Learning a dense multi-view representation for detection, viewpoint classification and synthesis of object categories

Hao Su[1,4*] Min Sun[2*] Li Fei-Fei[3] Silvio Savarese[2]

[1]Dept. of Computer Science, Princeton University, USA, {haosu}@cs.princeton.edu

[2]Dept. of Electrical and Computer Engineering, University of Michigan at Ann Arbor, USA, {sunmin,silvio}@eecs.umich.edu

[3]Dept. of Computer Science, Stanford University, USA, {feifeili}@cs.stanford.edu

[4]Dept. of Computer Science, Beihang University, China.

Abstract

Recognizing object classes and their 3D viewpoints is an important problem in computer vision. Based on a part-based probabilistic representation [31], we propose a new 3D object class model that is capable of recognizing unseen views by pose estimation and synthesis. We achieve this by using a dense, multiview representation of the viewing sphere parameterized by a triangular mesh of viewpoints. Each triangle of viewpoints can be morphed to synthesize new viewpoints. By incorporating 3D geometrical constraints, our model establishes explicit correspondences among object parts across viewpoints. We propose an incremental learning algorithm to train the generative model. A cellphone video clip of an object is first used to initialize model learning. Then the model is updated by a set of unsorted training images without viewpoint labels. We demonstrate the robustness of our model on object detection, viewpoint classification and synthesis tasks. Our model performs superiorly to and on par with state-of-the-art algorithms on the Savarese et al. 2007 and PASCAL datasets in object detection. It outperforms all previous work in viewpoint classification and offers promising results in viewpoint synthesis.

1. Introduction

Visual recognition is a cornerstone task for an artificial intelligence system. In computer vision, object recognition, particularly object categorization, has been one of the most widely researched areas in recent years. Tremendous progress has been made especially in image-level object classification under limited geometric transformations, such as classification of side-view cars, or frontal view faces (e.g.[35, 12, 10]). Also relevant is the line of work in object detection in cluttered real-world scenes, such as pedestrian detection, or car detection [38, 11, 25, 37, 4, 7].

But most of the previous approaches can only handle up to a small degree of viewpoint variations of the 3D objects. As a result, they can hardly be used for robust pose understanding, a crucial functionality for real-world applications where accurate recognition of objects under arbitrary view points and 3D poses are needed. A small but growing number of recent studies have begun to address the problem of object classification in a true multi-view setting [32, 16, 14, 37, 5, 27, 28, 21]. While this is an important step forward, the focus is still on object detection without extensive quantitative analysis of 3D viewpoint estimation (the exceptions being [27, 28, 21]).

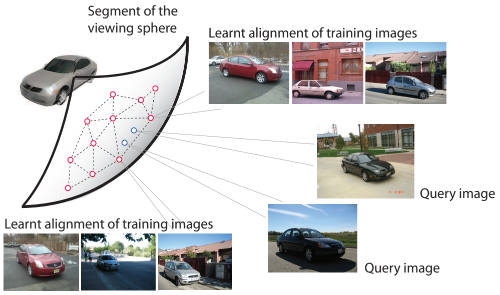

Segment of the viewing sphere
Learnt alignment of training images
Query image
Learnt alignment of training images
Query image

Figure 1: We propose a dense multi-view representation of 3D object categories. Consider a car category as an example. The red circles on the viewsphere indicate the viewpoints of the object class learned by our model. Some sample training images are shown for two of the viewpoints, demonstrating our model's ability to automatically align unlabeled poses at training time. Given a query image (dark blue circle) our model is capable of simultaneously categorize the object in the image and estimate its correct viewpoint by synthesizing a novel pose at recognition time. Notice that the viewpoint of the object in the query image does not have to be observed during training (the dark blue circle does not overlap with the red circle).

In this paper, we propose a new framework for learning a probabilistic 3D object model that can be used to categorize and detect an object in a cluttered scene, estimate its viewpoints accurately, or synthesize a new viewpoint given a single test image. We focus on overcoming two major challenges in representing and modeling 3D object classes.

- We develop a dense multi-view representation of object classes through an incremental learning algorithm. The probabilistic model construction process is initialized from a cellphone video sequence of a single object instance. Our algorithm then builds the object class model from unsorted images without any viewpoint

*indicates equal contributions

supervision by automatically aligning arbitrary poses at training time (Fig. 1).

- Our 3D object recognition algorithm is able to recognize objects under arbitrary viewpoints, even if the object instances were not observed during training. If we define the viewing sphere as a collection of viewpoints from which an object can be observed, our algorithm accurately estimates the pose of the object on it (Fig. 1). To our knowledge, this is the first probabilistic model that is capable of representing and recognizing unseen object views.

2. Related work

3D object recognition as well as pose estimation started with a number of seminal papers in single object recognition based on representing the object by highly discriminative features [3, 22, 34, 13, 26], or by exploring their topological structures [15, 2, 6]. Both types of models rely on encoding rigid spatial and geometrical constraints that cannot be used to accommodate large degree of intra-class variability in object categories.

Several recent papers for 3D object classification and detection can be roughly grouped as methods for object classification and detection without 3D pose estimation [32, 16, 14, 37, 5], and those with 3D pose estimation [27, 28, 21]. The former typically captures 3D object information by linking features across views in a discriminative learning framework. Compared to these previous work, our model is probabilistic (thus, the model parameters can be learned in a principled way) and requires less supervision (no need for viewpoint labeling as in [32, 16, 37, 5]).

