A Multi-View Probabilistic Model for 3D Object Classes

Min Sun\textsuperscript{1}\textsuperscript{*} Hao Su\textsuperscript{1,3}\textsuperscript{*} Silvio Savarese\textsuperscript{2} Li Fei-Fei\textsuperscript{1}

\textsuperscript{1}Dept. of Computer Science, Princeton University, USA, \{sun, haosu, feifeili\}@cs.princeton.edu
\textsuperscript{2}Dept. of Electrical and Computer Engineering, University of Michigan at Ann Arbor, USA, silvio@eecs.umich.edu
\textsuperscript{3}Dept. of Computer Science, Beihang University, China

Abstract

We propose a novel probabilistic framework for learning visual models of 3D object categories by combining appearance information and geometric constraints. Objects are represented as a coherent ensemble of parts that are consistent under 3D viewpoint transformations. Each part is a collection of salient image features. A generative framework is used for learning a model that captures the relative position of parts within each of the discretized viewpoints. Contrary to most of the existing mixture of viewpoints models, our model establishes explicit correspondences of parts across different viewpoints of the object class. Given a new image, detection and classification are achieved by determining the position and viewpoint of the model that maximize recognition scores of the candidate objects. Our approach is among the first to propose a generative probabilistic framework for 3D object categorization. We test our algorithm on the detection task and the viewpoint classification task by using “car” category from both the Savarese et al. 2007 and PASCAL VOC 2006 datasets. We show promising results in both the detection and viewpoint classification tasks on these two challenging datasets.

1. Introduction

Imagine a busy street in downtown Rome or Beijing. There might be cars coming from all possible directions, pedestrians walking on the sidewalk, or scooters crossing the street. The ability to interpret the scene, recognize the objects within, estimate their locations and poses is crucial if one wants to avoid obstacles, interact with other people, and find a target location. However, what appears to us as natural may become tremendously difficult for an artificial vision system. How do we handle occlusions? How do we deal with intraclass and pose variability of objects? Most of the current researches in object categorization \cite{7, 5, 9, 16, 1, 31} have focused on modeling object intraclass variability in single views (within a small range of planar rotations) or as a mixture of single view models \cite{27, 33}. Very few methods have leveraged on the intrinsic geometric aspects of object categories.

In this paper, we introduce a generative probabilistic framework for learning visual models of 3D object categories. Our approach is inspired by a number of recent work for representing 3D object categories where appearance information and geometric constraints are combined (Sec. 2). We represent an object as a coherent ensemble of parts linked across different viewpoints (Sec. 3). Each part is represented by a distribution of characteristic appearance elements (codewords). A generative model is used for learning the relative position of parts within each viewpoint, as well as corresponding part locations across viewpoints (Sec. 4). Epipolar constraints are applied to ensure that the configuration of each part is consistent under viewpoint transformations. This is one of the first generative probabilistic 3D object models that incorporates such geometric constraints in a principled way. Unlike other methods that focus on discriminative recognition of characteristic patches, our model is able to generate a coherent distribution of parts that are representative of the object class and are consistent under viewpoint transformation (see Fig. 1). We have used this model to predict whether an object class is present in the image or not, estimate the location of the object as well as determine its viewpoint (Sec. 5). Our experiments show superior detection and viewpoint classification results on the 3D Objects dataset \cite{25}, and comparable results on PASCAL VOC 2006 datasets \cite{6}.

2. Previous Work

Researchers have done extensive work in single object recognition from multiple poses. Notably, \cite{3, 32, 8, 24} have proposed methods for identifying and recognizing single objects such as a cereal box or a Teddy bear, under arbitrary viewing conditions. These methods are successful due to their reliance on the identification of strong geometrical constraints and highly discriminative object features. However, such constraints are not adequate in object categorization problems in which shape and appearance variability of each object class must be accounted for. Similar limitations...
exist for representations based on aspect graphs [12].

