Chapter 8
Multi-view Object Categorization and Pose Estimation

Silvio Savarese and Li Fei-Fei

Abstract. Object and scene categorization has been a central topic of computer vision research in recent years. The problem is a highly challenging one. A single object may show tremendous variability in appearance and structure under various photometric and geometric conditions. In addition, members of the same class may differ from each other due to various degrees of intra-class variability. Recently, researchers have proposed new models towards the goal of: i) finding a suitable representation that can efficiently capture the intrinsic three-dimensional and multi-view nature of object categories; ii) taking advantage of this representation to help the recognition and categorization task. In this Chapter we will review recent approaches aimed at tackling this challenging problem and focus on the work by Savarese & Fei-Fei [54, 55]. In [54, 55] multi-view object models are obtained by linking together diagnostic parts of the objects from different viewing point. Instead of recovering a full 3D geometry, parts are connected through their mutual homographic transformation. The resulting model is a compact summarization of both the appearance and geometry information of the object class. We show that such a model can be learnt via minimal supervision compared to competitive techniques. The model can be used to detect objects under arbitrary and/or unseen poses by means of a two-step algorithm. This algorithm, inspired by works in single object view synthesis (e.g., Seitz & Dyer [57]), has the ability to synthesize object appearance and shape properties at recognition time, and in turn estimate the object pose that best matches the observations. We conclude this Chapter by presenting experiments on detection, recognition and pose estimation results with respect to two datasets in [54,55] as well as to PASCAL Visual Object Classes (VOC) dataset [15]. Experiments indicate that representation and algorithms presented in [54,55] can be successfully employed in a number of generic object recognition tasks.

Silvio Savarese
University of Michigan at Ann Arbor, Department of Electical Engineering, USA
e-mail: silvio@umich.edu

Li Fei-Fei
Stanford University, Department of Computer Science USA
e-mail: feifeili@cs.stanford.edu

R. Cipolla, S. Battiato, & G.M. Farinella (Eds.): Computer Vision, SCI 285, pp. 205–231.
springerlink.com

Fig. 8.1 Categorize an Object Given An Unseen View. azimuth: [front,right,back,left]= $[0,90,180,270]^o$; zenith: [low, med., high]= $[0,45,90]^o$

8.1 Introduction

The ability to interpret a scene, recognize the objects within, estimate their location and pose is crucial for a robust, intelligent visual recognition system. In robotic manipulation, a robotic arm may need to detect and grasps objects in the scene such as a cup or book; in autonomous navigation, an unmanned vehicle may need to recognize and interpret the behavior of pedestrians and other vehicles in the environment. Critically, accurate pose recovery is not only important if one wants to interact with the objects in the environment (if a robotic arms wishes to grasp a mug, the system must estimate mug's pose with high degree of accuracy); it is also a key ingredient that enables the visual system to perform higher level tasks such activity or action recognition. Despite recent successful efforts in the vision community toward the goal of designing systems for object recognition, a number of challenges still remain: not only does one need to cope with traditional nuisances in object categorization problems (objects appearance variability due to intra-class changes, occlusions and lighting conditions), but also to handle view-point variability and propose representations that capture the intrinsic multi-view nature of object categories.

In this Chapter we describe a recent recognition paradigm for discovering object semantics under arbitrary viewing conditions as well as recovering the basic geometrical attributes of object categories and their relationships with the observer. Figure 8.1 illustrates more precisely the problem we would like to solve. Given an image containing some object(s), we would like to learn object category models that allow us to: i) detect and categorize the object as a car (or a stapler, or a computer mouse), and ii) estimate the pose (or view point) 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 Figure 8.2).

Most of the recent advances in object categorization have focused on modeling the appearance and shape variability of objects viewed from a limited number of poses [66, 19, 18, 35, 17, 23], or a mixture of poses [56, 67, 48, 72]. In these methods, different poses of the same object category result in completely independent models, wherein neither features nor parts are shared across views. These methods typically ignore the problem of recovering the object pose altogether. We refer to such models

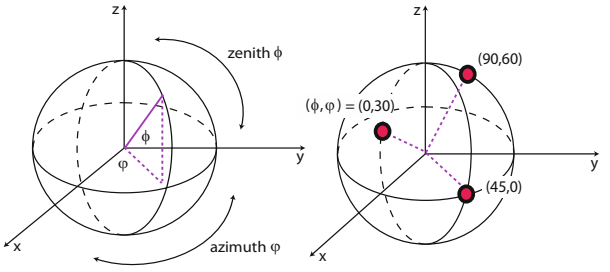

Fig. 8.2 Left: An object pose is represented by a pair of azimuth and zenith angles. Right: Some of the unseen poses tested during our recognition experiments (Figure 8.12).

as *single-view 2D models*. At the opposite end of the spectrum, several works have addressed the issue of single object recognition by modeling different degree of 3D information [40, 50, 6, 20]. Since these methods achieve recognition by matching local features under rigid geometrical transformations, they are successful in recovering the object pose, but they are difficult to extend to handle object classes. We refer to such models as *single instance 3D models*. Similar limitations are suffered by representations based on aspect graphs [31, 32, 59, 58, 13, 47, 5, 12, 10]. A small but growing number of recent studies have begun to address the problem of object classification in a true multi-view setting [63, 33, 27, 70, 9, 54, 55, 37, 49, 71, 2, 16, 61, 60, 43]. Since in these methods object elements (features, parts, contours) are connected across views so as to form an unique and coherent model for the object category (e.g., Figure 8.4), we refer to such models as *multi-view models*. These techniques bridge the gap between single view 2D models and single instance 3D object models. In this book Chapter we will focus on the framework introduced by [54, 55], which represents one of the first attempts to model the multi-view nature of 3D object categories. In particular we will discuss some of the critical contributions of such multi-view model [54, 55]:

- A part-based 3D model of an object class is proposed by encoding both the appearance and 3D geometric shape information (see Figure 8.3). Stable parts of objects from one class are linked together to capture both the appearance and shape properties of the object class. This model produces a compact yet power representation of an object class, differing from most of the previous works which store various image exemplars or model exemplars of different viewing angles.
- Toward the goal of learning the multi-view model of an object class, the algorithm demands less supervision in the learning process compared to previous works (i.e., [63]). The method is designed to handle either segmented or unsegmented objects in training. Most importantly, the method does not require view point labels or to have training images sorted in any particular order. A key assumption, however, is that multiple views of the same object instance are assumed to be available.

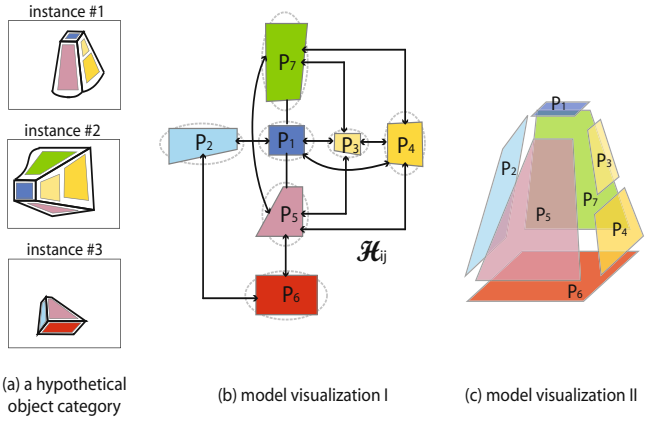

Fig. 8.3 Schematic illustration of the 3D object category model. (a) We illustrate our model by using a hypothetical 3D object category. Three instances of the hypothetical objects are shown here as sample training images. The colored regions of each instance are "canonical parts" of the objects that will be put together to form the final model. These "canonical parts" are made of patches usually provided by feature detectors (see also Figure 8.4). When parts across different instances share the same color code, it indicates that the model has learnt the correspondences among these regions based on the appearance and geometric consistency. (b) The final model of the 3D object category can be viewed as a connected graph of the object canonical parts (colored regions, P_i), canonical part relations ($\mathcal{H}$), and the encoded variabilities of the appearances and geometric structure (visualized by the dashed ellipses surrounding each part). (c) A more intuitive visualization of the model puts together the canonical parts in a 3D graph based of the learned geometric relations ($\mathcal{H}$). This figure is best viewed under color.

