View Synthesis for Recognizing Unseen Poses of Object Classes

Silvio Savarese[1] and Li Fei-Fei[2]

[1] Department of Electrical Engineering, University of Michigan at Ann Arbor
[2] Department of Computer Science, Princeton University

Abstract. An important task in object recognition is to enable algorithms to categorize objects under arbitrary poses in a cluttered 3D world. A recent paper by Savarese & Fei-Fei [1] has proposed a novel representation to model 3D object classes. In this representation stable parts of objects from one class are linked together to capture both the appearance and shape properties of the object class. We propose to extend this framework and improve the ability of the model to recognize poses that have not been seen in training. Inspired by works in single object view synthesis (e.g., Seitz & Dyer [2]), our new representation allows the model to synthesize novel views of an object class at recognition time. This mechanism is incorporated in a novel two-step algorithm that is able to classify objects under arbitrary and/or unseen poses. We compare our results on pose categorization with the model and dataset presented in [1]. In a second experiment, we collect a new, more challenging dataset of 8 object classes from crawling the web. In both experiments, our model shows competitive performances compared to [1] for classifying objects in unseen poses.

1 Introduction

An important goal in object recognition is to be able to recognize an object or an object category given an arbitrary view point. Humans can do this effortlessly under most conditions. Consider the search for your car in a crowded shopping center parking lot. We often need to look around 360 degrees in search of our vehicle. Similarly, this ability is crucial for a robust, intelligent visual recognition system. Fig. 1 illustrates the problem we would like to solve. Given an image containing some object(s), we want to 1) categorize the object as a car (or a stapler, or a computer mouse), and 2) estimate the pose (or view) of the car. Here by 'pose', we refer to the 3D information of the object that is defined by the viewing angle and scale of the object (i.e. a particular point on the viewing sphere represented in Fig. 6). If we have seen this pose in the training time, and have a way of modeling such information, the problem is reduced to matching the known model with the new image. This is the approach followed by a number of existing works where either each object class is assumed to be seen under an unique pose [3,4,5,6,7,8,9,10] or a class model is associated to a specific pose giving rise to mixture models [11,12,13]. But it is not necessarily possible for an algorithm to have been trained with all views of the objects. In many situations,

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part III, LNCS 5304, pp. 602–615, 2008.

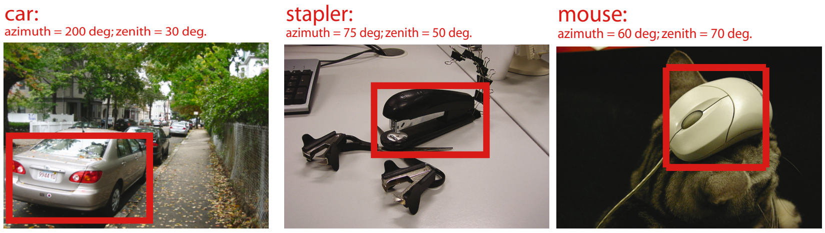

car:
azimuth = 200 deg; zenith = 30 deg.
stapler:
azimuth = 75 deg; zenith = 50 deg.
mouse:
azimuth = 60 deg; zenith = 70 deg.

Fig. 1. Categorize an Object Given An Unseen View. azimuth: [front,right,back, left]= $[0, 90, 180, 270]^o$; zenith: [low, med., high]= $[0, 45, 90]^o$

training is limited (either by the number of examples, or by the coverage of all possible poses of the object class); it is therefore important to be able to extrapolate information and make the best guess possible given this limitation. This is the approach we present in this paper.

In image base rendering, new view synthesis (morphing) have been an active and prolific area of research [14,15,16]. Seitz & Dyer [2] proposed a method to morph two observed views of an object into a new, unseen view using basic principles of projective geometry. Other researchers explored similar formulations [17,18] based on multi-view geometry [19] or extended these results to 3-view morphing techniques [20]. The key property of view-synthesis techniques is their ability to generate new views of an object *without* reconstructing its actual 3D model. It is unclear, however, whether these can be useful *as is* for recognizing unseen views of object categories under very general conditions: they were designed to work on single object instances (or at most 2), with no background clutter and with given features correspondences across views (Figs. 6–10 of [2]). In our work we try to inherit the view-morphing machinery while generalizing it to the case of object categories. On the opposite side of the spectrum, several works have addressed the issue of single object recognition by modeling different degree of 3D information. Again, since these methods achieve recognition by matching local features [21,22,23,24] or group of local features [25,26] under rigid geometrical transformations, they can be hardly extended for handling object classes.

Recently, a number of works have proposed interesting solutions for capturing the multi-view essence of an object category [1,27,28,29,30,31,32]. These techniques bridge the gap between models that represent an object category from just a single 2D view and models that represent single object instances from multiple views. Among these, [32] presents an interesting methodology for repopulating the number of views in training by augmenting the views with synthetic data. In [1] a framework was proposed in which stable parts of objects from one class are linked together to capture both the appearance and shape properties of the object class. Our work extends and simplifies the representation in [1]. Our critical contributions are:

- We propose a novel method for representing and synthesizing views of object classes that are not present in training. Our view-synthesis approach is inspired by previous research on view morphing and image synthesis from multiple views. However, the main contribution of our approach is that the synthesis takes place at the categorical level as opposed to the single object level (as previously explored).

