S}$ denote the set of PSMs that are matched at least once by matches in $\mathcal{S}$. Finally, let us denote the PSM corresponding to a PSM match m by u_m and the appearance of m by a_m. Recall that the appearance of a PSM match m is a binary vector containing one coordinate for every feature in a_m whose value is either 1 or -1 depending on whether the corresponding feature is matched in m or not. An explanation $E = \{\mathcal{O}_1, \mathcal{O}_2 \ldots \mathcal{O}_k, \mathcal{B}\}$ of $\mathcal{D}$ is a partition[2] of $\mathcal{D}$ where the matches in $\mathcal{O}_i$ correspond to the ith object instance and $\mathcal{B}$ is the set of background matches. Let $\mathcal{P}(\mathcal{D})$ be the set of all the possible explanations of $\mathcal{D}$.

The probability distribution $p(E, \mathcal{D})$ represents the probability that an explanation E generates the matches $\mathcal{D}$ in the image. In other words, if $E = \{\mathcal{O}_1, \ldots \mathcal{O}_k, \mathcal{B}\}$, then $p(E, \mathcal{D})$ represents the probability that 1) there are k object instances present in the scene, 2) the ith object instance generates the matches in $\mathcal{O}_i$, and 3) the background generates the matches in $\mathcal{B}$. Since all the objects in the image and the background generate matches independently of each other,

[2] E is not a partition in the strict sense, since it may contain empty blocks. An empty block for an object instance corresponds to the case when the object is present in the image but does not generate any matches (i.e. none of its PSMs are detected in the image). Such an explanation would have a low (but non-zero) probability. Similarly, the background block could also be empty indicating that the background does not generate matches for any of the PSMs.

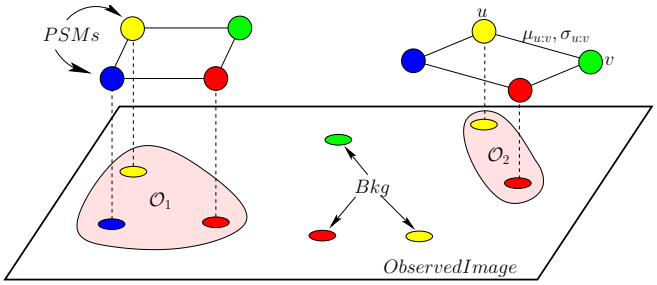


Figure 4. Explanation using two object instances.

we can write the distribution $p(E, \mathcal{D})$ as

$$p(E, \mathcal{D}) = p_b(\mathcal{B}) \;\; P_{poiss}(k|\lambda_{\mathcal{O}}) \prod_{1 \le i \le k} p_o(\mathcal{O}_i), \text{where :}$$

• The term $p_b(\mathcal{B})$ represents the probability that the PSM matches in $\mathcal{B}$ and only these matches are generated from the background. Since the background model assumes independence of the matches generated by the different PSMs, we can again decompose $p_b(\mathcal{B})$ as a product $\prod_{u \in \Phi} p_b(\mathcal{B}_u)$, where $\mathcal{B}_u$ is the set of n_u matches in $\mathcal{B}$ associated with u. Since the background model also assumes independence in the geometry and appearance for all the matches corresponding to any PSM we can write,

$$p_b(\mathcal{B}_u) = P_{poiss}(n_u|K_u) \prod_{m \in \mathcal{B}_u} p_g(m|\text{bkg}) P_{u_m}(a_m|\text{bkg}).$$

The term $p_g(m|\text{bkg})$ is the uniform pdf for the background model described earlier.

• The term $P_{poiss}(k|\lambda_{\mathcal{O}})$ represents the prior probability of the image containing k object instances.