- The algorithm has the ability to represent and synthesize views of object classes that are not present in training. The view-synthesis approach is inspired by previous research on view morphing and image synthesis from multiple views. The main contribution of [54, 55] is that the synthesis takes place at the categorical level as opposed to the single object level (as previously explored).
- Given a novel testing image containing an object class, not only does the algorithm classifies the object, but it also infers the pose and scale and localizes the object in the image. Furthermore, the algorithm takes advantage of our view-synthesis machinery for recognizing objects seen under arbitrary views. As opposed to [9] where training views are augmented by using synthetic data, we synthesize the views at *recognition* time.
- Extensive experimental validation is provided. Competitive categorization, localization and pose estimation performances are observed with respect to the dataset in [63] as well as the challenging 3D object datasets introduced in [54, 55].

8.2 Literature Review

Interest in solving the 3D/multi-view object recognition as well as pose estimation problem starts with a number of seminal work [3,24,39,41,46,51,65] in the 80s and early 90s, which form the foundation of modern object recognition. Beginning from the late 90s, researchers start proposing a new generation of techniques for solving single object recognition using *single instance 3D models*. In [38, 42, 44], objects are represented by highly discriminative and local invariant features related by local geometric constrains. Methods by [50, 6, 20] follow the idea of enforcing global geometric constraints and/or achieve 3D affine or Euclidean object reconstruction from multiple (often) unregistered views for recognizing single objects in cluttered scenes. These methods are successful thanks to their ability to identify strong geometrical constraints and highly discriminative 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. Another large body of literature on object recognition introduces the concept of *Aspect Graph* (AG). AGs represent 3D objects as a set of topologically distinct views based on visibility constraints. Starting from seminal works of [31,32], different AG representations are introduced during the 80s and 90s [59,58,13,47,5,14] until recent extensions [12,10]. Similarly to single instance 3D models, AG methods lack of generalization power in representing object categories, and have shown limited success in modeling intra-class variability. Also, most of AGs poorly handle nuisances such as occlusions and background clutter. A natural step forward to the categorization problem is offered by *3D object category classification methods* such as [52,28,26,22,30,62,36]. These methods focus on classifying objects that are expressed as collections of 3D points, 3D meshes or 3D synthetic cad models. Often 3D shape databases are used [1]. Due to their limited ability to coherently integrate real-world object albedo information with the underlying 3D structure, these methods are hardly used to identify real world objects in cluttered scenes from still images or videos. A recent survey summarizes relevant literature [62]. Partially to accommodate intra-class variability, researchers have proposed to leverage on the large literature on single 2D view object categorization and represent 3D object categories as a *mixture of 2D single view object category models* [56,67,7,64,4]. In mixture models, single view object models are completely independent, wherein neither features or parts are linked across views. An exception is the work by [64] where an efficient multi-class boosting procedure is introduced to limit the computational overload. The consequence is that mixture models fail to form a coherent and compact multi-view representation of an object category. Methods based on mixture models are costly to train and prone to false alarm, if several views need to be encoded. Finally, only few methods [67] have attempted to solve the pose estimation problem as we will discuss later in more details.

Recently, a new class of methods have tried to bridge the gap between single view 2D models and single instance 3D object models, and have begun to address the problem of object classification in a true multi-view setting (*multi-view models*). In these methods, object elements (features, parts, contours) are connected

across views so as to form an unique and coherent model for the object category (e.g., Figure 8.4). Pioneering multi-view techniques are introduced in the specific domain of face detection [48, 72]. Methods in [63, 33] extend single view models into the multi-view setting by linking relevant features across views. Alternative techniques [27, 70, 37, 69] represent an object category by using synthetic or reconstructed 3D models on top of which the typical distribution of appearance element is learned. Authors in [9] build an object representation upon a 3D skeleton model of predefined parts from 2D images. Very recent examples of multi-view representations are [61, 60, 49, 71, 2, 16]. The work by [54, 55] proposes a new representation where object categories are modeled as a collection of view point invariant parts connected by relative view point transformations. [54, 55] stand among the pioneering contributions on multi-view representation and are among the first methods that have addressed the issue of view point estimation for generic object categories. In the remainder of this book Chapter we discuss in details the representation introduced in [54, 55]. In Section 8.3.2 we explain the multi-view model based on canonical parts and linkage structure. Then we describe how to learn such multi-view model in Section 8.4. In Section 8.5 we present the machinery for synthesizing novel views in the viewing sphere. In Section 8.6 we demonstrate how a novel instance of an object category is recognized, localized and its pose inferred. Finally, we show experimental results in Section 8.7.

8.3 The Part-Based Multi-view Model

8.3.1 Overview

In [54, 55] models of an object category are obtained by linking together diagnostic parts (also called canonical parts) of the objects from different viewing points. As previous research has demonstrated, a part-based representation [34] is more stable for capturing appearance variability of object categories across instances and views. Canonical parts are discriminative and view invariant representations of local planar regions attached to the object physical surface. Such parts are modeled by distributions of vector quantized features [11]. Instead of expressing part relationships by recovering the full 3D geometry of the object, [50, 6, 20], canonical parts are connected through their mutual homographic transformations and positions. The resulting model is a compact summarization of both the appearance and geometrical information of the object categories across views (rather than being just a collection of single view models). Effectively, the linkage structure can be interpreted as the generalization to the multi-view case of single 2D constellation or pictorial structure models [68, 19, 18]) where parts or features are connected by a mere 2D translational relationship. [54, 55]'s framework requires less supervision than competing techniques ([63, 33, 27, 70, 37, 69]) (where often pose labels are required). Similarly to other constellation methods, [54, 55]'s model enables a recognition system that is robust to occlusions and background clutter. Finally, and most importantly, by introducing the view morphing constraints, [55] has demonstrated the ability to

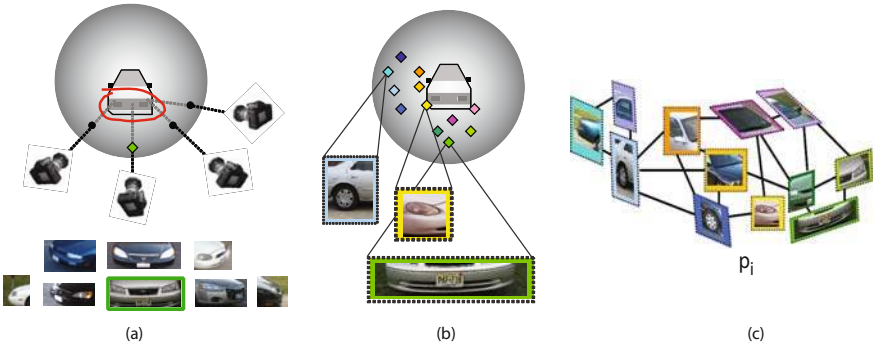

Fig. 8.4 Canonical parts and linkage structure. (a): A car within the viewing sphere. As the observer moves on the viewing sphere the same part produces different appearances. The location on the viewing sphere where the part is viewed the most frontally gives rise to a canonical part. The appearance of such canonical part is highlighted in green. (b): Colored markers indicate locations of other canonical parts. (c): Canonical parts are connected together in a linkage structure (see also Figure 8.3). 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 dashed rectangle.

predict appearance and location of parts that are not necessarily canonical. This is useful for recognizing objects observed from arbitrary viewing conditions (that is, from views that are not seen in learning) and critical for improving the false alarm rate (a consequence of single view object representations). [54, 55]'s framework for recognizing poses of generic object categories is among the earliest attempts of this kind (along with [48, 72, 67]).