- We propose a new algorithm that takes advantage of our view-synthesis machinery for recognizing objects seen under arbitrary views. As opposed to [32] where training views are augmented by using synthetic data, we synthesize the views at *recognition* time. Our experimental analysis validates our theoretical findings and shows that our algorithm is able to successfully estimate object classes and poses under very challenging conditions.

2 Model Representation for Unseen Views

We start with an overview of the overall object category model [1] in Sec. 2.1 and give details of our new view synthesis analysis in Sec. 2.2.

2.1 Overview of the Savarese et al. Model [1]

Fig. 2 illustrates the main ideas of the model proposed by [1]. We use the car category as an example for an overview of the model. There are two main components of the model: the *canonical parts* and the *linkage structure* among the canonical parts. A *canonical part* P in the object class model refers to a region of the object that tends to occur frequently across different instances of the object class (e.g. rear bumper of a car). It is automatically determined by the model. The canonical parts are regions containing multiple features in the images, and are the building blocks of the model. As previous research has shown, a part based representation [26,28,29] is more stable for capturing the appearance variability across instances of objects. A critical property introduced in [1] is that the canonical part retains the appearance of a region that is viewed most frontally on the object. In other words, a car's rear bumper could render different appearances under different geometric transformations as the observer moves around the viewing sphere (see [1] for details). The canonical part representation of the car rear bumper is the one that is viewed the most frontally (Fig. 2(a)).

Given an assortment of canonical parts (e.g. the colored patches in Fig. 2(b)), a *linkage structure* connects each pair of canonical parts $\{P_j, P_i\}$ if they can be both visible at the same time (Fig. 2(c)). The linkage captures the relative position (represented by the 2×1 vector $\mathbf{t}_{ij}$) and change of pose of a canonical part given the other (represented by a 2×2 homographic transformation $\mathcal{A}_{ij}$). If the two canonical parts share the same pose, then the linkage is simply the translation vector $\mathbf{t}_{ij}$ (since $\mathcal{A}_{ij} = \mathbf{I}$). For example, given that part P_i (left rear light) is canonical, the pose (and appearance) of all connected canonical parts must change according to the transformation imposed by $\mathcal{A}_{ij}$ for $j = 1 \cdots N, j \neq i$, where N is the total number of parts connected to P_i. This transformation is depicted in Fig. 2(c) by showing a slanted version of each canonical part (for details of the model, the reader may refer to [1]).

We define a *canonical view* V as the collection of canonical parts that share the same view V (Fig. 2(c)). Thus, each pair of canonical parts $\{P_i, P_j\}$ within

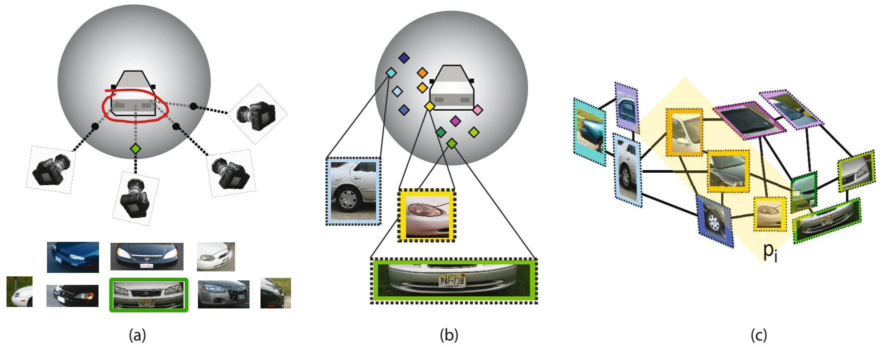


Fig. 2. Model Summary. Panel a: A car within the viewing sphere. As the observer moves on the viewing sphere the same part produces different appearances. The location on the viewing sphere where the part is viewed the most frontally gives rise to a canonical part. The appearance of such canonical part is highlighted in green. **Panel b:** Colored markers indicate locations of other canonical parts. **Panel c:** Canonical parts are connected together in a linkage structure. The linkage indicates the relative position and change of pose of a canonical part given the other (if they are both visible at the same time). This change of location and pose is represented by a translation vector and a homographic transformation respectively. The homographic transformation between canonical parts is illustrated by showing that some canonical parts are slanted with respected to others. A collection of canonical parts that share the same view defines a canonical view (for instance, see the canonical parts enclosed in the area highlighted in yellow.

V is connected by $\mathcal{A}_{ij} = \mathbf{I}$ and a translation vector $\mathbf{t}_{ij}$. We can interpret a canonical view V as a subset of the overall linkage structure of the object category. Notice that by construction a canonical view may coincide with one of the object category poses used in learning. However, not all the poses used in learning will be associated to a canonical view V. The reason is that a canonical view is a collection of canonical parts and each canonical part summarizes the appearance variability of an object category part under different poses. The relationship of parts within the same canonical view is what previous literature have extensively used for representing 2D object categories from single 2D views (e.g. the constellation models [4,6]). The linkage structure can be interpreted as its generalization to the multi-view case. Similarly to other methods based on constellations of features or parts, the linkage structure of canonical parts is robust to occlusions and background clutter.

2.2 Representing an Unseen View

The critical question is: how can we represent (synthesize) a novel non-canonical view from the set of canonical views contained in the linkage structure? As we will show in Sec. 3, this ability becomes crucial if we want to recognize an object category seen under an arbitrary pose. Our approach is inspired by previous research on view morphing and image synthesis from multiple views. We show that it is possible to use a similar machinery for synthesizing appearance, pose and position of canonical parts from two or more canonical views. Notice that the output of this representation (synthesis) is a novel view of the object *category*, not just a novel view of a single object instance, whereas all previous morphing techniques are used for synthesizing novel views of single objects.