Our model is built upon our previous paper [31]. In [31], we proposed a preliminary probabilistic model for representing multi-view object categories. Our method is superior to [31] in that: i) it requires far less supervision (no object part or region segmentation and viewpoint labels); ii) it allows much more accurate viewpoint representation. We propose a learning scheme in which the probabilistic model is initialized by learning from a cellphone video clip of an object class. To allow for accurate viewpoints alignment in training (without labels), we introduce a morphing parameter that enables the synthesis of the object appearance in arbitrary locations on the viewing sphere based on view morphing theories [30, 36]. We demonstrate the superiority of our model over [31] and other related algorithms in three types of recognition tasks: object detection, accurate viewpoint estimation and new viewpoint synthesis given a single test image.

3. Learning the probabilistic model

This section covers the learning of a 3D object category model from a short video clip of a single real world object and a set of unsorted images of the same object class collected from the Internet. Our goal is to build a probabilistic model that is a dense, multi-view representation of a 3D object class with minimal supervision[1]. We choose a part-based model for describing an object class (Fig. 2(b), Fig. 4). For each viewpoint, an object class is an ensemble of geometrically arranged parts, and each part is a collection of image patch features. Unlike [12, 18, 29], however, our model establishes explicit correspondences among parts across different viewpoints.

We start with a viewing sphere centered around the object. A large number of viewpoints (termed as *key views*) are sampled on the viewing sphere from a video sequence that portrays the object from a continuum of view points (Sec. 3.1). The 3D object generative model is constructed by using viewpoints and the object parts that are linked across different key views (Sec. 3.3). An important concept introduced here is the morphing parameter S, which allows the model to infer new poses when queried by an image of unseen viewpoint. The model is then updated using a set of images collected from the Internet (Sec. 3.4).

3.1. Parameterizing the viewing sphere

Most of the previous work in 3D object categorization has taken the approach of assuming a small number of discrete poses for the object class [32, 16, 37]. We argue that a dense representation of poses can provide a more accurate estimation of the object model (Fig. 8-Center). However, training a 3D object model by using human labeled poses is not only laborious and expensive, but also prone to errors because humans are not adept at quantifying 3D viewpoints [24]. Our goal is, therefore, to learn a dense representation without pose labels.

We circumvent the human labeling problem by initializing the dense multi-view representation of the 3D object class with a (cellphone) video clip of an object instance (Fig. 2(a)). We use the cellphone camera to take a short movie by walking around the object, allowing the cameraperson to change viewing angles by raising and lowering the device. We describe now how to parameterize the viewing sphere.

Obtaining key views. Given the short clip of video, we first apply a Lucas-Kanade tracker to obtain feature-level correspondences between every consecutive frames (Fig. 2(a)) [23]. For every set of linked frames[2], we sample uniformly a small number of them as "key views". Given a typical video clip, we obtain in the order of ~100 key views.

Triangulating the viewing sphere for view synthesis and defined $\{T, S, A\}$. Neighboring key views are grouped

[1] Only object bounding boxes are provided in the unsorted images.

[2] The linkage between two frames is considered broken when the percent of feature correspondences falls below an empirical threshold (in this case 20%).

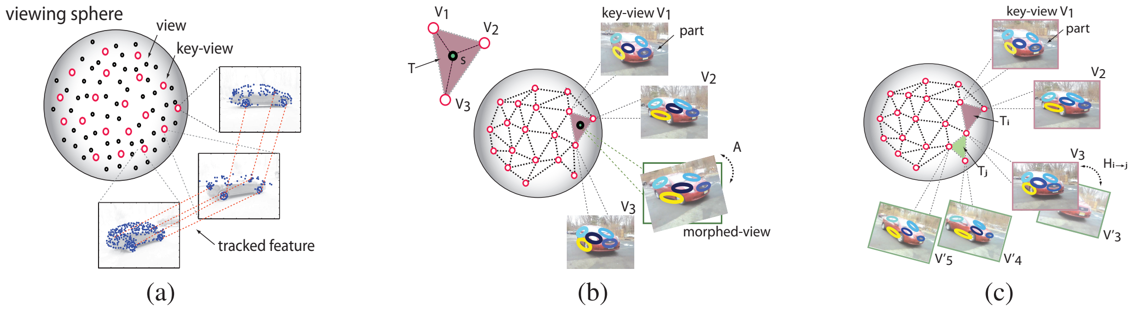


Figure 2: Schematic illustration of key concepts of our model. **(a)** Viewing sphere, key views and tracked video frames. Using a cellphone video clip of a single object, we obtain a dense sample of viewpoints (blue dots and red circles) on the viewing sphere. Features (dots on the car images) are tracked between consecutive video frames using the Lucas-Kanade algorithm [23]. Some of the tracks are shown in red dotted lines between two pictures. A subset of the viewpoints are assigned as key views (Sec. 3.1). **(b)** The viewing sphere and all the key views are parameterized as a triangle mesh (Sec. 3.1). We formulate the view synthesis problem based on the view morphing theory. A morphing parameter S interpolates and extrapolates the triplet of viewpoints V_i in a given viewpoint triangle T. The post-warping transformation of viewpoint alignment is denoted by A. **(c)** Illustration of a 3D geometrical constraint across viewpoint triangles. Given any key view on the viewing sphere, it is shared by two connected viewpoint triangles. An affine transformation constraint $H_{i \to j}$ is then enforced to ensure consistent part estimation across different views. This figure is best viewed in colors and with PDF magnification.