• The term $p_o(\mathcal{O}_i)$ is the probability distribution that the matches in $\mathcal{O}_i$ and only these matches are generated by the ith instance of the object model. Since the appearance, the geometry and the occlusion state of the PSM matches are independent, we can write $p_o(\mathcal{O}_i)$ as a product of an appearance term $P_a(\mathcal{O}_i|\text{obj})$, a geometry term $p_g(\mathcal{O}_i|\text{obj})$ and the visibility term $P_{vis}(\Phi_{\mathcal{O}_i}) P_{occ}(\Phi \setminus \Phi_{\mathcal{O}_i})$. Since we assume that the appearances of the PSMs generated by the object are independent, we can write $P_a(\mathcal{O}_i|\text{obj}) = \prod_{m \in \mathcal{O}_i} P_{u_m}(a_m|\text{obj})$. The term $P_g(\mathcal{O}_i|\text{obj})$ represents the probability of the geometric configuration of the PSM matches in $\mathcal{O}_i$ and is computed using the PSM graph. Finally, since the occlusion variables for each PSM are independent, we can write $P_{vis}(\Phi_{\mathcal{O}_i}) = \prod_{u \in \Phi_{\mathcal{O}_i}} P_{vis}(u)$ and $P_{occ}(\Phi \setminus \Phi_{\mathcal{O}_i}) = \prod_{u \in \Phi \setminus \Phi_{\mathcal{O}_i}} P_{occ}(u)$.

We want to find the most likely explanation of the scene:

$$E^* = \underset{E \in \mathcal{P}(\mathcal{D})}{\text{argmax}} \; P(E|\mathcal{D}) = \underset{E \in \mathcal{P}(\mathcal{D})}{\text{argmax}} \; p(E, \mathcal{D}).$$

It is not feasible to search to over all possible explanations of $\mathcal{D}$, and we use a greedy algorithm to build up the "best"

explanation. The algorithm is initialized with the explanation $E_0 = \{\mathcal{D}\}$, assuming that all the matches in $\mathcal{D}$ are generated by the background. We then add a single object to the explanation and compute the most likely explanation $E_1 = \{\mathcal{O}_1, \mathcal{B}_1\}$, assuming that a single object generates the matches in $\mathcal{O}_1$ and the remaining matches in $\mathcal{B}_1 = \mathcal{D} \setminus \mathcal{O}_1$ are generated by the background. If $p(E_1, \mathcal{D}) > p(E_0, \mathcal{D})$, we fix the matches in $\mathcal{O}_1$ as belonging to the first object instance. Next, the most probable explanation $E_2 = \{\mathcal{O}_1, \mathcal{O}_2, \mathcal{B}_2\}$ with 2 objects is constructed by splitting the matches in $\mathcal{B}_1$ into $\mathcal{O}_2$ and $\mathcal{B}_2$. The algorithm iteratively adds the ith object instance to the explanation if $p(E_i, \mathcal{D}) > p(E_{i-1}, \mathcal{D})$, and terminates when adding more objects decreases the probability of the explanation. Figure 4 shows an explanation containing two object instances. During the ith iteration, the algorithm has fixed the matches in $\mathcal{O}_1, \ldots \mathcal{O}_{i-1}$ and needs to compute the most likely split of the matches in $\mathcal{B}_{i-1}$ into $\mathcal{O}_i$ and $\mathcal{B}_i$. Since it is not feasible to search over all the possible splits of $\mathcal{B}_i$, we use a greedy scheme to populate $\mathcal{O}_i$ starting from an empty set. We move matches from $\mathcal{B}_i$ to $\mathcal{O}_i$ one at a time so as to achieve the maximum increase in $p(E_i, \mathcal{D})$ at each step. We compute the ratio of $p(E_i, \mathcal{D})$ before and after moving a single match m from $\mathcal{B}_i$ to $\mathcal{O}_i$, and use this ratio to choose the best match to move. The process terminates when the best ratio drops below 1. While computing this ratio, the contributions of the terms corresponding to the first $i-1$ object instances cancel out and we obtain:

$$\frac{p(\{\mathcal{O}_1 \ldots \mathcal{O}_{i-1}, \mathcal{O}_i \cup \{m\}, \mathcal{B}_i \setminus \{m\}\})}{p(\{\mathcal{O}_1 \ldots \mathcal{O}_{i-1}, \mathcal{O}_i, \mathcal{B}_i\})} = \frac{p_{u_m}(a_m|\text{obj})}{p_{u_m}(a_m|\text{bkg})} \times \frac{p_{\text{vis}}(m|\text{obj})}{p_{\text{occ}}(m|\text{obj})} \times \frac{P_{poiss}(n_m - 1|K_{u_m})}{P_{poiss}(n_m|K_{u_m})} \times \frac{p_g(\mathcal{O}_i \cup m|\text{obj})}{p_g(\mathcal{O}_i|\text{obj}) \cdot p_g(m|\text{bkg})}$$