Representing Canonical Parts. In [1], each canonical part is represented by a distribution of feature descriptors along with their x, y location within the part. In our work, we simplify this representation and describe a canonical part P by a convex quadrangle B (e.g., the bounding box) enclosing the set of features. The appearance of this part is then characterized by a *bag of codewords* model [5] - that is, a normalized histogram h of vector quantized descriptors contained in B. Our choice of feature detectors and descriptors is the same as in [1]. A standard K-means algorithm can be used for extracting the codewords. B is a 2×4 vector encoding the $b = [x, y]^T$ coordinates of the four corners of the quadrangle, i.e. $B = \begin{bmatrix} b_1 \ldots b_4 \end{bmatrix}$; h is a $M \times 1$ vector, where M is the size of the vocabulary of the vector quantized descriptors. Given a linked pair of canonical parts $\{P_i, P_j\}$ and their corresponding $\{B_i, B_j\}$, relative position of the parts $\{P_i, P_j\}$ is defined by $\mathbf{t}_{ij} = c_i - c_j$, where the centroid $c_i = \frac{1}{4} \sum_k b_k$; the relative change of pose is defined by A_{ij} which encodes the homographic transformation acting on the coordinates of B_i. This simplification is crucial for allowing more flexibility in handling the synthesis of novel non-canonical views at the categorical level.

View Morphing. Given two views of a 3D object it is possible to synthesize a novel view by using view-interpolating techniques without reconstructing the 3D object shape. It has been shown that a simple linear image interpolation (or appearance-morphing) between views do not convey correct 3D rigid shape transformation, unless the views are parallel (that is, the camera moves parallel to the image planes) [15]. Moreover, Seitz & Dyer [2] have shown that if the camera projection matrices are known, then a geometrical-morphing technique can be used to synthesize a new view even without having parallel views. However, estimating the camera projection matrices for the object category may be very difficult in practice. We notice that under the assumption of having the views in a neighborhood on the viewing sphere, the cameras can be approximated as being *parallel*, enabling a simple linear interpolation scheme (Fig. 3). Next we show that by combining appearance and geometrical morphing it is possible to synthesize a novel view (meant as a collection of parts along with their linkage) from two or more canonical views.

Two-View Synthesis. We start by the simpler case of synthesizing from two canonical views V^n and V^m. A synthesized view V^s can be expressed as a collection of linked parts morphed from the corresponding canonical parts belonging to V^n and V^m. Specifically, a pair of linked parts $\{P_i^s, P_j^s\} \in V^s$ can be synthesized from the pair $\{P_i^n \in V^n, P_j^m \in V^m\}$ if and only if P_i^n and P_j^m are linked by the homographic transformation $\mathcal{A}_{ij} \neq I$ (Fig. 3). If we represent $\{P_i^s, P_j^s\}$ by the quadrangles $\{B_i^s, B_j^s\}$ and the histograms $\{h_i^s, h_j^s\}$ respectively, a new view is expressed by:

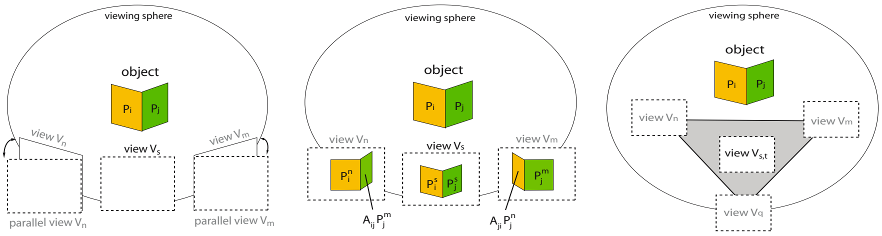


Fig. 3. View Synthesis. Left: If the views are in a neighborhood on the viewing sphere, the cameras can be approximated as being *parallel*, enabling a linear interpolation scheme. **Middle:** 2-view synthesis: A pair of linked parts $\{P_i^s, P_j^s\} \in V^s$ is synthesized from the pair $P_i^n \in V^n$, and $P_j^m \in V^m$ if and only if P_i^n and P_j^m are linked by the homographic transformation $\mathcal{A}_{ij} \neq I$. **Right:** 3-view synthesis can take place anywhere within the triangular area defined by the 3 views.

$$B_i^s = (1-s)B_i^n + s\mathcal{A}_{ij}B_i^n; \qquad B_j^s = sB_j^m + (1-s)\mathcal{A}_{ji}B_j^m; \tag{1}$$

$$h_i^s = (1-s)h_i^n + sh_i^m; \qquad h_j^s = sh_j^n + (1-s)h_j^m; \tag{2}$$

The relative position between $\{P_i^s, P_j^s\}$ is represented as the difference $\mathbf{t}_{ij}^s$ of the centroids of B_i^s and B_j^s. $\mathbf{t}_{ij}^s$ may be synthesized as follows:

$$\mathbf{t}_{ij}^s = (1-s)\mathbf{t}_{ij}^n + s\mathbf{t}_{ij}^m \tag{3}$$

In summary, Eqs. 1 and 3 regulate the synthesis of the linkage structure between the pair $\{P_i^s, P_j^s\}$; whereas Eqs. 2 regulate the synthesis of their appearance components. By synthesizing parts for all possible values of i and j we can obtain a set of linked parts which give rise to a new view V^s between the two canonical views V^n and V^m. Since all canonical parts in V^n and V^m (and their linkage structures) are represented at the categorical level, this property is inherited to the new parts $\{P_i^s, P_j^s\}$, thus to V^s.