into triangles T such that the viewing sphere is covered by a triangle mesh (Fig. 2(b)). This parameterization allows the synthesis of new views within each triangle. Assume for now that three key views are lying on the same plane (i.e. the views are parallel or rectified). We denote such a plane as a *view plane*. Under the assumption that feature correspondences are available across key views, a new view within T can be synthesized by introducing an interpolating (*morphing*) parameter S and a homography A [30, 36], called *post-warping transformation*. S is a 3D vector in the simplex space that regulates the synthesis of the new view from the three key views. A enables the correct alignment (registration) between the synthesized view and a query view. If the assumption of parallel views does not hold (e.g. the key views forming the triangle are not close enough) we can use feature correspondences across the key views to align the key views to the plane formed by the triangle [36]. Note that different triangles may correspond to view planes that are not mutually parallel (Fig. 2(c)). In this case key views may need to be re-aligned and their viewpoints adjusted through a homographic transformation H (*pre-warping transformation*).

3.2. Obtaining candidate parts

We are now ready to learn the part-based generative model of a 3D object class (Fig.4). We define parts as regions within an object that: i) enclose discriminative appearance features that are frequently observed across different instances of the object class; ii) form a more-or-less planar region such that affine transformation becomes a good approximation when the part undergoes viewpoint changes. We use a modified J-Linkage clustering algorithm [33] to obtain candidate parts. In [33], given feature correspondences, the algorithm segments them into a number of planar regions so that the corresponding regions can be fitted by a unique affine transformation. Notice that tracks between every pair of views are equivalent to feature correspondences. We can therefore apply the J-Linkage algorithm to segment the tracks into planar regions for each pair of several neighboring key views. We then apply an agglomerative clustering algorithm to finalize the grouping of tracks to planar regions. Fig. 3 shows some sample results of this step.

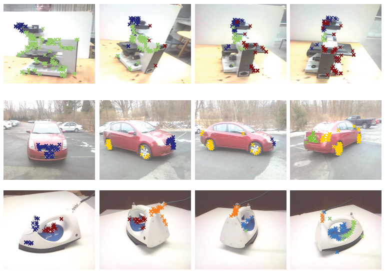


Figure 3: Examples of candidate parts of different object classes. Image patch features are denoted by "x". x's of the same color indicate that they belong to the same candidate part.

3.3. A probabilistic model representation and initial model learning

We use a probabilistic formulation to model a 3D object class (Fig. 4), built upon our work in [31]. We first describe the generative model in Sec. 3.3.1 and highlight the 3D geometrical constraints encoded in the model in Sec. 3.3.2. We then discuss the key differences between the current model and the model in [31]. In Sec. 3.3.4, we describe briefly how initial learning of the probabilistic model is done by using a short cellphone video clip of 3D objects.

3.3.1 The generative model for view morphing

Given an image, a set of N image patches are extracted and vector quantized into visual words[3]. Hence, for each patch the model observes its position X_n and visual word assignment Y_n (Fig.4). To generate image patches, a viewpoint is sampled from the parameterized viewing sphere. Recall that we parameterize a viewpoint by view triangle

[3] we use the same feature detector and descriptor as [31] to generate the codeword.

$T \sim Mult(\phi)$ and morphing parameter $S \sim Dir(\beta)$, where ϕ and β denote the parameters of the Multinomial and Dirichlet distributions respectively (Sec. 3.1). Given $\{T, S\}$, we are now ready to synthesize the sampled view and generate the parts and image patches. The model first generates a set of part location and appearance distribution parameters (θ and η), as well as a part proportion parameter (π). Image patches $\{X, Y\}$ within each part K can then be generated based on these parameters in the following steps.

Generate part parameters (θ, η, π). There are three sets of parameters governing the distribution of each object part K: part position (θ), part appearance (η) and part proportion (π), each of them depending on the view triangle T and morphing parameter S.

θ is further parameterized by a 2D Gaussian distribution of mean m_{TK} (part center) and covariance Σ_{TK} (part shape).

$$m_{TK}(S) = \textstyle\sum_{g=1}^{3} \hat{m}^{g}_{TK} \cdot S^{g} \qquad (1)$$

$$\Sigma_{TK}(S) = \hat{\Sigma}^{g^*}_{TK} \text{ , s.t. } g^* = G(S) = \underset{g=1,2,3}{\text{argmax}} S^{g} \qquad (2)$$

where $\hat{m}^{g}_{TK}$ and $\hat{\Sigma}^{g}_{TK}$ are the mean and covariance of part K in key view $g = \{1, 2, 3\}$ of the triangle T, and S^g is the morphing proportion of each of the three key views ($\sum_g S^g = 1$). $m_{TK}(S)$ denotes the generated new center of the morphed part. Intuitively, it is the linear combination of the centers of part K on the key views of T. $\Sigma_{TK}(S)$, the new covariance matrix of the morphed part, is approximated to be equal to the covariance on the closest key view in T, denoted by $\hat{\Sigma}^{g^*}_{TK}$.

Similarly, the part appearance parameter $\eta_{TK}(S)$ equals to $\hat{\eta}^{g^*}_{TK}$, the appearance parameter of part K of closest key view g^* to the sampled viewpoint in triangle T.

We sample part proportion parameter $\pi \sim Dir(\alpha_T)$. π governs the likelihood of the different parts that will appear under this view. For example, for a car model, π should be large for the wheel part in the side view but small in a frontal view. An object part K is then sampled from $Mult(\pi)$.