Recently, a number of methods have brought new ideas into the problem of representing object categories from multiple views. Thomas et al. [30] have explored the idea of linking features across multiple views. Kushal et al. [14] propose to connect groups of features across views which retain pairwise local geometrical transformations in an MRF framework. Other methods [11, 34] propose to represent the object category by using a rough 3D model, on top of which the typical distribution of appearance element is learned. These methods have the advantage of yielding a compact view-point invariant representation of the object category, as opposed to [30, 14] but fail to accommodate intra-class 3D shape variability. Except for [4], none of the above models have the capability of generating spatial and appearance information of the object category under arbitrary viewing conditions. The framework presented in [25, 26] represents an object category as a collection of view-invariant regions linked by transformations that capture the relative change of pose among parts. The model has the ability to generate unseen views, but achieves limited accuracy in classification due to the lack of an explicit background model.

Our method tries to overcome some of the limitations of previous works. Unlike [30], but similar to [14, 25, 26] our model parts are linked across views under affine transformations. As opposed to [11, 34], we generalize on both appearance and geometry. Unlike [30, 25, 26], our method offers a principled way for learning a rigorous generative model for 3D objects.

3. Model Representation

Our proposed model learns a coherent part-based object representation across instances and viewpoints of an object class. In addition to being able to discover object parts that appear consistently across object instances, it can simultaneously establish part-level correspondences by exploiting the underlying geometric structure across nearby viewpoints (see Fig 1).

3.1. The Generative Model

A generative graphical model (Fig.2 top left panel) illustrates the backbone of the model structure. We first introduce the main representation of this model. We then describe in details how epipolar geometry is introduced to regularize the coherent representation of object parts across different viewpoints. The bottom panel of Fig.2 summarizes the important variables in the model.

Imagine a generative process for sampling an object image given a learned 3D model representation. We start with sampling a particular viewpoint $v$ out of $K$ possible 3D views of this object. Given $v$, we then sample an object part type assignment $l$ for each image patch. For a robust 3D representation of an object class, it is important to point out that different viewpoints would render very different part distributions (e.g. wheel parts are more likely to be observed in a sideview of a car than a frontal view of a car). Given a part type assignment $l$ for this patch, we sample the image patch based on the appearance $(y)$ distribution as well as position $(x)$ distribution.

We now introduce the generative process in a more mathematically rigorous way. In order to simplify the model description, we first assume that the different viewpoints of an object class are affine aligned. In other words, viewpoints, scales and orientations are first assumed to be matched. We will relax this assumption once the generative model is described. The overall model structure is a Dirichlet Process (DP) Gaussian mixture. The choice of using DP is particularly important to accommodate a variable number of object part types per viewpoint. We start with an image $i$.

1. **Generate viewpoint $v_i$**
   - Draw viewpoint $v_i \sim \text{Uniform}(\phi)$. For a total of $K$ viewpoints, $\phi = 1/K$. 

---

1[14] is the only other reported 3D object class model that is probabilistic in nature. The generative representation of our model differs from their discriminative framework.
2. For each of the $n = 1, ..., N_i$ feature patches of the image, generate part type assignment $l_n$.

   - Draw part type assignment $l_n \sim \text{Mult}(s_k)$, where $s_k = \{s_{kt} | t = 1, ..., \infty\}$, and $s_{kt}$ is computed from $s_{kt} = u_{kt}\prod_{m=1}^{t-1}(1-u_{km})$, the expected part proportion of part type $t$ in viewpoint $k$. $u_{kt}$ is a global parameter indicating the number of parts with type $t$ in viewpoint $k$, where $u_{kt} \sim \text{Beta}(\phi)$. Note that the Dirichlet Process model allows us to consider an infinite number of possible part types per viewpoint.

3. For each of the $n = 1, ..., N_i$ feature patches of the image, generate the patch appearance $(y_{ni})$ and patch location $(x_{ni})$.

   - Given viewpoint $v_i$ and part type assignment $l_n$, sample the patch location $x_{ni} \sim N(\theta_{kt})$, where $N(\cdot)$ indicates a Gaussian distribution. $\theta_{kt}$ is the global parameter of the Gaussian distribution for part type $t$ and viewpoint $k$. $\theta_{kt}$ is governed by a Gaussian-Wishart distribution $\text{NW}(\lambda_t)$, the conjugate prior of a Gaussian distribution, where $\lambda_t$ consists of hyperparameters $\{\beta_0, m_0, W_0, v_0\}$.