where n_m denotes the number of times m is matched in $\mathcal{B}_i$. The first term can be computed using the logistic regression model for the PSM. The second and third terms are computed using the learned occlusion parameters $p_{\text{vis}}, p_{\text{occ}}$ and background Poisson distribution parameter K_u. The final term is computed (approximately) by running loopy belief propagation on the graph. In fact, what we need to compute is the ratio $\frac{p_g(\mathcal{O}_i \cup m|\text{obj})}{p_g(\mathcal{O}_i|\text{obj})}$ which can be computed as follows. The nodes in $\Phi_{\mathcal{O}_i}$ are fixed and belief propagation is run to compute the "approximate" marginal distributions $p_g(\mathcal{A}_u|\Phi_{\mathcal{O}_i})$ on all the nodes $u \in \Phi \setminus \Phi_{\mathcal{O}_i}$. The required ratio is just $p_g(\mathcal{A}_{u_m}|\Phi_{\mathcal{O}_i})$. The term $p_g(\{m\}|\text{obj})$ for an object instance containing a single PSM match m is assumed to be the same as $p_g(m|\text{bkg})$.

5.2. Localization

Once we have computed an explanation E for the image, we use the PSM matches within each object instance $\mathcal{O}_i$ to predict a bounding box for it. For every PSM match m_u in $\mathcal{O}_i$, we transform the bounding box in the base image of u to the test image using $\mathcal{A}_u$. The predicted bounding box for $\mathcal{O}_i$ is just the mean of these bounding boxes for all the matches $m \in \mathcal{O}_i$. We compute a score $\beta_{\mathcal{O}_i}$ (needed to plot Precision-Recall curves) for each $\mathcal{O}_i$ using the appearance and visibility terms for the PSMs within $\mathcal{O}_i$ as

$$\beta_{\mathcal{O}_i} = \log\left[P_a(\mathcal{O}_i) P_{vis}(\Phi_{\mathcal{O}_i}) P_{occ}(\Phi \setminus \Phi_{\mathcal{O}_i})\right].$$

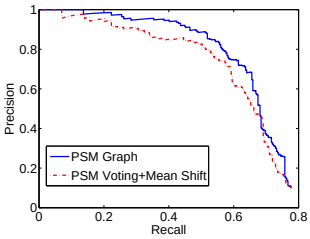
6. Experiments and Discussion

We have conducted experiments on the PASCAL VOC Challenge 2005 Cars Test 1 dataset [14], which consists of 275 images containing one or more cars (in a variety of poses) and 414 background images with no cars present. The training data consisted of 272 positive images with 320 annotated cars, and 412 background images.

• **Pre-processing of the training data:** Since images of cars facing left are identical (up to a flip about the vertical) to cars facing right, we learn a model for detecting cars facing right. The detector is run on the original image as well as its flipped version to be able to detect cars facing both to the left and to the right. To simplify the learning of the object model we correct each image of the training set so that the car is pointing towards the right (by flipping the image, if required). Then we annotate the images with a rough viewing direction from the set $\{Rear, RearSide, Side, SideFront, Front\}$. Images from a given class are only matched to others from the same class or from neighboring classes during the PSM validation phase. Since our method first learns the object part appearances "independently" and later learns the geometry, it requires a reasonably "clean" set of validation matches while building the model. Thus, we prune the validation image matches based on whether the location of the matching patches within the bounding box in the base image is roughly consistent with the location of the corresponding matches in the bounding box of the validation image.