Generate image features $\{X_n, Y_n\}$. Given the part K, the position $\hat{X}$ of each image feature on the view plane of triangle T is generated from $\mathcal{N}(m_{TK}(S), \Sigma_{TK}(S))$. In the image plane, the feature location $X = A^{-1}\hat{X}$, where A is the post-warping affine transformation parameter. The appearance of each image feature Y_n is encoded by a codeword in a pre-obtained codebook. We obtain $Y_n \sim Mult(\eta_{TK}(S))$, where $\eta_{TK}(S)$ is the multinomial distribution parameter that governs the proportion of the codewords in each view given the object class.

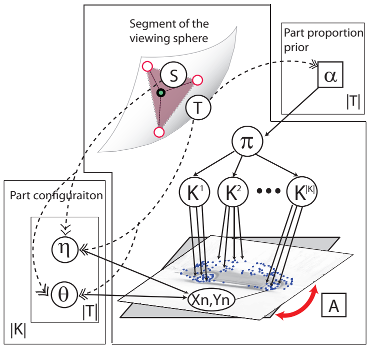


Figure 4: A schematic representation of our 3D object model. Each circular node represents a random variable, whereas each rectangular node represents a parameter. Solid arrows indicate conditional probability relationship between a pair of variables. Dashed arrows indicate the influence of the viewpoint triangle T and morphing parameter S on the variables. $\{X_n, Y_n\}$ indicate the position and appearance of an image patch feature. $K^1, K^2, \cdots, K^{|K|}$ are part assignment of image features. π is the part proportion parameter governed by the Dirichlet parameter α. θ and η are the part position and part appearance parameters describing each image feature $\{X_n, Y_n\}$. Finally A is the post-warping affine transformation parameter. Note that the graphical model does not depict the 3D geometrical constraints used by this model (Sec. 3.3.2).

Putting all the observable variables (X, Y, T, S) and latent variables (K, π) together with their corresponding parameters, we write down the joint probability of the model.

$$P(X, Y, T, S, K, \pi) = P(T|\phi)P(\pi|\alpha_T)P(S|\beta)$$
$$\textstyle\prod_n^N \{P(x_n|\theta_{TK_n}(S), A)P(y_n|\eta_{TK_n}(S))P(K_n|\pi)\} \quad (3)$$

3.3.2 3D geometric constraints

A fundamental difference between our model and a typical mixture of parts model is the explicit correspondence among parts across different views. This is done by applying two types of 3D geometric constraints on the model.

A. Within T constraints. Part configurations of key views should be consistent with each other by an affine transformation. Specifically, we use the tracks obtained by Lucas-Kanade algorithm between key views V_i and V_j in a triangle T to estimate the affine transformation $M^T_{i\to j}$, expecting $M^T_{i\to j}\hat{m}^i_{TK} = \hat{m}^j_{TK}$. This information is encoded as a penalty term (C in Eq.4) in our Variational EM algorithm described in Sec. 3.3.4. Thus, our probabilistic model learning process favors those part configurations that have consistent geometrical relationships across different views.

B. Across T constraints. As is shown in Fig. 2(c), each key view is shared by neighboring triangles $\{T_i, T_j, \ldots\}$. It is therefore important to make the correct correspondences between parts across the triangles. We estimate a transformation from T_i to T_j by using an affine transformation operation $H_{i\to j}$ and enforcing that $H_{i\to j}\hat{m}^{g_i}_{T_iK} = \hat{m}^{g_j}_{T_jK}$ for part K, where $\hat{m}^{g_i}_{T_iK}$ is the part center of the key view g_i in triangle T_i. Intuitively, this means that we can establish part correspondences by enforcing the centers of the same part in different planes defined by neighboring triangles to share the same configuration in the coordinate system of the key view. This constraint is encoded by F in Eq.4 in the Variational EM algorithm described in Sec. 3.3.4.

3.3.3 Differences between our model and [31]

We have made comparisons with [31] throughout the paper. Here we highlight two fundamental differences.

- The generative 3D object class model of views and parts in [31] encodes a small set of discrete viewpoints. Without a morphing variable S, it has no ability to recognize and synthesize a new pose. To our knowledge, we present here the first probabilistic representation of a 3D object class model that is capable of new viewpoint synthesis[4].
- Through a dense, triangle mesh representation of the viewing sphere, we introduce two convenient 3D geometrical constraints that are critical for establishing robust object part correspondences. These constraints are simple affine transformation constraints automatically estimated from tracks in the video clip. In contrast, [31] used an epipolar line constraint that are provided by human supervision. The training process is laborious and error-prone, and the result is less robust compared to our method (Fig.10).

3.3.4 Learning the initial model with a video clip

To learn a 3D object class representation, we first use a cellphone video clip of a single object to initialize the model. We see two advantages. First it enables us to obtain a robust initial model of parts and viewpoints by using a single object instance, making it easier for the model to learn and adjust to the intra-class variations in the ensuing training stage where it sees a large set of unsorted images. Second this method allows the algorithm to learn a 3D object class model with little human supervision, compared to all existing methods.

As a first step of learning, we initialize $\{T, S, A\}$ of each frame by using the image feature correspondences:

$$\{T, S, A\} = \underset{T,S,A}{\operatorname{argmax}} \textstyle\sum_{i \in (\text{tracks in } T)} \|Ax^i - \overline{x}^i_T S\|^2,$$

where $\overline{X}_T = \{\overline{x}^i_T \in R^{2\times3}\}$ is a set of tracks on the three key views in triangle T, and $X = \{x^i \in R^2\}$ are the correspondences in the frame. The estimated viewpoint parameters are treated as observed variables in Fig.4.