   - Given viewpoint $v_i$ and part type assignment $l_n$, sample the patch appearance $y_{ni} \sim \text{Mult}(\eta_{kt})$, where $\text{Mult}(\cdot)$ indicates a Multinomial distribution. $\eta_{kt}$ is the global parameter of the Multinomial distribution for part type $t$ and viewpoint $k$. $\eta_{kt}$ is governed by a Dirichlet distribution $\text{Dir}(\alpha_t)$, the conjugate prior of a Multinomial distribution, where $\alpha_t$ is the hyperparameter.

Putting all the observable variables (feature patches $X$ and $Y$, viewpoints $V$) and latent variables (part type assignments $L$) together with their corresponding parameters, we can write down the joint probability of this model.

$$P(X, Y, L, V, u, \eta, \theta) = \prod_i^I \left( \prod_n^{N_i} P(v_i|\phi) P(l_n|s_{v_i}) \right) \cdot \prod_{i=1}^{K} \left( \prod_{t=1}^{\infty} P(y|\nu, \eta, \phi_t) P(x|\theta, \lambda_t) \right)$$

(1)

where $I$ is the total number of images, and $N_i$ is the number of feature patches in image $i$.

**Affine transformation** So far we have introduced the model by assuming complete affine alignment of viewpoints, scales and orientations. An important contribution of our model is the ability to automatically find an affine transformation across different viewpoints of object images such that object recognition can be accomplished under arbitrary view conditions. This is achieved by introducing the affine transformation variable $A$ for each image, as shown in the top left panel of Fig.2. The transformation $A$ operates on the patch location $x$ to obtain the optimal patch alignment for the model corresponding to viewpoint $v$. The modified patch location probability becomes:

$$P(x|\theta, A) = N(x; \hat{A} \mu + b, \hat{A} \Lambda \hat{A}^T)$$

(2)

where $A = [\hat{A} \ b]$, $\theta = \{\mu, \Lambda\}$, $\mu$ and $\Lambda$ are the means and the covariance matrices of the Gaussian distribution of the reference object coordinate respectively. Sudderth et al [29] have proposed a DP mixture model for 2D object classes by allowing a translational transformation for different patches of objects. But their model does not capture the 3D structure
of an object class, nor does it encode part correspondences across viewpoints.

3.2. Constraining Generative Model Across Viewpoints

Up to this point, for each viewpoint, object class are only governed by the hyper-parameters $\alpha_t$, $\lambda_t$, and $\phi$, where $\alpha_t$ and $\lambda_t$ control the appearance and position variability of part type $t$, and $\phi$ regulates the part type proportions in different viewpoints. When learning and fitting 3D object class models, the number of possible transformations allowed for the object part locations and appearances can be orders of magnitude larger than those observed in 2D image based object models. This poses a potentially detrimental problem for a model regulated only by these hyper-parameters. In order to reduce the complexity of the parameter space, we propose to take advantage of the geometrical relationships regulating the appearance of 3D objects observed from multiple vantage points. The challenge here is to strike a good balance between enforcing such relationships and maintaining the ability to generalize well across instances within the same object class. We propose to enforce the following constraints:

- **Part appearance similarity constraints.** It is reasonable to assume that parts in nearby viewpoints share similar appearance. For example, the front wheels in the first row of images in Fig. 1 are likely to have similar distributions of visual words. When building the model, we enforce corresponding parts to look similar by penalizing the sum of the square errors of the difference of the visual word distributions in nearby viewpoints. A similar approach is proposed by [20]. Fig. 3 illustrates the appearance similarities of corresponding parts in different views.

- **Part configuration constraints via epipolar geometry.** The 2D configuration of object parts in the image plane changes as the viewpoint varies. We can use epipolar geometry [10] to constrain the 2D part configuration in images corresponding to nearby viewpoints. More specifically, given images of a particular object instance taken from nearby viewpoints, we obtain epipolar constraints via feature matching. We then apply these constraints on the locations of the corresponding parts across nearby viewpoints during model learning (Fig 4). Each constraint is equivalent to a linear constraint in part locations (see Sec. 4.1 for details).