Three-View Synthesis. One limitation of the interpolation scheme described in Sec. 2.2 is that a new view can be synthesized only if it belongs to the linear camera trajectory from one view to the other. By using a bi-linear interpolation we can extend this to a novel view from 3 canonical views. The synthesis can take place anywhere within the triangular area defined by the 3 views (Fig. 3) and is regulated by two interpolating parameters s and t. Similarly to the 2-view case, 3-view synthesis can be carried out if and only if there exist 3 canonical parts $P_i^n \in V^n$, $P_j^m \in V^m$, and $P_k^q \in V^q$ which are pairwise linked by the homographic transformations $\mathcal{A}_{ij} \neq I$, $\mathcal{A}_{ik} \neq I$ and $\mathcal{A}_{jk} \neq I$. The relevant quantities can be synthesized as follows:

$$B_i^{st} = [\,(s-1)I \;\; sI\,] \begin{pmatrix} B_i^n & \mathcal{A}_{ik}B_i^n \\ \mathcal{A}_{ij}B_i^n & \mathcal{A}_{ik}\mathcal{A}_{ij}B_i^n \end{pmatrix} \begin{bmatrix} (1-t)I \\ t\,I \end{bmatrix} \tag{4}$$

$$h_i^{st} = [\,(s-1)I \;\; sI\,] \begin{pmatrix} h_i^n & h_i^q \\ h_i^m & h_i^p \end{pmatrix} \begin{bmatrix} (1-t)I \\ t\,I \end{bmatrix} \tag{5}$$

$$\mathbf{t}_{ij}^{st} = \left[\, (s-1)I \;\; sI \,\right] \begin{pmatrix} \mathbf{t}_{ij}^{n} & \mathbf{t}_{ik}^{q} \\ \mathbf{t}_{ij}^{m} & \mathbf{t}_{ij}^{m} + \mathbf{t}_{ik}^{q} - \mathbf{t}_{ij}^{n} \end{pmatrix} \begin{bmatrix} (1-t)I \\ t\,I \end{bmatrix} \tag{6}$$

Analogous equations can written for the remaining indexes.

3 Recognizing Object Class in Unseen Views

Sec. 2.2 has outlined all the critical ingredients of the model for representing and synthesizing new views. We discuss here an algorithm for recognizing pose and

Algorithm step 1

```
$I \leftarrow$ list of parts extracted from test image
for each model $C$
    for each canonical view $V \in C$
        $[R(n), V^*(n)] \leftarrow$ MatchView($V, C, I$); % return similarity $R$
        $n$ ++;
$L \leftarrow$ KMinIndex($R$) % return shortlist $L$

MatchView($V, C, I$)
for each canonical part $P \in V$
    $M(p) \leftarrow$ MatchKPart ($P, I$)); % return K best matches
    $p$ ++;
for each canonical part $\bar{P} \in C$ linked to $V$
    $\bar{M}(q) \leftarrow$ MatchKPart ($\bar{P}, I$); % return K best matches
    $q$ ++;
$[M^*, \bar{M}^*] \leftarrow$ Optimize($V, M, \bar{M}$);
$V^* \leftarrow$ GenerateTestView($M^*, \bar{M}^*, I$);
$R \leftarrow$ Distance($V, V^*$);
return $R$, $V^*$;
```

Fig. 4. Pseudocode of the step 1 algorithm. MatchView(V, C, I) returns the similarity score between V and I. KminIndex() returns pointers to the the K smallest values of the input list. MatchKPart (P, I) returns the best K candidate matches between P and I. A match is computed by taking into account the appearance similarity S_a between two parts. S_a is computed as the distance between the histograms of vector quantized features contained in the corresponding part's quadrangles B. Optimize($V, M, \bar{M}$) optimizes over all the matches and returns the best set of matches $M^*, \bar{M}^*$ from the candidate matches in $M, \bar{M}$. The selection is carried out by jointly minimizing the overall appearance similarity S_a (computed over the candidate matches) and the geometrical similarity S_g (computed over pairs of candidate matches). S_g is computed by measuring the distance between the relative positions $\mathbf{t}_{ij}$, $\bar{\mathbf{t}}_{ij}$. GenerateTestView($M^*, \bar{M}^*, I$) returns a linkage structure of parts (B, appearances h and relative positions $\mathbf{t}$) given $M^*, \bar{M}^*$. This gives rise to the estimated matched *view* V^* in the test image. Distance(V_i, V_j) returns an estimate of the overall combined appearance and geometrical similarity $S_a + S_g$ between the linkage structures associated to V_i, V_j. S_a is computed as in MatchKPart over all the parts. S_g is computed as the geometric distortion between the two corresponding linkage structures.