We are now ready to estimate the model variables K, π and model parameters θ, η[5] in Eq. 3. Recall the 3D geometric constraints introduced in Sec. 3.3.2, we therefore formulate the variational EM algorithm as an optimization problem [1]:

$$\text{maximize} \quad \lambda L(u) - (1-\lambda)\, C(u) \quad \text{s.t.} \quad F(u) = 0, \tag{4}$$

where L is the marginal log likelihood $P(X, Y|\cdot)$, C is the within triangle constraint function (Sec. 3.3.2(A)), F is the across triangle constraint function (Sec. 3.3.2(B)), u denotes all the model variables and parameters, and λ is the weight to balance the importance of the within triangle constraints vs the log likelihood.

We use a mean-field variational distribution to approximate the true posterior $P(T, S, K, \pi|X, Y)$ as follows:

$$q(T, S, K, \pi) = q(T|\delta) q(S|\epsilon) q(\pi|\gamma) \textstyle\prod_n^N q(K_n|\rho_n) \tag{5}$$

where δ denotes the variational parameter of Multinomial distribution of triangle T, ϵ and γ denote the variational parameters of the Dirichlet distribution which governs morphing parameter S and part proportion π respectively. ρ_n represents the variational parameter for the part assignment variable K_n. It is initialized by the candidate parts described in Sec. 3.2. $\{T, S, A\}$ are obtained in the preprocessing step through the video tracks.

We can now iterate between the following M- and E-steps for model parameter updates.

M-Step: 1. Part appearance parameter η update.

$$\hat{\eta}^{gw}_{tK} = \tfrac{N^{gw}_{tK}}{N^{g}_{tK}} \tag{6}$$

$$N^{gw}_{tK} = \textstyle\sum_{j \in (T_j = t, G(S_j) = g)} \sum_{n \in (y_{nj} = w)} \rho^K_{nj} \tag{7}$$

$$N^{g}_{tK} = \textstyle\sum_w N^{gw}_{tK} \tag{8}$$

where $\hat{\eta}^{gw}_{tK}$ is the probability that codeword w appears for part K on key view g in triangle index t. N^{gw}_{tK} is the sufficient statistics of the Multinomial distribution $Mult(\hat{\eta}^{g}_{tK})$.

2. Part position parameter $\theta = \{m, \Sigma\}$ updates. Since part center m is constrained by the 3D geometric constraints, the update of m is formulated as a quadratic programming problem with linear equality constraints which can be solve efficiently. The update of part shape Σ is formulated as a convex optimization problem, detailed in the technical report.

E-Step: 1. Part proportion π update. For each image, we update the variational distribution $q(\pi|\gamma)$ of part proportion π as $\gamma = \alpha_T + \hat{N}$, where $\hat{N} = \sum_{n=1}^N \rho_n$. Note $\hat{N} \in R^{|K|}$ is the sufficient statistics of the Dirichlet distribution $Dir(\gamma)$.

2. Part assignment K update. For each patch, we update the variational distribution $q(K_n|\rho_n)$ of part assignments, detailed in the technical report.

3.4. Incremental learning with unsorted images

Having initialized our 3D object class model with a video clip of a single object instance, we can now complete model learning through a set of unsorted images downloaded from the Internet. We propose an incremental learning procedure in the following three steps.

[4] [28] is the only other object class model capable of synthesizing new viewpoints. But without a coherent probabilistic formulation, the method relies on heuristics for learning the optimal values of the parameters.

[5] Due to space limitation, more detailed derivations can be found in an accompanying technical report on the authors website.

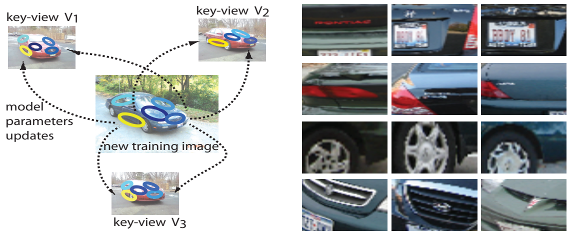


Figure 5: **Left.** Illustration of the updates for position parameter θ during incremental learning. As a new training image is assigned to the triangle T, new evidence on the sufficient statistics is produced which results in updating relevant model parameters of the key-views. **Right.** Examples of object parts illustrated by automatically cropped image regions from different training images. Three examples are shown for each part.

Initial viewpoint estimation. Given the set of video frames of the initial object instances as well as the all the training images seen so far, we first obtain the viewpoint $\{T, S, A\}$ of the new training image by matching it to the closest video frame. This is done in a similar image reranking scheme as proposed by [20][6].

Obtain object parts by variational inference. Given $\{T, A, S\}$ of the new training image, we can now run the updates of $\{m, \pi, K\}$ (See Sec. 3.3.4) to extract its object parts (some examples are shown in Fig.5-Right).

Model parameter $\{\eta, \theta\}$ updates. Given a new object instance j, appearance parameter $\hat{\eta}_{TK}^{G(S_j)w}$ can be updated according to Eq.6 by using the sufficient statistics $N_{T_jK}^{G(S_j)w} \leftarrow N_{T_jK}^{G(S_j)w} + \sum_{n\in(y_{nj}=w)} \rho_{nj}^{K}$, where $\{T_j, S_j\}$ are the viewpoint parameter of the object instance obtained from the matching algorithm, and w is the codeword index.