Such constraints enable our model to capture the underlying structure of a 3D object category across viewpoints. While simple in nature, these constraints play an important role in model learning, allowing much more accurate estimations of number of parts, their locations and alignment. In the next section, we describe how to incorporate these implicit constraints during model learning.

4. Model Learning

Given the model representation in Fig. 2, the goal of learning is to infer the latent variables and estimate the hidden parameters by maximizing the log marginal probability

\[ \ln P(X, Y) = \ln \sum_{V, L} \int P(X, Y, L, V, u, \eta, \theta) du dv d\eta d\theta. \]  

(3)

In our current setting, we make the following assumptions during the learning process: i) object bounding boxes are provided; and ii) viewpoint labels are given to all images.

2In our current learning process, only one object instance (and its multiple examples across different viewpoints) is used for establishing this constraint. About 90% of the image pairs are automatically matched correspondences. But about a dozen missed pairs are hand pruned.

3Due to space limitation, more detailed derivations can be found in an accompanying technical report on the authors’ website.
4.1. Updates of the Model Parameters

Computing the exact marginal distribution is intractable. We employ variational EM to learn the model in the stick-breaking representation [28]. Using the variational distribution, we maximize $L(X, Y)$, the lower bound of $\ln P(X, Y)$, and fit the variational parameters by coordinate ascent. The mean-field variational distribution equations are

\[
q(V, L, u, \theta, \eta) = q(V, L)q(u, \theta, \eta)
\]

\[
q(V, L) = \prod_{i=1}^{I} q(v_i|\delta_i) q(l_t|\rho_{invl})
\]

\[
q(u, \theta, \eta) = \prod_{k=1, t=1}^{K, T} q(u_k|\gamma_k) q(\eta_k|\eta_k) q(\theta_k|\epsilon_k)
\]

where $\delta_i$ is the variational multinomial parameter over the $K$ viewpoints, $\rho_{invl}$ is the variational multinomial parameter over the feature part type assignment, $\gamma_k$, $\tau_k$, and $\epsilon_k$ are the variational beta, the variational Dirichlet, the variational Gaussian-Wishart parameters of part type $t$ in viewpoint $k$, and finally $T$ is the truncation number of part types in our model [2].

In the following, we summarize the variational update equations for each of the model parameters. Particularly, we show how the 3D structure constraints introduced in Sec. 3.2 are applied to guide the variational distribution towards convergence to the true posterior.

Spatial Parameters Updates Feature patch location $x$ is governed by the Gaussian distribution of parameter $\theta_{kt}$, whose variational parameters are consisted of $\beta_{kt}$, $m_{kt}$, $W_{kt}$ and $\nu_{kt}$, $m_{kt}$ encodes the expected mean of part centers, and the other parameters model the degree of intra-class variation of the part center locations. The expected mean of part centers $m_{kt}$ and $m_{kt'}$ across nearby viewpoints are constrained by the epipolar geometry. In our model, we capture the epipolar geometry between image in viewpoint $k$ and image in viewpoint $k'$ by using the affine fundamental matrix \(F_{k'k} = [m_{kt}; 1]^T F_{kkt} [m_{kt}; 1] \approx 0\), where $F_{k'k}$ is estimated from the epipolar geometry between viewpoint $k$ and $k'$ using the reference instance $x$.

The variational parameters are then updated as follow:

\[
\beta_{kt} = \beta_0 + N_{kt}, \quad \nu_{kt} = \nu_0 + N_{kt}
\]

\[
W_{kt}^{-1} = W_0^{-1} + N_{kt} \times S_{kt} + N_{kt} (\overline{x}_{kt} - m_0) + \beta_0 (m_{kt} - m_0) (m_{kt} - m_0)^T
\]

where the sufficient statistics of the spatial terms are:

\[
N_{kt} = \sum_{t,n} \delta_{ik} \rho_{inkt}, \quad \overline{x}_{kt} = \frac{1}{N_{kt}} \sum_{t,n} \delta_{ik} \rho_{inkt} x_{tn}
\]