Algorithm step 2

```
for each canonical view $V \in L$
    $V^* \leftarrow L(l)$
    $V' \leftarrow$ FindClosestView($V, C$);
    $V'' \leftarrow$ FindSecondClosestView($V, C$);
    for each 2-view synthesis parameter $s$
        $V^s \leftarrow$ 2-ViewSynthesis($V, V', s$);
        $R(s) \leftarrow$ Distance($V^s, V^*$);
    for each 3-view synthesis parameters $s$ and $t$
        $V^{s,t} \leftarrow$ 3-ViewSynthesis($V, V', V'', s, t$);
        $R(s,t) \leftarrow$ Distance($V^{s,t}, V^*$);
    $L(l) \leftarrow$ Min($R$);
    $l$ ++;
$[C_w V_w] \leftarrow$ MinIndex($L$);
```

Fig. 5. Pseudocode of the step 2 algorithm. FindClosestView(V, C) (FindSecondClosestView(V, C)) returns the closest (second closest) canonical pose on the viewing sphere. 2-ViewSynthesis(V, V', s) returns a synthesized view between the two views V, V' based on the interpolating parameters s. 3-ViewSynthesis(V, V', s, t) is the equivalent function for three view synthesis. C_w and V_w are the winning categories and poses respectively.

categorical membership of a query object seen under arbitrary view point. We consider a two-step recognition procedure. The first step is a modified version of [1]. The output of this algorithm is a short list of the K best model views across all views and all categories. The second step is a novel algorithm that refines the error scores of the short list by using the view-synthesis scheme.

3.1 A Two-Step Algorithm

In the first step (Fig. 4), we want to match the query image with the best object class model and pose. For each model, we find hypotheses of canonical parts consistent with a certain canonical view of an object model. Given such canonical parts, we infer the appearance, pose and position of other parts that are not seen in their canonical view (*MatchView* function). This information is encoded in the object class linkage structure. An optimization process finds the best combination of hypothesis over appearance and geometrical similarity (*Optimize*). The output is a similarity score as well as a set of matched parts and their linkage structure (the estimated matched *view* V^*) in the test image. The operation is repeated for all possible canonical views and for all object class models. Finally, we create a short list of the N best canonical views across all the model categories ranked according to their similarity (error) scores. Each canonical view is associated to its own class model label. The complexity of step-1 is $O(N^2 N_v N_c)$, where N is the total number of canonical parts (typically, 200–500); N_v = number of views per model; N_c = number of models.

In the second step (Fig. 5), we use the view synthesis scheme (Sec. 2.2) to select the final winning category and pose from the short list. The idea is to consider a canonical view from the short list, pick up the nearest (or two nearest) canonical pose(s) on the corresponding model viewing sphere (*FindClosestView* and *FindSecondClosestView*), and synthesize the intermediate views according to the 2-view-synthesis (or 3-view-synthesis) procedure for a number of values of s (s, t) (*2-ViewSynthesis* and *3-ViewSynthesis*). For each synthesized view, the similarity score is recomputed and the minimum value is retained. We repeat this procedure for each canonical view in the short list. The canonical view associated with the lowest score gives the winning pose and class label. The complexity of step-2 is just $O(N_l N_s)$, where N_l is the size of the short list and N_s is the number of interpolating steps (typically, 5–20).

4 Experiments and Results

In this section, we show that our algorithm is able to successfully recognize an object class viewed under a pose that is not seen during training. In addition to classification, we also measure the accuracy in pose estimation of an object.

4.1 Experiment I: Comparison with [1]

In the first set of experiments we compare the performances of our algorithm with those reported in [1]. We use the same dataset as in [1,33] and the same learning and testing methodology. The dataset comprises images of 8 different object categories, each containing 10 different instances. Each of these are photographed under a range of poses, described by a pair of azimuth and zenith angles (i.e., the angular coordinates of the observer on the viewing sphere, Fig. 6) and distance (or scale). The total number of angular poses in this dataset is 24: 8 azimuth angles and 3 zenith angles. Each pose coordinate is kept identical across instances and categories. Thus, the number and type of poses in the test set are the same as in the training set. The data set is split into a training and test set as in [1].

To assess the performance of our algorithm to recognize unseen views, we train both the model in [1] and ours by using a reduced set of poses in training. The

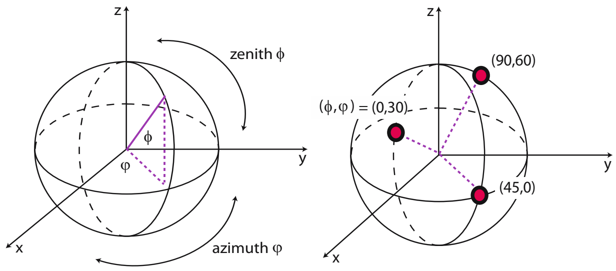
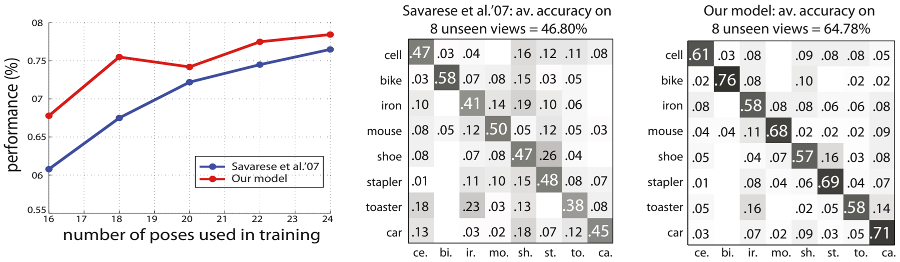


Fig. 6. Left: An object pose is represented by a pair of azimuth and zenith angles. **Right:** Some of the unseen poses tested during our recognition experiments (Fig. 7).