• **Results:** Qualitative detection and localization results on the Test 1 data are shown in Fig. 5. As per the rules of the VOC Challenge [14], a detection is considered correct if the intersection area of the predicted bounding box and the annotated bounding box is at least 50% of the union of the two. Also, multiple detections for the same object instance are considered false positives. We have implemented a simple baseline method to judge the performance of the PSMs without the geometric model: Each PSM match m_u in the test image casts a vote for its predicted bounding box using $\mathcal{A}_u$ (similar to Section 5.2) and a mean shift procedure is used to find the modes in this space. The VOC Challenge competition used the average precision (AP) score [14] to rank the results submitted by the participants. Our baseline approach achieves an AP score of 0.590, which is just below the best score of of 0.613 obtained by Dalal and Triggs [2]

Figure 5. Successful detections on the PASCAL VOC 2005 Cars Test 1. The last two images in the bottom row show correct detections (small red rectangles) showing up as false positives because they were not annotated in the dataset. Please view in color.



Algorithm	AP
PSM graph	**0.628**
Dalal & Triggs [2]	0.613
Voting+MS	**0.590**
Fritz et al. [11]	0.489
Garcia & Duffner [6]	0.353

Figure 6. Precision/recall curves (left), and AP score comparison (right). The results for Dalal and Triggs, Fritz et al., and Garcia and Duffner are taken from [14]. The references in brackets are the publications where the methods used in the challenge have been first described.

in the competition. The PSM graph approach achieves an even higher score of 0.628—the highest obtained so far for this dataset. Figure 6 shows the precision/recall curves corresponding to the baseline method and the full PSM graph algorithm, and compares our AP scores with those obtained by the participants in the VOC 2005 Challenge [14].

• **Discussion:** These experimental results demonstrate the strength of our model: First, we can see that PSMs are very discriminative, since when combined with a simple voting scheme, they significantly outperform comparable methods based on individual local features [11]. This style of voting can be thought of as a (simple) implementation of a rigid star model. Our experiments also demonstrate the power added by our loose geometric model, which significantly outperforms all other methods on the Cars Test 1 data of the Pascal 2005 VOC Challenge.

Acknowledgments: This work was supported in part by the National Science Foundation under grant IIS-0535152, the UIUC/INRIA/CNRS collaboration agreement, and the INRIA associated team Thetys.

References

[1] J. Besag. Spatial interaction and the statistical analysis of lattice systems. *J. of the Royal Stat. Soc.*, 1974.

[2] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[3] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR*, 2003.

[4] V. Ferrari, T. Tuytelaars, and L. V. Gool. Integrating multiple model views for object recognition. In *CVPR*, 2004.

[5] V. Ferrari, T. Tuytelaars, and L. Van Gool. Simultaneous object recognition and segmentation by image exploration. In *ECCV*, 2004.

[6] C. Garcia and M. Delakis. Convolutional face finder: A neural architecture for fast and robust face detection. *PAMI*, 2004.

[7] F. Jurie and C. Schmid. Scale-invariant shape features for recognition of object categories. In *CVPR*, 2004.

[8] J. Koenderink and A. Van Doorn. Affine structure from motion. *JOSA A*, 1990.

[9] A. Kushal and J. Ponce. Modeling 3D objects from stereo views and recognizing them in photographs. In *ECCV*, 2006.

[10] S. Lazebnik, C. Schmid, and J. Ponce. A maximum entropy framework for part-based texture and object recognition. In *ICCV*, 2005.

[11] B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *ECCV WSLCV*, 2004.

[12] N. Loeff, H. Arora, A. Sorokin, and D. Forsyth. Efficient unsupervised learning for localization and detection in object categories. In *NIPS*, 2005.

[13] D. Lowe. Object recognition from local scale-invariant features. In *ICCV*, 1999.

[14] M. Everingham et al. The 2005 PASCAL Visual Object Classes Challenge. In *The First PASCAL Challenges Workshop*, 2006.

[15] K. Murphy, Y. Weiss, and M. Jordan. Loopy belief propagation for approximate inference: An empirical study. In *UAI*, 1999.

[16] F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce. 3D object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints. *IJCV*, 2006.

[17] T. J. Santner and D. E. Duffy. *The Statistical Analysis of Discrete Data*. Springer Verlag, 1989.

[18] A. Thomas, V. Ferrari, B. Leibe, T. Tuytelaars, B. Schiele, and L. V. Gool. Towards multi-view object class detection. In *CVPR*, 2006.

[19] J. Zhang, M. Marsza, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. *IJCV*, 2007.