Part position parameter θ updates are detailed in the technical report. Fig. 5-Left is a schematic illustration of how this is done.

4. Experiments and results

We have introduced a new probabilistic multi-view representation for 3D object classes. With minimal supervision, our algorithm is capable of learning the 3D structures of this part-based model across viewpoints. We test now how our model can be used to perform three challenging recognition tasks: object detection in cluttered background, viewpoint classification upon detection, and new viewpoint synthesis given a single test image.

4.1. Object class detection

A robust visual recognition system needs to detect and categorize real-world objects under arbitrary viewpoints. Having trained a part-based 3D object model, we could now use this model to build a robust object class detector. Three datasets are used for evaluating this task: the car and bicycle classes in Savarese et al. [27], the car and bicycle classes in PASCAL VOC 2006 dataset [9], and 8 household object classes collected from the ImageNet [8]. We first describe briefly an object detector based on the learned object parts and 3D structure.

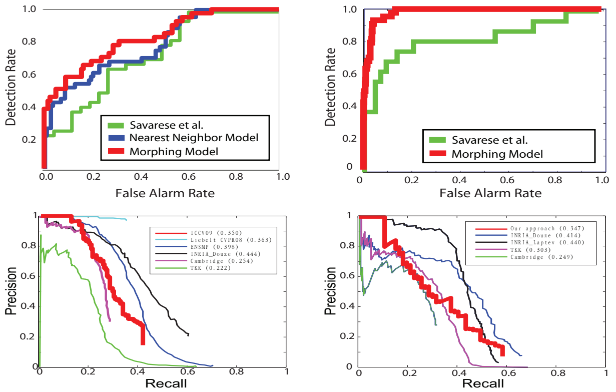


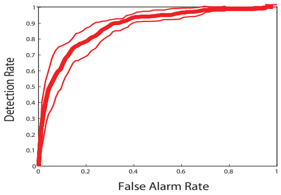
Figure 6: Object detection results using the 3DObjects dataset [27] (upper left & upper right) and the PASCAL VOC06 dataset [9] (lower left & lower right). **Upper Left.** We use an ROC curve to show the car detection results. Our model (red line) shows a performance of 82.3% measured by area under the curve (AUC), compared to 73.7% by using [27] (green line), and 78.1% by using [31] (blue line). **Upper Right.** Bicycle detection results. Our model (red line) obtains a 98.1% performance compared to 82.9% by [27] (green line).**Lower Left** Object detection using the PASCAL VOC06 car dataset. **Lower Right** Object detection using the PASCAL VOC06 bicyle dataset. We use precision-recall curves to show the results of our model (red line) compared with [21] and the detection result of the 2006 challenges [9]-INRIA_Douze , [9]-INRIA_Laptev, [9]-TKK, [9]-Cambridge, and [9]-ENSMP. Average precision (AP) scores are shown in the legends.

Given all training images, we first obtain the image regions of the corresponding parts by the method described in Sec. 3.4. We then train a random forest classifier for each part, by keeping the relative position information of the center of the object. Positive examples are sampled from each of the extracted part regions, whereas negative examples are sampled in regions outside of the object parts. Given a test image, we first apply the random forest classifier for each object part to obtain their candidate locations. Based on the 3D model trained for this object class, an object center is proposed for each of the candidate parts. Next, a generalized Hough transform voting scheme is used to locate candidate object locations [19]. Besides, to fuse evidences of part level together with object level, an object verification classifier is trained using a linear SVM. An object is represented as a spatial pyramid of codewords [17]. When testing, we reclassify a proposed object candidate by assigning the detection score as a linear combination of the original score and this object verification score.

Fig. 6 compare the detection results[7] of our model with other state-of-the-art algorithms on the 3DObjects dataset [27] and PASCAL VOC06 dataset [9] respectively. Our model outperforms [27] and [31] significantly on the

[6] Instead of the classification model used in [20], we use a KNN classifier with parzen window to do the matching.

[7] We choose an ROC measurement for the 3DObjects [27] dataset a precision-recall measurement for the PASCAL VOC06 dataset [9] in order to compare with the previous results.



Watch	84.9
Sewing Machine	98.1
Microscope	87.9
Swivelchair	91.2
Calculator	97.2
Flashlight	87.1
Teapot	86.4

Figure 7: **Left.** Object detection results using the Household Objects dataset. The thick red line shows the average ROC curve, whereas the thin red lines show the standard deviation over 8 classes. Average AUC score is 90.1%. **Right.** AUC score for each of the household object class.

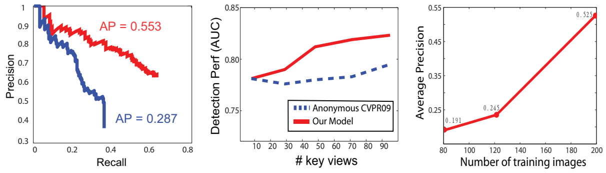
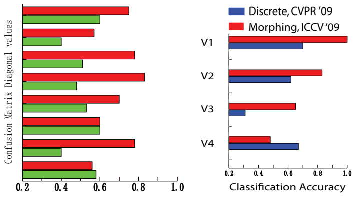


Figure 8: **Left.** Object detection with or without using the object parts. We use a car detection task (3DObjects dataset [27]) to show the performance difference between an object detector using the object parts learned by the 3D model (red line) and an object detector built without the object parts (blue line). **Center.** Effect of view synthesis for recognition via learning with the morphing parameter S and number of viewpoints. We demonstrate this by showing a binary detection task result (measured by AUC) versus the number of key views used by the model. Our model (red solid line) is compared with a discretized viewpoint model (blue dashed line) [31]. **Right.** Effect of incremental learning.