8.3.2 Canonical Parts and Linkage Structure

In this Section we describe in details the concept of canonical parts and linkage structures. The central ideas are summarized in Figure 8.3 and Figure 8.4. They offer a schematic view of the core components of the model through a hypothetical 3D object category.

The **appearance** information is captured in the diagnostic parts of the objects in one class, denoted as P_i. Each "part" is a region of an object that tends to appear consistently throughout different instances of the same category (shown in colored patches in Figure 8.3(a)). It is a collection of a number of smaller image patches usually provided by the feature detectors, constraint by some geometric consistency.

Readers familiar with the current object recognition literature are reminded that our "part" is not a single detected region such as Harris corner or DoG detection, but rather a larger structure that contains many detected local regions. Given P_i, our model also encodes the appearance variations observed in training in the form of distributions of descriptors. In our model, we call such diagnostic parts *canonical parts* as they are representative of parts viewed in their most frontal position. For example, the canonical part representation of the car rear bumper is the one that is viewed the most frontally (Figure8.3(b) and 8.4(a)).

Given an assortment of canonical parts (e.g., the colored patches in Figure 8.4(b)), a *linkage structure* connects each pair of canonical parts $\{P_j, P_i\}$ if they can be both visible at the same time (Figure 8.3(b) and 8.4(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 Figure 8.4(c) by showing a slanted version of each canonical part.

We define a *canonical view* V as the collection of canonical parts that share the same view V (Figure 8.4(c)). Thus, each pair of canonical parts $\{P_i, P_j\}$ within 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 [68, 19]). 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.

8.4 Building the Model

We detail here the algorithm for building a 3D object class model from a set of training images. We assume that each training image contains one instance of the target object class. We do not, however, have information about the instance membership or pose of the object. The task of learning is to start with this set of raw images, extract features to form parts, obtain a set of *canonical parts* and finally form the object class model by connecting these canonical parts across views.

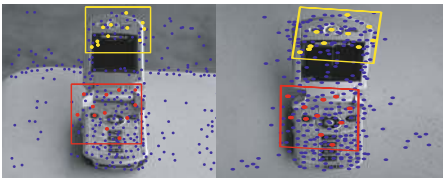

Fig. 8.5 Detected features using the scaled invariant saliency detector [29]. All interest points are indicated by blue dots. The boxed regions in each image denote the learnt parts for this pair. When two parts across images share the same color (i.e., red boxes), they are connected by the algorithm. This figure should be viewed in color.

8.4.1 Extract Features

Local image patches are the basic building blocks of an object image. The algorithm, however, works independently of any particular choice of feature detectors or descriptors [44, 38]). In practice, we choose the Saliency detector [29] and the SIFT descriptor [38] to characterize local features. An image i therefore contains hundreds of detected patches, each represented as $f_i = (\mathbf{a_i}, \mathbf{x_i})$, where $\mathbf{a_i}$ is the appearance of the patch, described by a 128-dimension SIFT vector, and $\mathbf{x_i}$ is the location of the feature on the 2D image. Figure 8.5 shows two examples of cellphone images and their detected patches.

8.4.2 Form Parts

The 3D object category model is represented in a hierarchical way. Local image features are first grouped into larger regions (called "parts"). A selected subset of these parts (according to appearance and geometric consistency) are then linked together as a full 3D model. This choice stems from the observation that larger regions of objects often carry more discriminative information in appearance and are more stable in their geometric relationships with other parts of the object [34].

The goal of this step is to group local image features into "parts" that are consistent in appearance and geometry across images. A global geometrical constraint is obtained by imposing that feature match candidates (belonging to different views) are related by the fundamental matrix $\mathcal{F}$. A local geometrical constraint is enforced by imposing that features belonging to a neighborhood are related by homographic transformation $\mathcal{H}$ induced by $\mathcal{F}$ [25]. We use a scheme based on RANSAC [21] to enforce such constraints while the optimal $\mathcal{F}$ and $\mathcal{H}$ are estimated. Below is a brief sketch of the algorithm.

1. Obtain a set of M candidate features based on appearance similarity measured by $d(\mathbf{a_i} - \mathbf{a_j})$ across 2 training images.
2. Run RANSAC algorithm on M to obtain a new (and smaller) set of matches $M_F \in M$ based on $\mathbf{x_i}\mathcal{F}\mathbf{x_j} = 0$, where $\mathcal{F}$ denotes the fundamental matrix.

3. Further refine the matches using RANSAC to obtain a set of M_H matches such that $\mathbf{x}_i - \mathcal{H}\mathbf{x}_j = 0$, where $M_H \in M_F \in M$.

Step 2 and Step 3 can be iterated until the residual error computed on the inliers stops decreasing. Step 3 returns a pair of local neighborhood regions across the 2 training images in which all features $f_i \in M_H^{(i,j)}$ satisfy a vicinity constraint. We call them a matched "part". We follow this procedure for every pair of training images. Figure 8.5 shows example parts indicated by boxes on these two cellphone images. Note that there is no presumed shape or size of these parts.

Implementation Details. On average parts contain $50-200$ features, sufficient to effectively represent the local structure of the object from a particular view. We obtain on average $700-1000$ matched parts within a training set of 48 images. We use a mask to remove spurious matches coming from the background. This is not a requirement for the algorithm to work. [6] shows that spurious matches can be effectively removed by enforcing global constraints across all the views. Finally, even if matched parts can be obtained from pairs of images belonging to different instances of a given category, we have noticed that the algorithm in 8.4.2 mostly produces matched parts from images belonging to the same object instance. This is due to the inherent lack of flexibility of RANSAC to handle intra-class variability. In fact, this is an advantage because it guarantees robustness and stability in the part matching process. Actual matching of corresponding parts belonging to different object instances is achieved in the optimization process detailed in 8.4.4.

8.4.2.1 Representing Canonical Parts

Each canonical part is represented by a distribution of feature descriptors along with their x,y location within the part. Specifically, we 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 [11] - that is, a normalized histogram h of vector quantized descriptors contained in B. 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., $B = \left[b_1 \ \ldots \ b_4 \right]$; 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.