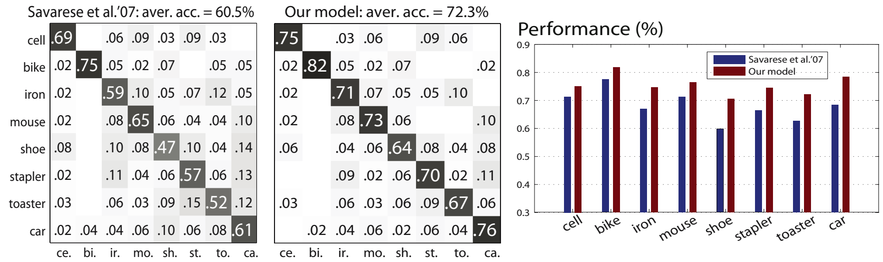
Savarese et al.'07: av. accuracy on 8 unseen views = 46.80%

	ce.	bi.	ir.	mo.	sh.	st.	to.	ca.
cell	.47	.03	.04		.16	.12	.11	.08
bike	.03	.58	.07	.08	.15	.03	.05	
iron	.10		.41	.14	.19	.10	.06	
mouse	.08	.05	.12	.50	.05	.12	.05	.03
shoe	.08		.07	.08	.47	.26	.04	
stapler	.01		.11	.10	.15	.48	.08	.07
toaster	.18		.23	.03	.13		.38	.08
car	.13		.03	.02	.18	.07	.12	.45

Our model: av. accuracy on 8 unseen views = 64.78%

	ce.	bi.	ir.	mo.	sh.	st.	to.	ca.
cell	.61	.03	.08		.09	.08	.08	.05
bike	.02	.76	.08		.10		.02	.02
iron	.08		.58	.08	.08	.06	.06	.08
mouse	.04	.04	.11	.68	.02	.02	.02	.09
shoe	.05		.04	.07	.57	.16	.03	.08
stapler	.01		.08	.04	.06	.69	.04	.07
toaster	.05		.16		.02	.05	.58	.14
car	.03		.07	.02	.09	.03	.05	.71

Fig. 7. Left: Performances of our model (red) and Savarese et al. model [1] (blue) as a function of the number of views used in training. Note that the performances shown here are testing performances, obtained by an average over all 24 testing poses. **Middle:** Confusion table results obtained by the Savarese et al. model [1] for 8 object classes on a sample of 8 unseen views only (dataset [33]). **Right:** Confusion table results obtained by our model under the same conditions.



Savarese et al.'07: aver. acc. = 60.5%

	ce.	bi.	ir.	mo.	sh.	st.	to.	ca.
cell	.69		.06	.09	.03	.09	.03	
bike	.02	.75	.05	.02	.07		.05	.05
iron	.02		.59	.10	.05	.07	.12	.05
mouse	.02		.08	.65	.06	.04	.04	.10
shoe	.08		.10	.08	.47	.10	.04	.14
stapler	.02		.11	.04	.06	.57	.06	.13
toaster	.03		.06	.03	.09	.15	.52	.12
car	.02	.04	.04	.06	.10	.06	.08	.61

Our model: aver. acc. = 72.3%

	ce.	bi.	ir.	mo.	sh.	st.	to.	ca.
	.75		.03	.06		.09	.06	
	.02	.82	.05	.02	.07			.02
	.02		.71	.07	.05	.05	.10	
	.02		.08	.73	.06			.10
	.06		.04	.06	.64	.08	.04	.08
			.09	.02	.06	.70	.02	.11
	.03		.06	.03	.06	.09	.67	.06
		.02	.04	.06	.02	.06	.04	.76

Fig. 8. Left: Confusion table results obtained by [1] for 8 object classes (dataset [35]). **Middle:** Confusion table results obtained by our model under the same conditions. **Right:** Performance improvement achieved by our model over [1] for each category.

reduced set is obtained by randomly removing poses from the original training set. This was done by making sure that no more than one view is removed from any quadruplet of adjacent poses in the viewing sphere[1]. The number of poses used in testing is kept constant (to be more specific, all 24 views are used in this case). This means some of the views in testing have not been presented during training. Fig. 7 illustrates the performances of the two models as a function of the number of views used in training. The plots shows that our model systematically outperforms that of [1]. However, notice that the added accuracy becomes negligible as the number of views in training approaches 24. In other words, when no views are missing in training, the performance of the model used in [1] approximates our model. For a baseline comparison with a pure bag-of-world model the reader can refer to [1]. Fig. 7(middle, right) compare the confusion table results obtained by our model and that of [1] for 8 object classes on a sample of 8 unseen views only.

[1] We have found experimentally that this condition is required to guarantee there are sufficient views for successfully constructing the linkage structure for each class.