Appearance Parameters Updates As described in Sec. 3.2, our model enforces a feature patch appearance similarity in nearby viewpoints $k$ and $k'$. We define a regularized marginal likelihood as:

\[
O(X, Y, G) = (1 - \zeta) L(X, Y) - \zeta R(G)
\]

\[
R(X, Y, G) = \frac{1}{2} \sum_{k} \sum_{(k',l') \in E} W_{k'}^w (\tau_{kl} - \tau_{kl'})^2
\]

where $W$ is the total number of visual codewords, $G$ is the graph structure over the viewing sphere, $E$ is the set of the edges defined in $G$, and $\zeta$ is the parameter that determines the significance of the regularization, whose value can range from 0 to 1. The variational parameter updates then become

\[
N_{ktw} = \sum_{t} \sum_{n \in \{y_{tn} = w\}} \delta_{ik} \rho_{inkt}
\]

\[
\tilde{\tau}_{ktw} = \alpha_t^{w} + N_{ktw}, \quad \tau_{ktw} = \frac{1}{K} \sum_{k} (\tilde{\tau}_{ktw})
\]

where $\tilde{\tau}_{ktw}$ is the updated $\tau_{ktw}$ when $\zeta$ is set to 0, $\tilde{\tau}$ is $\tau_{ktw}$ when $\zeta$ is set to 1, and $N_{ktw}$ is the sufficient statistics of the multimodal distribution. We further define $\tau_{ktw} = (1 - \pi) \tilde{\tau} + \pi \hat{\tau}$. Similar to [20], we update $\tau$ in a greedy approach. Starting from $\pi = \varphi = 0$, we use gradient descent algorithm to search for $\pi$ such that $O(X, Y, G)$ decreases.

Part Type Proportion Parameters Update The update equations for the variational parameters of the part type are straightforward.

\[
\gamma_{kt,1} = 1 + N_{kt}, \quad \gamma_{kt,2} = \varphi + \sum_{t+1}^{T} N_{kt}
\]

Latent Variables Updates Since viewpoint $v$ is provided in training, $\delta$ is fixed during learning, $\rho$ has a close form update rule. Please refer to the technical report for more details.

\[\text{All hyper-parameters } \{\phi, \alpha, \lambda, \varphi\}, \text{ except } A, \text{ are fixed.}
\]

\[\text{We use a sinlge object instance with all viewpoints captured as the reference instance.}\]
4.2. Implementation Details

We have described a principled framework for learning each parameter of the generative model through variational inference. In principle, all training images can be applied simultaneously to jointly update these parameters. This requires a joint estimation of affine transformations in every EM iteration. The update is equivalent to solving a semidefinite programing problem whose complexity is quadratic to the number of images. We therefore adopt an incremental learning framework to curtail the amount of computation.

Neal and Hinton [22] provide a theoretical ground for incrementally learning of mixture models via sufficient statistics updates. At every iteration of the incremental learning, we fix the sufficient statistics of the parameters $N_{kt}, T_{kt}, S_{kt}$, and $N_{ktw}$ associated with the previously learned images in all later updates.

Initialization We initialize our model using one single object instance (reference instance) across all viewpoints. This can be done by applying an existing feature matching algorithm in all pairs of the viewpoint images [35]. We then apply variational EM learning to all the reference instance training images.

Accurately learning part correspondences across viewpoints is critical for the rest of the learning. To demonstrate the importance of the additional constraints introduced in Sec. 3.2, we compare the initialized model with two simplified models – a basic DP Gaussian mixture model, and a DP Gaussian mixture model with epipolar constraints but no appearance constraints. Fig. 5 demonstrates that the full model shows the best part correspondence compared to the two simpler versions. But epipolar constraint alone can already boost the correspondences significantly. Fig. 6 shows examples of the resulting average images of the aligned parts using the full model.

Incremental Update Given a new training image, we estimate its affine transformation with respect to the reference instance in the corresponding viewpoint. The estimation is done by solving a semi-definite programming problem. Meanwhile, the rest of the model will be updated by adding the sufficient statistics of the new image. In this scheme, the model estimates the affine transformation for one image at a time.