8.4.3 Find Canonical Parts Candidates

Our goal is to represent the final object category with "canonical parts" and their mutual geometric relations. To do so, we need to first propose a set of canonical part

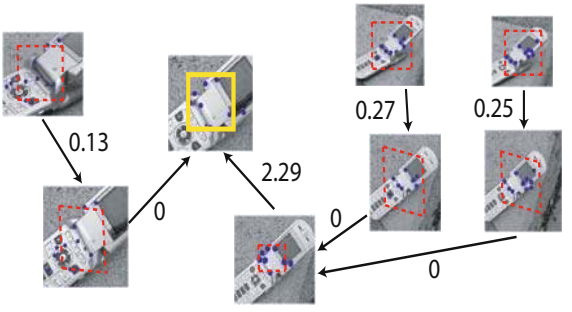

Fig. 8.6 Illustration of linked parts for proposing one canonical part of the cellphone model using directed graph. The boxes indicate parts associated with this canonical part. The blue dots indicate detected local features within the parts. The yellow box is the proposed canonical part by summarizing all *factors of compression* (indicated by the numerical value adjacent to each arrow) given all the connected paths.

candidates based on a view-point criteria. What we have from the training images is a large set of "parts" that are paired across different images, each part consisting of a number of local features. Many of these parts linked across different images correspond to one *actual* part of the object (e.g., LCD screen of a cellphone). Figure 8.6 is a illustration of the connected parts estimated from Step 8.4.2. The most possible front view of an actual object part defines a *canonical part candidate*. This will be by definition the canonical pose attached to the canonical part candidate. A canonical part candidate can be computed from the set of linked parts as follows.

Between every connected pair of parts, we associate them with a *factor of compression* cost $\mathcal{K}_{ij}$. $\mathcal{K}_{ij}$ is a function of $\mathcal{A}_{ij}$ in the homographic relationship $\mathcal{H}_{ij}$ between these two parts. $\mathcal{H}_{ij}$ is provided by the algorithm in section 8.4.2. Specifically, $\mathcal{K}_{ij} = \left(\lambda_1^{ij}\lambda_2^{ij} - 1\right)$, where $\lambda_{1,2}^{ij}$ are the two singular values of $\mathcal{A}_{ij}$. $\mathcal{K}_{ij}$ is greater than 0 when P_i is a less compressed version than P_j under affine transformation. Using the sign of $\mathcal{K}_{ij}$, we assign the direction between two parts. The full set of parts and their directed connections weighted by $\mathcal{K}_{ij}$ form a weighted directed graph (Figure 8.6). It is easy to show that the path associated to highest value of the total factor of compression cost $\left(\sum_{(i,j)\in \text{path}} \mathcal{K}_{ij}\right)$ gives rise to a canonical part candidate for it can be identified as the part P attached to the terminal node of such maximum cost path. The intuition here is that the maximum cost path is the one that leads to the part with smallest compression, thus the canonical one. The maximum cost path can be found with a simple greedy algorithm.

Implementation Details. The graph structure is on average composed of $10-15$ parts but can go as low as 2, if a part is shared by only two views. For that reason, the greedy algorithm finds the optimal solution very quickly. Special care, however, needs to be taken if the graph contains loops. This may occur when the orientation of a part is estimated with low accuracy from the previous step. Typically the number of canonical part candidates is one third of the initial set of part candidates.

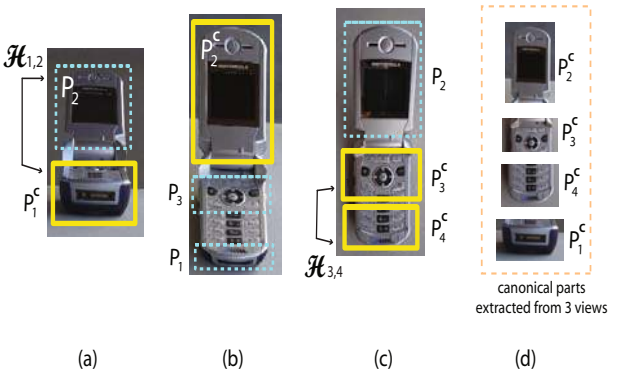

Fig. 8.7 Illustration of the canonical parts and their geometric relations for three views of the same object. The yellow box indicates the canonical part of interest that is viewed given its canonical pose (i.e., most frontal view by definition). The examples of canonical regions extracted from these three views are shown in the box on the right. The dashed cyan boxes indicate parts that do not share the same pose with the yellow canonical part. The cyan parts have a canonical counter part in a different pose. In this example we use the symbol "c" to differentiate a canonical part from its "non-canonical" counterpart. For instance, there exists a linkage structure between canonical parts P_1^c and P_2^c. The $\mathcal{H}_{12}$ denotes the transformation to observe P_2^c when P_1^c is viewed in its canonical position (thus, generating cyan P_2). In the right most pose, two canonical parts P_3^c and P_4^c share the same canonical pose. In this case, the transformation $\mathcal{H}_{34}$ is just a translation because P_3^c and P_4^c are canonical at the same time.

8.4.4 Create the Model

Section 8.4.3 has proposed a number of canonical part candidates from the training images. So far, we have only utilized local appearance or pair-wise geometry information to find correspondences between parts and find the canonical part candidates. Now we are ready to take all these candidates to obtain a canonical part at the categorical level. This allows propose a 3D object category model by finding a globally consistent and optimal combination of canonical parts.

We use the same notation (P_i) to indicate a canonical part of a given category. The context can help differentiate the categorical case from the single instance case. As anticipated in Section 8.3.2, given two different canonical part P_i and P_j, there are two ways that they are placed with respect to each other onto the 3D object model. In the first case, when P_i is viewed frontally, P_j is also viewed frontally (Figure 8.7, right panel). In this case the homographic linkage between these two canonical parts is $\mathcal{H}_{ij} = \begin{bmatrix} \mathbf{I} & \mathbf{t}_{ij} \\ \mathbf{0} & 1 \end{bmatrix}$, where $\mathbf{I}$ is the identity matrix. In the second case, P_i and P_j are not viewed frontally simultaneously. They are, therefore, related by a full homographic $\mathcal{H}_{ij} = \begin{bmatrix} \mathcal{A}_{ij} & \mathbf{t}_{ij} \\ \mathbf{0} & 1 \end{bmatrix}$. $\mathcal{H}_{ij}$ denotes the transformation to observe P_j when P_i is viewed in its most front view position. Parts P_1 when P_2 in Figure 8.7 have this type of linkage.

$\mathcal{A}_{ij}$ captures both the 2D relationship (e.g., position) between canonical parts as well as a soft 3D relationship which provided by the affinity transformation $\mathcal{A}_{ij}$ between parts. Canonical parts that are not connected correspond to sides of the object that can never be seen at the same time. As introduced in Section 8.3.2, we define a *canonical view* V as the collection of canonical parts that share the same view V (Figure 8.4(c)). Thus, each pair of canonical parts $\{P_i, P_j\}$ within V is connected by $\mathcal{A}_{ij} = \mathbf{I}$ and a translation vector $\mathbf{t}_{ij}$.

Given the pool of candidate canonical parts from all the instances of a given category, we wish to calculate the set of canonical parts at the categorical level. This can be done by matching corresponding candidate canonical parts across all the instances. This correspondence problem can be solved by means of an optimization process that jointly minimizes the appearance difference between matching candidates and their corresponding linkage structure $\mathcal{A}_{ij}$.

The 1st row of Figure 8.14 (2^{nd} and 3^{rd} columns) shows an illustration of the learnt cellphone model. The model obtained thus far provides a compact representation of object parts from all the views.