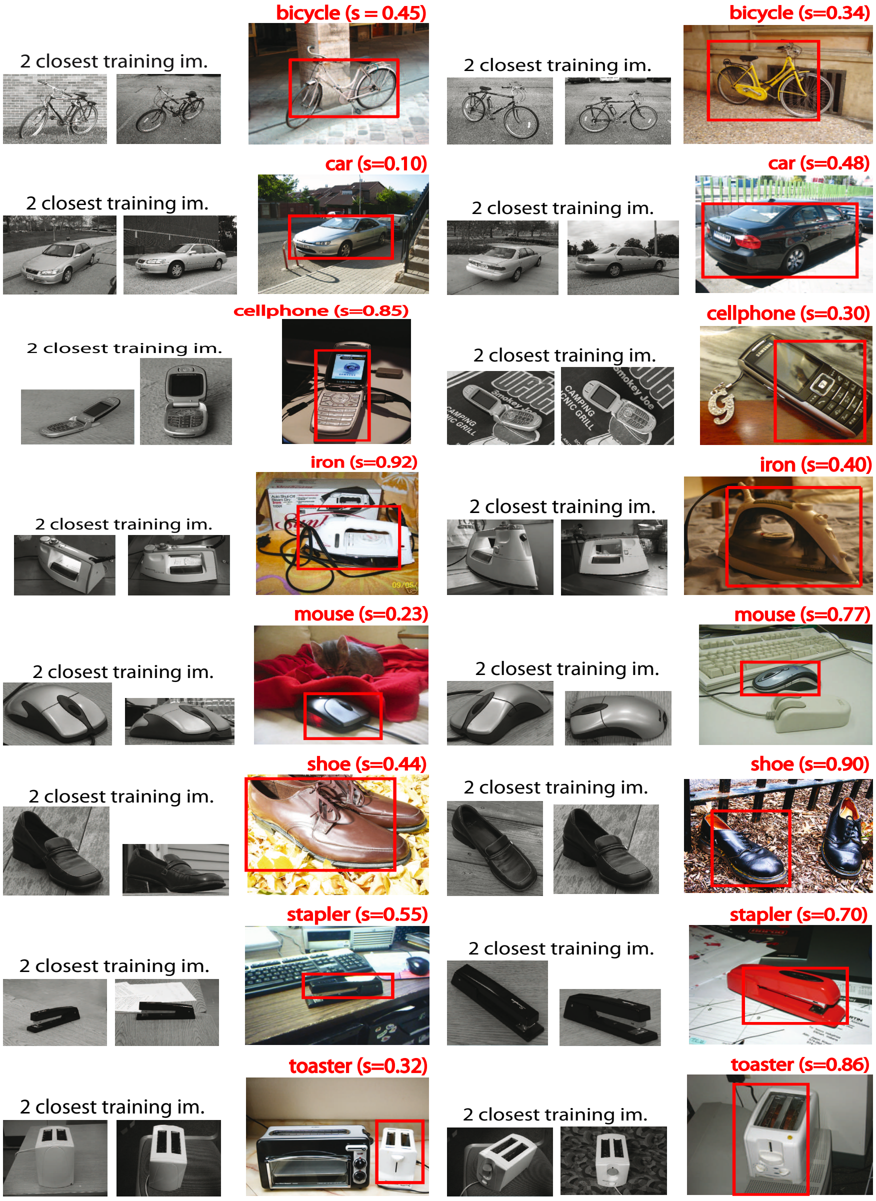


Fig. 9. Estimated pose for each object that was correctly classified by our algorithm. Each row shows two test examples (the colored images in column 3 and column 6) from the same object category. For each test image, we report the estimated location of the object (red bounding box) and the estimated view-synthesis parameter s. s gives an estimate of the pose as it describes the interpolating factor between the two closest model (canonical) views selected by our recognition algorithm. For visualization purposes we illustrate these model views by showing the corresponding training images (columns 1-2 and 4-5). (This figure is best viewed in color with PDF magnification.)

4.2 Experiment II: A New Testing Dataset

In this experiment we test our algorithm on a much more challenging testset. We have collected this testset for two reasons. First, we would like to test our models for recognizing object under arbitrary poses. Second, the dataset in [1] has been collected under relatively controlled settings. While training and testing images are well separated, the background, lighting and the cameras used for this dataset are similar. In this new dataset of 8 object classes, 7 classes of images (cellphone, bike, iron, shoe, stapler, mouse, and toaster) are collected from the Internet (mostly Google and Flickr) by using an automatic image crawler. The initial images are then filtered to remove outliers by a paid undergraduate with no knowledge of our work. We eventually obtain a set of 60 images for each category. The 8^{th} class, the car, is from the LabelMe dataset [34]. A sample of the dataset is available at [35]. As in the previous experiment, we compare the performances of our algorithm to [1]. This time we have trained the models by using the full dataset from Savarese et al. [33] (48 available poses, 10 instances, for a total number of 480 images per category).

Results by both models are reported in Fig. 8. Again, our model achieves better overall results. Fig. 8 (right panel) shows the performance comparison broken down by each category. Notice that for some categories such as cellphone or bikes, the increment is less significant. This suggests our algorithm is more effective for categories composed by a richer 3D structure (as opposed to cellphone and bike that are almost planar objects). All the experiments presented in this section use the 2-view synthesis scheme. The 3-view scheme is currently tested and will be presented in future work. Fig. 9 illustrates a range of pose estimation results on the new dataset. See Fig. 9 caption for details.

5 Conclusion

Recognizing objects in 3D space is an important problem in computer vision. Many works recently have been devoted to this problem. But beyond the possibility of semantic labeling of objects seen under specific views, it is often crucial to recognize the pose of the objects in the 3D space, along with its categorical identity. In this paper, we have proposed an algorithm to deal with the unseen (and/or untrained) poses in recognition. We achieve this by modifying the model proposed by Savarese et al. [1] and by taking advantage of a variant of the view morphing technique proposed by Seitz & Dyer [2]. Our initial testing of the algorithm shows promising results. But a number of issues remain. Our algorithm still requires a good number of views to be used during training in order to generalize. More analysis and research need to be done to make this as minimal as possible. Further research is also needed to explore to what degree the inherent *nuisances* in category-level recognition (lighting variability, occlusions and background clutter) affect the view morphing formulation. Finally, it would be interesting to extend our framework and incorporate the ability to model non-rigid objects.