3DObjects dataset, and shows comparable results to most of the state-of-the-art methods on the PASCAL VOC06 dataset. Fig. 7 shows detection results on the new 3D object dataset of eight household objects. These items have significantly different image features compared to the often used car and bicycle datasets. We show very promising detection results in all eight classes of objects. Some example detection results are shown in Fig. 10.

An important contribution of our work is to propose a method that is able to learn these parts automatically and in turn use them for building an object class detector. In an object detection experiment using the 3DObjects dataset (car class), we show that an object detector built by using these proposed object parts significantly outperforms an object detector that does not use the parts (Fig. 8-Left).

In a separate experiment, we examine the effect of using dense viewpoint representation and view morphing framework for building 3D object models. Fig. 8-Center shows a detection experiment on the same car dataset measured by AUC. Our model (red curve) is compared with a discretized viewpoint model (blue curve) [31]. We observe two trends. For both of these models, as the number of viewpoints increases during training, the detection performance increases. But our model performs consistently better than [31] even given the same number of viewpoints. This is due to the effect of the morphing parameter S in our model. It is capable of synthesizing intermediate views to mitigate the unavoidable discrepancies existing in a discretized representation.



Watch	61.9
Sewing Machine	71.4
Microscope	63.9
Iron	73.5
Swivelchair	58.6
Calculator	69.2
Flashlight	68.4
Teapot	60.0

Figure 9: Viewpoint classification results: **Left.** 8-view classification of the 3DObjects car dataset. We compare our model (red bar) with [28] (green bar). **Center.** 4-view classification of the PASCAL VOC06 car dataset. Our model (red bar) is compared with with [31] (blue bar). **Right.** Viewpoint classification accuracy for the household objects dataset.

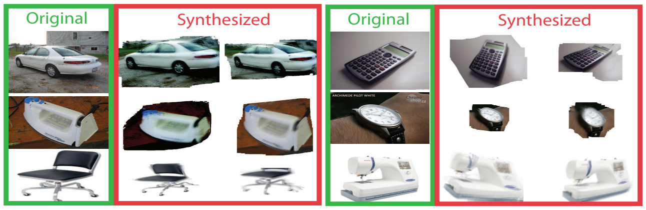


Figure 11: New views can be synthesized given a single test image. The further right column (green) of each triplet of images indicates the original test image. The other two columns (red) are two synthesized views.

We also evaluate the effect of incremental learning by using detection of car on the 3DObjects dataset. Fig.8-Right shows that as the number of training images increases, our model is capable of taking advantage of the additional data and continues to achieve higher performances.

4.2. Object viewpoint classification

Our model is capable of predicting the viewpoint of a query object by estimating $\{T, S, A\}$. We show examples of the viewpoint classification results in Fig.10. Similar to the object detection experiment, we evaluate our model using three datasets: the 3DObjects dataset [27], PASCAL VOC06 dataset [9] and the 8 classes of Household Objects dataset. The results of the first two are shown in Fig. 9-Left&Center[8], whereas results of household objects are shown in Fig.9-Right[9]. On the 3DObjects our model significantly outperforms [28], largely due to its richer representation and view morphing ability to refine pose estimations.

4.3. Viewpoint synthesis

Given a single test image, our model has the ability to recognize its object class and synthesize an unseen view. This is done through the recognition of object parts as described in Sec. 3.4. Given the extracted parts, we are able to synthesize any view specified by the viewpoint parameters $\{T, S, A\}$. We show in Fig. 11 several synthesized views of test images.

[8] The numbers of views are provided by the datasets. For 3DObjects, there are 8 viewing angles, 3 scales and 2 heights. For PASCAL VOC06, there are 4 viewpoints.

[9] For convenience, we discretize all views into 8 canonical views when evaluating the performances.

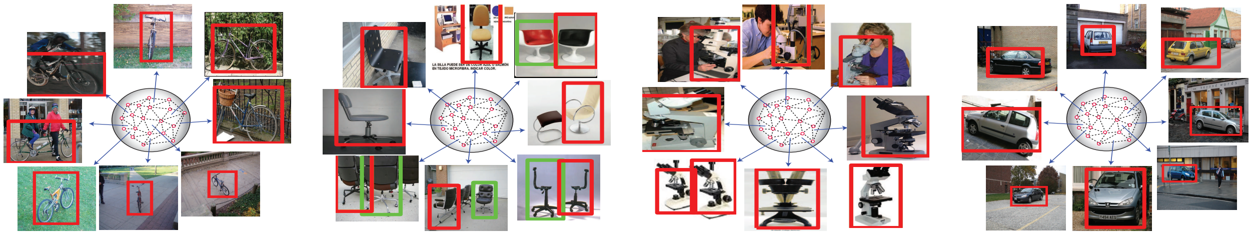


Figure 10: Examples of viewpoint estimation for bicycle [27, 9], swivel chair, microscope and car [27, 9]. Blue arrows indicate the viewpoint T for the detected object (in red bounding box). Green bounding box indicates correct detections of the objects, but in a different viewpoint.