Implementation Details. The optimization is carried out by exploiting similarity of appearance and the estimated linkage structure between canonical part candidates belonging to different object instances. The appearance similarity is computed as a chi-square distance between the histograms representing the canonical region appearances. Similarity of linkage structure is computed by comparing $\mathcal{A}_{ij}$ for every pairs of canonical parts candidates P_i, P_j. Notice that this optimization step greatly benefits from the fact that parts-to-be-matched are canonical. This means that all the parts are already normalized in term of their viewing angle and scale. Furthermore, the number of canonical part candidates is a small subset the initial number of parts. All this greatly simplifies the matching process which could have been hardly feasible otherwise.

8.5 View Synthesis

8.5.1 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 Section 8.6, this ability becomes crucial if we want to recognize an object category seen under an arbitrary pose. The 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.

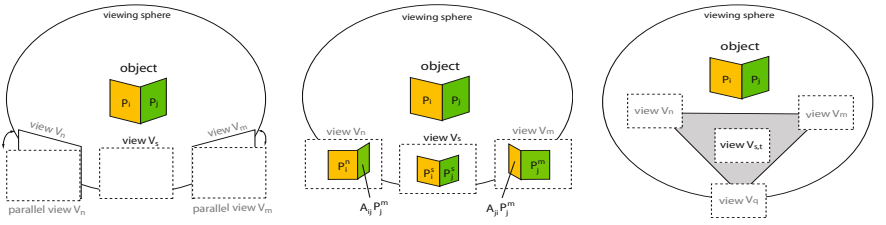

Fig. 8.8 View Synthesis. Left: If the views are in a neighborhood on the viewing sphere, the cameras can be approximated as being *parallel*, enabling a linear interpolation scheme. Middle: 2-view synthesis: A pair of linked parts $\{P_i^s, P_j^s\} \in V^s$ is synthesized from the pair $P_i^n \in V^n$, 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.

8.5.1.1 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) [8]. Moreover, Seitz & Dyer [57] 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 (Figure8.8). 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.

8.5.1.2 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 $\mathcal{A}_{ij} \neq I$ (Figure8.8). 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:

$$B_i^s = (1-s)B_i^n + s\mathcal{A}_{ij}B_i^n; \quad B_j^s = sB_j^m + (1-s)\mathcal{A}_{ji}B_j^m; \tag{8.1}$$

$$h_i^s = (1-s)h_i^n + sh_i^m; \quad h_j^s = sh_j^n + (1-s)h_j^m; \tag{8.2}$$

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:

$$\mathbf{t}_{ij}^s = (1-s)\mathbf{t}_{ij}^n + s\mathbf{t}_{ij}^m \tag{8.3}$$

In summary, Equation 8.1 and 8.3 regulate the synthesis of the linkage structure between the pair $\{P_i^s, P_j^s\}$; whereas Equation 8.2 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.

8.5.1.3 Three-View Synthesis

One limitation of the interpolation scheme described in Section 8.5.1.2 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 (Figure8.8) 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:

$$B_i^{st} = [\,(s-1)I \;\; sI\,] \begin{pmatrix} B_i^n & \mathcal{A}_{ik}B_i^n \\ \mathcal{A}_{ij}B_i^n & \mathcal{A}_{ik}\mathcal{A}_{ij}B_i^n \end{pmatrix} \begin{bmatrix} (1-t)I \\ tI \end{bmatrix} \tag{8.4}$$

$$h_i^{st} = [\,(s-1)I \;\; sI\,] \begin{pmatrix} h_i^n & h_i^q \\ h_i^m & h_i^p \end{pmatrix} \begin{bmatrix} (1-t)I \\ tI \end{bmatrix} \tag{8.5}$$

$$\mathbf{t}_{ij}^{st} = [\,(s-1)I \;\; sI\,] \begin{pmatrix} \mathbf{t}_{ij}^n & \mathbf{t}_{ik}^q \\ \mathbf{t}_{ij}^m & \mathbf{t}_{ij}^m + \mathbf{t}_{ik}^q - \mathbf{t}_{ij}^n \end{pmatrix} \begin{bmatrix} (1-t)I \\ tI \end{bmatrix} \tag{8.6}$$

Analogous equations can be written for the remaining indexes.

8.6 Recognizing Object Class in Unseen Views

Section 8.5.1 has outlined all the critical ingredients of the model for representing and synthesizing new views. We discuss here an algorithm for recognizing pose and categorical membership of a query object seen under arbitrary view point. We consider a two-step recognition procedure. Similarly to the training procedure, we first extract image features and use these to propose candidate canonical parts. This provides the input for the first step of the algorithm whose output is a short list of

Algorithm step 1

1. $I \leftarrow$ list of parts extracted from test image
2. for each model C
3. for each canonical view $V \in C$
4. $[R(n), V^*(n)] \leftarrow \text{MatchView}(V, C, I)$; % return similarity R
5. n ++;
6. $L \leftarrow \text{KMinIndex}(R)$ % return shortlist L

MatchView(V, C, I)
1. for each canonical part $P \in V$
2. $M(p) \leftarrow$ MatchKPart (P, I)); % return K best matches
3. p ++;
4. for each canonical part $\bar{P} \in C$ linked to V
5. $\bar{M}(q) \leftarrow$ MatchKPart $(\bar{P}, I)$; % return K best matches
6. q ++;
7. $[M^*, \bar{M}^*] \leftarrow \text{Optimize}(V, M, \bar{M})$;
8. $V^* \leftarrow \text{GenerateTestView}(M^*, \bar{M}^*, I)$;
9. $R \leftarrow \text{Distance}(V, V^*)$;
10. Return R, V^*;

Fig. 8.9 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.

the K best model views across all views and all categories. The second step refines the error scores of the short list by using the view-synthesis scheme.

8.6.1 Extract Features and Get Part Candidates

We follow the same procedure as in learning to find local interest points by using the Saliency detector [29]. Each detected patch is then characterized by a 128-dimension SIFT vector [38]. Given an object model, say the "cellphone" model,

Algorithm step 2

1. for each canonical view $V \in L$
2. $V^* \leftarrow L(l)$
3. $V' \leftarrow \text{FindClosestView}(V, C)$;
4. $V'' \leftarrow \text{FindSecondClosestView}(V, C)$;
5. for each 2-view synthesis parameter s
6. $V^s \leftarrow \text{2-ViewSynthesis}(V, V', s)$;
7. $R(s) \leftarrow \text{Distance}(V^s, V^*)$;
8. for each 3-view synthesis parameters s and t
9. $V^{s,t} \leftarrow \text{3-ViewSynthesis}(V, V', V'', s, t)$;
10. $R(s,t) \leftarrow \text{Distance}(V^{s,t}, V^*)$;
11. $L(l) \leftarrow \text{Min}(R)$;
12. l ++;
13. $[C_w V_w] \leftarrow \text{MinIndex}(L)$;

Fig. 8.10 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.

we first find a list of canonical part candidates by the following procedure. For each canonical part of the model, we greedily search through the test image by a scanning window across pixel locations, scales and orientations. Canonical parts and test parts are matched by comparing the distributions of features belonging to the relevant regions. The most probably N firings (typically 5) are retained as the N candidates for a canonical part P_i. This provides hypotheses of canonical parts consistent with a certain canonical view of an object model.

8.6.2 Recognition Procedure: First Step

In the first step (Figure 8.9), we want to match the query image with the best object class model and pose. Given hypotheses of canonical parts consistent with a certain canonical view of an object model, 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 optimization process is very similar to the one introduced in learning when different constellations of canonical parts are matched. 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.