References

1. Savarese, S., Fei-Fei, L.: 3D generic object categorization, localization and pose estimation. In: IEEE Int. Conf. on Computer Vision, Rio de Janeiro, Brazil (October 2007)
2. Seitz, S., Dyer, C.: View morphing. In: SIGGRAPH, pp. 21–30 (1996)
3. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proc. Computer Vision and Pattern Recognition (2001)
4. Weber, M., Welling, M., Perona, P.: Unsupervised learning of models for recognition. In: Vernon, D. (ed.) ECCV 2000. LNCS, vol. 1842, pp. 18–32. Springer, Heidelberg (2000)
5. Dance, C., Willamowski, J., Fan, L., Bray, C., Csurka, G.: Visual categorization with bags of keypoints. In: ECCV International Workshop on Statistical Learning in Computer Vision, Prague (2004)
6. Fergus, R., Perona, P., Zisserman, A.: Object class recognition by unsupervised scale-invariant learning. In: Proc. Comp. Vis. and Pattern Recogn. (2003)
7. Grauman, K., Darrell, T.: The pyramid match kernel: Discriminative classification with sets of image features. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), Beijing, China (2005)
8. Leibe, B., Schiele, B.: Combined object categorization and segmentation with an implicit shape model. In: Proc. Workshop on satistical learning in computer vision, Prague, Czech Republic (2004)
9. Berg, A., Berg, T., Malik, J.: Shape matching and object recognition using low distortion correspondences. In: Proc. Computer Vis. and Pattern Recog. (2005)
10. Todorovic, S., Ahuja, N.: Extracting subimages of an unknown category from a set of images. In: CVPR (2006)
11. Schneiderman, H., Kanade, T.: A statistical approach to 3D object detection applied to faces and cars. In: Proc. CVPR, pp. 746–751 (2000)
12. Weber, M., Einhaeuser, W., Welling, M., Perona, P.: Viewpoint-invariant learning and detection of human heads. In: Int. Conf. Autom. Face and Gesture Rec. (2000)
13. Torralba, A., Murphy, K., Freeman, W.: Sharing features: efficient boosting procedures for multiclass object detection. In: Proc. Conference on Computer Vision and Pattern Recognition (CVPR) (2004)
14. Beier, T., Neely, S.: Feature-based image metamorphosis. In: SIGGRAPH (1992)
15. Chen, S., Williams, L.: View interpolation for image synthesis. Computer Graphics 27, 279–288 (1993)
16. Szeliski, R.: Video mosaics for virtual environments. Computer Graphics and Applications 16, 22–30 (1996)
17. Avidan, S., Shashua, A.: Novel view synthesis in tensor space. In: Proc. Computer Vision and Pattern Recognition, vol. 1, pp. 1034–1040 (1997)
18. Laveau, S., Faugeras, O.: 3-d scene representation as a collection of images. In: Proc. International Conference on Pattern Recognition (1994)
19. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press, Cambridge (2004)
20. Xiao, J., Shah, M.: Tri-view morphing. CVIU 96 (2004)
21. Brown, M., Lowe, D.: Unsupervised 3D object recognition and reconstruction in unordered datasets. In: 5th International Conference on 3D Imaging and Modelling (3DIM 2005), Ottawa, Canada (2005)
22. Lowe, D.: Object recognition from local scale-invariant features. In: Proc. International Conference on Computer Vision, pp. 1150–1157 (1999)

23. Ullman, S., Basri, R.: Recognition by linear combination of models. Technical report, Cambridge, MA, USA (1989)
24. Rothganger, F., Lazebnik, S., Schmid, C., Ponce, J.: 3d object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints. IJCV 66(3), 231–259 (2006)
25. Ferrari, V., Tuytelaars, T., Van Gool, L.: Simultaneous object recognition and segmentation from single or multiple model views. IJCV (2006)
26. Lazebnik, S., Schmid, C., Ponce, J.: Semi-local affine parts for object recognition. In: Proceedings of BMVC, Kingston, UK, vol. 2, pp. 959–968 (2004)
27. Bart, E., Byvatov, E., Ullman, S.: View-invariant recognition using corresponding object fragments. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3022, pp. 152–165. Springer, Heidelberg (2004)
28. Thomas, A., Ferrari, V., Leibe, B., Tuytelaars, T., Schiele, B., Van Gool, L.: Towards multi-view object class detection. In: Proceedings IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 1589–1596 (2006)
29. Kushal, A., Schmid, C., Ponce, J.: Flexible object models for category-level 3d object recognition. In: Proc. Conf. on Comp. Vis. and Patt. Recogn. (2007)
30. Hoeim, D., Rother, C., Winn, J.: 3D layoutcrf for multi-view object class recognition and segmentation. In: Proc. In IEEE Conference on Computer Vision and Pattern Recognition (2007)
31. Yan, P., Khan, D., Shah, M.: 3d model based object class detection in an arbitrary view. In: ICCV (2007)
32. Chiu, H., Kaelbling, L., Lozano-Perez, T.: Virtual training for multi-view object class recognition. In: CVPR (2007)
33. `http://vangogh.ai.uiuc.edu/silvio/3ddataset.html`
34. Russell, B., Torralba, A., Murphy, K., Freeman, W.: Labelme: a database and web-based tool for image annotation. Int. Journal of Computer Vision (2007)
35. `http://vangogh.ai.uiuc.edu/silvio/3ddataset2.html`