Part Expansion Learned parts of an object class do not overlap with each other. But given the learned geometric configurations among parts, we can now further generate additional object parts by linearly interpolating and extrapolating between nearby parts. Empirical experiments confirm such expansion is helpful in object recognition tasks when objects are embedded in cluttered background.

5. Object Classification, Detection and Viewpoint Recognition

Given an unknown image, we can use the learned model to determine whether an instance of the object class is present, and to estimate the corresponding viewpoint. Inspired by the implicit shape model for object recognition [16], we accomplish the task in the following steps:

- **Extract features and propose candidate parts** We use Hessian-Affine feature detector [21], Maximally stable extremal regions (MSER) [19] detector, and canny edge detector [23] to detect locally adapted elliptical regions. A feature codebook of size 1000 is obtained by vector quantizing the SIFT descriptors computed over these detected regions [18].

Given a test image, we first detect a number of candidate parts for each of the learned parts in a particular viewpoint and scale. This is done by running a scanning window search method. The appearance similarities of the candidate parts are evaluated against the learned part model by using spatial pyramid matching [15] as the similarity score.

- **Object localization and viewpoint classification by mean-shift voting** Our object detection algorithm is similar to [16]. Given the spatial extent and the appearance similarity scores of the candidate parts, we look up the relative position of the object center to the object part in the learned spatial models. Each of the candidate part proposes an object center in the image. A mean-shift procedure is then applied to locate local maxima of the possible object centers based on each viewpoint model of the object class. Using this approach, we could handle multiple instance detection.
We first test our model on the car dataset of the multi-view object dataset proposed by [25]. We assess the performances of our algorithm to localize cars and classify the viewpoints of the car. The car dataset comprises 320 images from up to 10 object instances. Each instance comprises 8 angles, 2 heights and 2 distances, a total of 32 viewpoints. We train our 3D object class model by using 160 images from 5 object instances with known viewpoint labels. The remaining 5 object instances are used as testing images. Binary classification result for car category is reported in Fig 7. Our model shows a superior performance over [25]. We also show consistently higher classification results across 8 viewpoints\(^7\) compared to [25] (results in Fig 7, and examples in Fig 9 Top row).

We also conduct a car detection and viewpoint classification experiment by using the more challenging PASCAL VOC 2006 car dataset. We use the training data provided by PASCAL and the multi-view object datasets. The object class model is trained by assuming 8 different viewpoints for cars. Fig. 8(a) shows the car detection results of our model compared with the other state-of-the-art algorithms. Our algorithm performs on par with most of these discriminative methods. In addition, we show for the first time a quantitative viewpoint classification results on the PASCAL VOC 2006 car dataset. We test the performance of the model in the PASCAL test images by collapsing the 8 viewpoints into 4 views\(^8\). Fig 8(b) shows the confusion table result of the 4-way viewpoint classification task. Note that there is a relatively large confusion between the front and back view points. Besides the large visual similarity between these two views, we attribute the low discriminative ability of our model to the unbalanced training set (PASCAL VOC 2006 cars), which contains a much larger number of back-view car instances. Parts that are reliably detected provide strong evidence for object detection and viewpoint classification. For example, our model can robustly detect car wheels, which significantly contributes to the detection of left and right views. The bottom 2 rows of Fig 9 presents some sample recognition results.

6. Conclusions

We have proposed a rigorous generative model for learning the 3D structure of object classes. The model captures the geometric configurations of different parts of an object class linked across different viewpoints. Epipolar constraints are employed to ensure part consistencies when training object parts. We test the model in a car detection and car viewpoint classification experiment. We are esp-

\(^7\)This is done on correctly detected cars.

\(^8\)PASCAL data provides viewpoint labeling of only 4 views: front, right, left, back.
cially encouraged to see the model’s ability to perform satisfying viewpoint classification once the objects are detected. In the future, we plan to explore further the robustness and discriminability of the model.

Acknowledgments

The authors would like to thank Jia Deng, Barry Chai, and Bangpeng Yao for their help for data collection. The authors especially thank Juan Carlos Niebles for helpful comments and discussions. Hao Su is funded by China Scholarship Council. Li Fei-Fei is funded by a Microsoft Research fellowship and a Google award.

References