8.6.3 Recognition Procedure: Second Step

In the second step (Figure 8.10), we use the view synthesis scheme (Section 8.5.1) 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, $5-20$).

8.7 Experiments and Results

In this section we show that the multi-view model introduced so far is able to successfully detect object categories and estimate objects pose. Furthermore, we demonstrate that the view synthesis machinery does improve detection and pose estimation accuracy when compared to the model that does not take advantage of the synthesis abilities. We collect our results in three set of experiments as described below.

8.7.1 Experiment I: Comparison with Thomas et al. [63]

We first conduct experiments on two known 3D object class datasets: the motorbikes and sport shoes used by Thomas et al. [63], provided by PASCAL Visual Object Classes (VOC) Challenge [15]. For fair comparison, we use the same testing images in both these classes as in [63]. Specifically, 179 images from the 'motorbikes-test2' set and 101 images from the sport shoes testing set are used. The models are learnt by using the provided image set of 16 motorbike instances and 16 shoe instances (each instance has 12-16 poses). We evaluate the categorization and localization performance by using precision-recall curves, under exactly the same conditions as stated by [63]. Figure 8.11 illustrates our results. In both the motorbike and shoe classes, the proposed algorithm significantly outperforms [63].

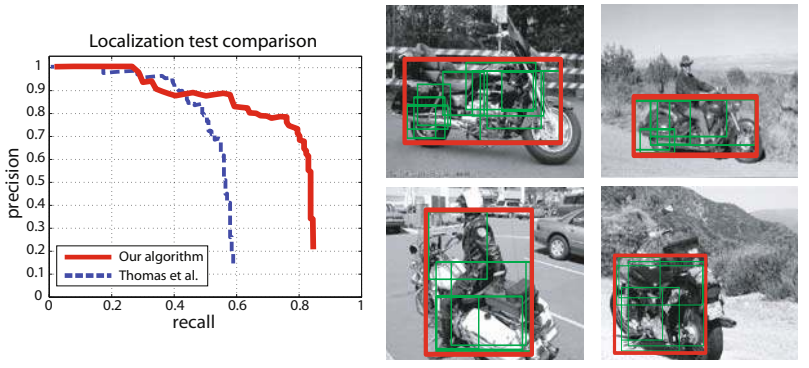

Fig. 8.11 Localization experiment compared with [63]. The precision-recall curves are generated under the PASCAL VOC protocol. Example detections are shown for both the motorbike and shoe datasets.

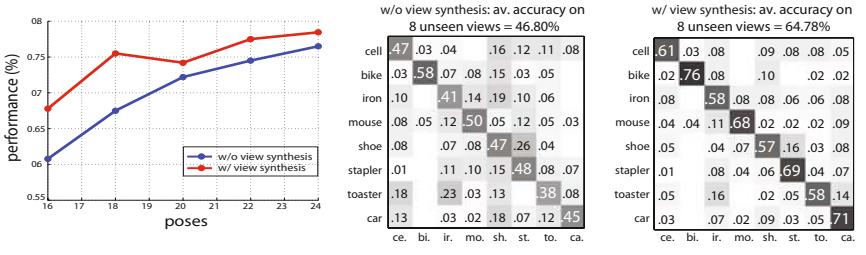

Fig. 8.12 Left: Performances of the model with (red) and without (blue) view synthesis as a function of the number of views used in training. Note that the performances shown here are testing performances, obtained by an average over all 24 testing poses. Middle: Confusion table results obtained by the model without view synthesis for 8 object classes on a sample of 8 unseen views only (dataset [54]). Right: Confusion table results obtained by the model with view synthesis under the same conditions.

8.7.2 Experiment II: Detection and Pose Estimation Results on the Dataset in [54]

Next, we compare the performances of multi-view model algorithm with and without view-synthesis capabilities. The comparison is performed on the dataset presented in [54]. This dataset comprises images of 8 different object categories (car, stapler, iron, shoe, monitor, computer mouse, head, bicycle, toaster and cellphone), 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, Figure 8.2) 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,

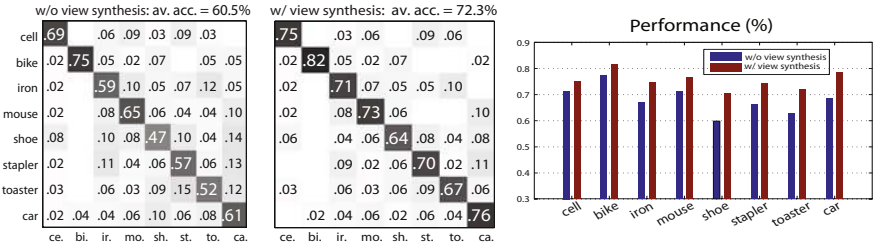

Fig. 8.13 Left: Confusion table results obtained by the model without view synthesis for 8 object classes (dataset [55]). Middle: Confusion table results obtained by the model with view synthesis under the same conditions. Right: Performance improvement achieved by the model with view synthesis for each category.

the number and type of poses in the test set are the same as in the training set. To learn each category, we randomly select 7 object instances to build the model, and 3 novel object instances. The farthermost scale is not considered in the current results. Figure 8.14 is a summary of learnt models for 8 object categories. The 3^{rd} column of Figure 8.14 visualizes the learnt model of each object category. We show in this panel a single object instance from the training images. Each dashed box indicate a particular view of the object instance. A subset of the learnt canonical parts is presented for each view. Across from different views, the canonical parts relationships are denoted by the arrows. Note that for clarity, we only visualize a small number of canonical parts as well as their $\mathcal{H}$. To illustrate the appearance variability, we show in the 4^{th} column different examples of a given canonical part. For each object model, 3 or 4 canonical parts are shown, indicated by the boxes. For each canonical part (i.e., within each box), we show a number of examples that belong to the same part. Note that these parts not only share a similar appearance, but also similar locations with respect to the object. The 1^{st} column of Figure 8.14 presents two correctly identified sample testing images. The red bounding box on each image indicates the best combination of canonical parts (i.e., that of the smallest error function), whereas the thin green boxes inside the red box correspond to the canonical parts of detected on the object. Using the pose estimation scheme, we are able to predict which pose this particular instance of the model comes from. Finally we present the binary detection result in ROC curves in the 2^{nd} column.

To assess the ability of the view-synthesis algorithm to improve detection and pose estimation in presence of views that have not been seen in training, we tested the algorithm using a reduced set of poses in training. The 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

[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.