5. Conclusion

We have proposed a 3D object class model based on a dense, multiview representation of the viewing sphere. A morphing parameter S is introduced to allow our model to recognize and synthesize unseen views. Our experiments show promising results in object detection, viewpoint classification and synthesis tasks. For future work, we would like to incorporate a more discriminative learning process into the model building step, as well as to combine viewpoint synthesis into incremental learning framework to maximize the usage of the data.

Acknowledgments

The authors would like to thank Jia Deng and Barry Chai for their help for data collection. The authors especially thank Jia Li 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

[1] D. M. Blei. Variational methods for the dirichlet process. *ICML*, 2004. 5

[2] K. Bowyer and R. Dyer. Aspect graphs: An introduction and survey of recent results. *Int. J. of Imaging Systems and Technology*, 1990. 2

[3] M. Brown and D. Lowe. Unsupervised 3D object recognition and reconstruction in unordered datasets. *3DIM05*, 2005. 2

[4] P. Carbonetto, G. Dorkó, C. Schmid, H. Kück, and N. Freitas. Learning to recognize objects with little supervision. *IJCV*, 2008. 1

[5] H. Chiu, L. Kaelbling, and T. Lozano-Perez. Virtual training for multi-view object class recognition. *CVPR*, 2007. 1, 2

[6] C. Cyr and B. Kimia. A similarity-based aspect-graph approach to 3D object recognition. *IJCV*, 2004. 2

[7] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. *CVPR*, 2005. 1

[8] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei. ImageNet:A Large-Scale Hierarchical Image Database. *CVPR*, 2009. 6

[9] M. Everingham, A. Zisserman, C. K. I. Williams, and L. Van Gool. The PASCAL Visual Object Classes Challenge 2006 Results. Technical Report, PASCAL Network, 2006. 6, 7, 8

[10] L. Fei-Fei, R. Fergus, and A. Torralba. Recognizing and learning object categories. *CVPR*, 2007 Short Course. 1

[11] P. Felzenszwalb, D. Mcallester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. *CVPR*, 2008. 1

[12] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. *CVPR*, 2003. 1, 2

[13] V. Ferrari, T. Tuytelaars, and L. Van Gool. Simultaneous object recognition and segmentation from single or multiple model views. *IJCV*, 2006. 2

[14] D. Hoeim, C. Rother, and J. Winn. 3d layoutcrf for multi-view object class recognition and segmentation. *CVPR*, 2007. 1, 2

[15] J. Koenderink and A. van Doorn. The singularities of the visual mappings. *Biological Cybernetics*, 1976. 2

[16] A. Kushal, C. Schmid, and J. Ponce. Flexible object models for category-level 3d object recognition. *CVPR*, 2007. 1, 2

[17] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. *CVPR*, 2006. 6

[18] B. Leibe and B. Schiele. Combined object categorization and segmentation with an implicit shape model. *Workshop on satistical learning in computer vision*, Prague, Czech Republic, 2004. 2

[19] B. Leibe and B. Schiele. Scale invariant object categorization using a scale-adaptive mean-shift search. *DAGM'04 Annual Pattern Recognition Symposium*, 2004. 6

[20] L.-J. Li, G. Wang, and L. Fei-Fei. Optimol: automatic online picture collection via incremental model learning. *CVPR*, 2007. 6

[21] J. Liebelt, C. Schmid, and K. Schertler. Viewpoint-independent object class detection using 3d feature maps. *CVPR*, 2008. 1, 2, 6

[22] D. Lowe. Object recognition from local scale-invariant features. *ICCV*, 1999. 2

[23] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. *DARPA* Image Understanding Workshop, 1981. 2, 3

[24] S. Palmer. *Vision Science*. 1999. 2

[25] D. Ramanan. to verify object hypotheses. *CVPR*, 2007. 1

[26] 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. 2

[27] S. Savarese and L. Fei-Fei. 3d generic object categorization, localization and pose estimation. *ICCV*, 2007. 1, 2, 6, 7, 8

[28] S. Savarese and L. Fei-Fei. View synthesis for recognizing unseen poses of object classes. *ECCV*, 2008. 1, 2, 5, 7

[29] H. Schneiderman and T. Kanade. A statistical approach to 3D object detection applied to faces and cars. *CVPR*, 2000. 2

[30] S. Seitz and C. Dyer. View morphing. *SIGGRAPH*, 1996. 2, 3

[31] M. Sun, H. Su, S. Savarese and L. Fei-Fei. A Multi-View Probabilistic Model for 3D Object Classes. *CVPR*, 2009. 1, 2, 3, 5, 6, 7

[32] A. Thomas, V. Ferrari, B. Leibe, T. Tuytelaars, B. Schiele, and L. Van Gool. Towards multi-view object class detection. *CVPR*, 2006. 1, 2

[33] R. Toldo and A. Fusiello. Robust multiple structures estimation with j-linkage. *ECCV*, 2008. 3

[34] S. Ullman and R. Basri. Recognition by linear combination of models. Technical report, Cambridge, MA, USA, 1989. 2

[35] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. *CVPR*, 2001. 1

[36] J. Xiao and M. Shah. Tri-view morphing. *CVIU*, 2004. 2, 3

[37] P. Yan, D. Khan, and M. Shah. 3d model based object class detection in an arbitrary view. *ICCV*, 2007. 1, 2

[38] A. Zisserman. An exemplar model for learning object classes. *CVPR*, 2007. 1