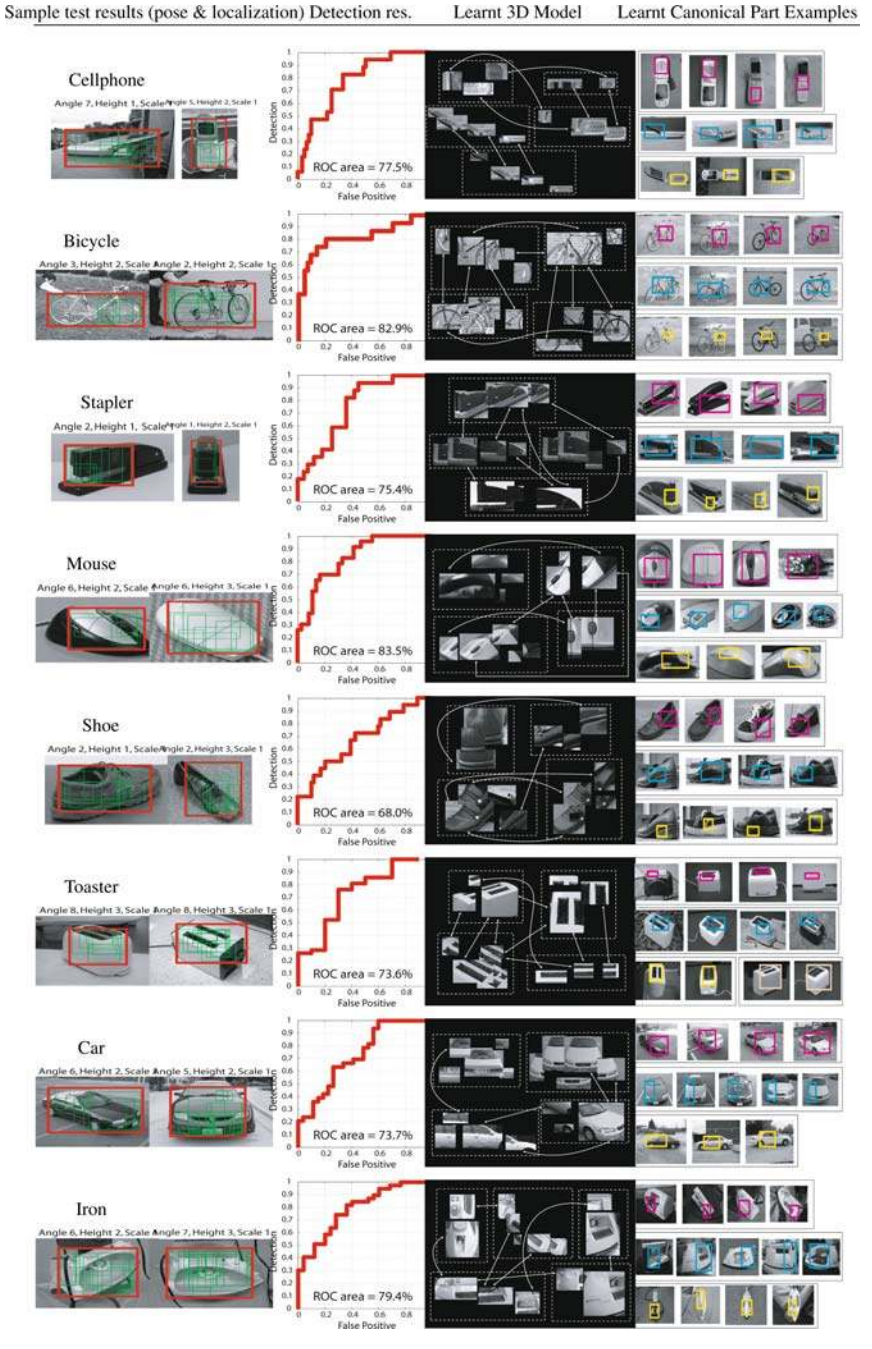

Fig. 8.14 Summary of the learnt 3D object category models, sample test images and binary detection results (ROC). Details of the figure is explained in Section8.7.2.

Fig. 8.15 Estimated pose for each object that was correctly classified by the 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 the recognition algorithm. For visualization purposes we illustrate these model views by showing the corresponding training images (columns 1-2 and 4-5). Images belong to the dataset in [55].

testing have not been presented during training. Figure 8.12 illustrates the performances of the two models (with and without view-synthesis) as a function of the number of views used in training. The plots shows that method which uses the two-step algorithm systematically outperforms the one that does only uses the first step. 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 two methods become similar. For a baseline comparison with a pure bag-of-world model the reader can refer to [54]. Figure 8.12(middle, right) compare the confusion table results obtained by the models with and without view-synthesis for 8 object classes on a sample of 8 unseen views only.

8.7.3 Experiment III: Detection and Pose Estimation Results on the Dataset in [55]

In this experiment we test the algorithm on a more challenging dataset [55]. While in [54] view points in testing and training are very similar, the dataset in [55] comprises objects portrayed under generic uncontrolled view points. Specifically, in [55] 7 (out of 8) classes of images (*cellphone, bike, iron, shoe, stapler, mouse, 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 the work so as to obtain a set of 60 images for each category. The 8^{th} class (i.e., *car*) is from the LabelMe dataset [53]. A sample of the dataset is available at [55]. As in the previous experiment, we compare the performances of the algorithm with or without view-synthesis. Results by both models are reported in Figure 8.13. Again, the method that uses the two-step algorithm achieves better overall results. Figure 8.13 (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. All the experiments presented in this section use the 2-view synthesis scheme. The 3-view scheme, along with the introduction of a more sophisticated probabilistic model, has been recently employed in [61,60]. Figure 8.15 illustrates a range of pose estimation results on the new dataset. See Figure 8.15 caption for details.

8.8 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 their categorical identity. In this Chapter we have introduced a new recognition paradigm for tackling these challenging problems which consists of linking together diagnostic parts or features of the object from different viewing points. We focused on recent work by [54, 55] and presented the details of the multi-view part-based model in [54, 55], relevant learning and recognition algorithms, as well as practical implementation details. We

have shown that such as a model can be learnt via minimal supervision and used to detect objects under arbitrary and/or unseen poses by means of a two-step algorithm. Experimental validation aimed at demonstrating the ability of the algorithm to recognize objects and estimate their pose have produced promising results. A number of open issues remain. The presented algorithms still require large number of views in training in order to generalize. More analysis needs 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 synthesis formulation. Finally, new solutions are required for incorporating the ability to model non-rigid objects.

References

1. The princeton shape benchmark. In: Proceedings of the Shape Modeling International, pp. 167–178 (2004)
2. Arie-Nachimson, M., Basri, R.: Constructing implicit 3d shape models for pose estimation. In: Proceedings of the International Conference on Computer Vision (2009)
3. Ballard, D.H.: Generalizing the hough transform to detect arbitrary shapes. Pattern Recognition 13(2), 111–122 (1981)
4. 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)
5. Bowyer, K., Dyer, R.: Aspect graphs: An introduction and survey of recent results. International Journal of Imaging Systems and Technology 2(4), 315–328 (1990)
6. Brown, M., Lowe, D.G.: Unsupervised 3d object recognition and reconstruction in unordered datasets. In: Proceedings of the Fifth International Conference on 3-D Digital Imaging and Modeling, pp. 56–63 (2005)
7. Burl, M.C., Weber, M., Perona, P.: A probabilistic approach to object recognition using local photometry and global geometry. In: Burkhardt, H., Neumann, B. (eds.) ECCV 1998. LNCS, vol. 1407, p. 628. Springer, Heidelberg (1998)
8. Chen, S., Williams, L.: View interpolation for image synthesis. Computer Graphics 27, 279–288 (1993)
9. Chiu, H.P., Kaelbling, L.P., Lozano-Perez, T.: Virtual training for multi-view object class recognition. In: Proceedings of the International Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2007)
10. Cyr, C., Kimia, B.: A similarity-based aspect-graph approach to 3D object recognition. International Journal of Computer Vision 57(1), 5–22 (2004)
11. Dance, C., Willamowski, J., Fan, L., Bray, C., Csurka, G.: Visual categorization with bags of keypoints. In: Proceedigs of the ECCV International Workshop on Statistical Learning in Computer Vision (2004)
12. Dickinson, S.J., Pentland, A.P., Rosenfeld, A.: 3-d shape recovery using distributed aspect matching. IEEE Transaction on Pattern Analisys and Machine Intelligence 14(2), 174–198 (1992)
13. Eggert, D., Bowyer, K.: Computing the perspective projection aspect graph of solids of revolution. IEEE Transaction on Pattern Analisys and Machine Intelligence 15(2), 109–128 (1993)
14. Eggert, D., Bowyer, K., Dyer, C., Christensen, H., Goldgof, D.: The scale space aspect graph. IEEE Transactions on Pattern Analysis and Machine Intelligence 15(11), 1114–1130 (1993)

15. Everingham, M., et al.: The 2005 pascal visual object class challenge. In Proceedings of the 1st PASCAL Challenges Workshop (to appear)
16. Farhadi, A., Tabrizi, J., Endres, I., Forsyth, D.: A latent model of discriminative aspect. In: Proceedings of the International Conference on Computer Vision (2009)
17. Fei-Fei, L., Fergus, R., Torralba, A.: Recognizing and learning object categories. CVPR Short Course (2007)
18. Felzenszwalb, P., Huttenlocher, D.: Pictorial structures for object recognition. In: Proceedings of the International Conference on Computer Vision and Pattern Recognition, pp. 2066–2073 (2000)
19. Fergus, R., Perona, P., Zisserman, A.: Object class recognition by unsupervised scale-invariant learning. In: Proceedings of the International Conference on Computer Vision and Pattern Recognition, pp. 264–271 (2003)
20. Ferrari, V., Tuytelaars, T., Van Gool, L.: Simultaneous object recognition and segmentation from single or multiple model views. Iternational Journal of Computer Vision (2006)
21. Fischler, M., Bolles, R.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Comm. of the ACM 24, 381–395 (1981)
22. Frome, A., Huber, D., Kolluri, R., Bulow, T., Malik, J.: Recognizing objects in range data using regional point descriptors. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3023, pp. 224–237. Springer, Heidelberg (2004)
23. Fulkerson, B., Vedaldi, A., Soatto, S.: Class Segmentation and Object Localization with Superpixel Neighborhoods. In: Proceedings of the International Conference on Computer Vision (2009)
24. Grimson, W., Lozano-Perez, T.: Recognition and localization of overlapping parts in two and three dimensions. In: Proceedings of the International Conference on Robotics and Automation, pp. 61–66 (1985)
25. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press, Cambridge (2004)
26. Hetzel, G., Leibe, B., Levi, P., Schiele, B.: 3d object recognition from range images using local feature histograms. IEEE Transactions on Pattern Analysis and Machine Intelligence 2 (2001)
27. Hoeim, D., Rother, C., Winn, J.: 3d layout crf for multi-view object class recognition and segmentation. In: Proceedings of the International Conference on Computer Vision and Pattern Recognition (2007)
28. Johnson, A., Hebert, M.: Using spin images for efficient object recognition in cluttered 3d scenes. IEEE Transactions on Pattern Analysis and Machine Intelligence 5 (1999)
29. Kadir, T., Brady, M.: Scale, saliency and image description. International Journal of Computer Vision 45(2), 83–105 (2001)
30. Kazhdan, M., Funkhouser, T., Rusinkiewicz, S.: Rotation invariant spherical harmonic representation of 3d shape descriptors. In: Proceedings of the Symposium on Geometry Processing (2003)
31. Koenderink, J., van Doorn, A.: The singularities of the visual mappings. Biological Cybernetics 24(1), 51–59 (1976)
32. Koenderink, J.J., van Doorn, A.J.: The internal representation of solid shape with respect to vision. Biological cybernetics 32(4), 211–216 (1979)
33. Kushal, A., Schmid, C., Ponce, J.: Flexible object models for category-level 3d object recognition. In: Proceedings of the International Conference on Computer Vision and Pattern Recognition (2007)
34. Lazebnik, S., Schmid, C., Ponce, J.: Semi-local affine parts for object recognition. In: Proceedings of the British Machine Vision Conference, vol. 2, pp. 959–968 (2004)

35. Leibe, B., Schiele, B.: Scale Invariant Object Categorization Using a Scale-Adaptive Mean-Shift Search. In: Rasmussen, C.E., Bülthoff, H.H., Schölkopf, B., Giese, M.A. (eds.) DAGM 2004. LNCS, vol. 3175, pp. 145–153. Springer, Heidelberg (2004)
36. Li, X., Guskov, I., Barhak, J.: Feature-based alignment of range scan data to cad model. International Journal of Shape Modeling 13, 1–23 (2007)
37. Liebelt, J., Schmid, C., Schertler, K.: Viewpoint-independent object class detection using 3d feature maps. In: Proceedings of the International Conference on Computer Vision and Pattern Recognition (2008)
38. Lowe, D.: Object recognition from local scale-invariant features. In: Proceedings of the International Conference on Computer Vision, pp. 1150–1157 (1999)
39. Lowe, D.G.: Three-dimensional object recognition from single two-dimensional images. Artificial Intelligence 31, 355–395 (1987)
40. Lowe, D.G.: Local feature view clustering for 3d object recognition. In: Proceedings of the International Conference on Computer Vision and Pattern Recognition (2001)
41. Marr, D.: Vision: A computational investigation into the human representation and processing of visual information. Freeman, New York (1982)
42. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide baseline stereo from maximally stable extremal regions. In: Proceedings of the British Machine Vision Conference, pp. 384–393 (2002)
43. Mei, L., Sun, M., Carter, K., Hero, A., Savarese, S.: Object pose classification from short video sequences. In: Proceedings of the British Machine Vision Conference (2009)
44. Mikolajczyk, K., Schmid, C.: An affine invariant interest point detector. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2350, pp. 128–142. Springer, Heidelberg (2002)
45. Mikolajczyk, K., Schmid, C.: Scale and affine invariant interest point detectors. International Journal of Computer Vision 60(1), 63–86 (2004)
46. Murase, H., Nayar, S.K.: Learning by a generation approach to appearance-based object recognition. In: Proceedings of the International Conference on Pattern Recognition (1996)
47. Nayar, S.K., Nene, S.A., Murase, H.: Real-time 100 object recognition system. In: Proceedings of the International Conference on Robotics and Automation, pp. 2321–2325 (1996)
48. Ng, J., Gong, S.: Multi-view face detection and pose estimation using a composite support vector machine across the view sphere. In: Proceedings of the International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems (1999)
49. Ozuysal, M., Lepetit, V., Fua, P.: Pose estimation for category specific multiview object localization. In: Proceedings of the International Conference on Computer Vision and Pattern Recognition (2009)
50. Rothganger, F., Lazebnik, S., Schmid, C., Ponce, J.: 3d object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints. International Journal of Computer Vision 66(3), 231–259 (2006)
51. Rothwell, C.A., Zisserman, A., Forsyth, D.A., Mundy, J.L., Joseph, L.: Canonical frames for planar object recognition. In: Sandini, G. (ed.) ECCV 1992. LNCS, vol. 588. Springer, Heidelberg (1992)
52. Ruiz-Correa, S., Shapiro, L., Meila, M.: A new signature-based method for efficient 3-d object recognition. In: Proceedings of the International Conference on Computer Vision and Pattern Recognition (2001)
53. Russell, B., Torralba, A., Murphy, K., Freeman, W.: Labelme: a database and web-based tool for image annotation. International Journal of Computer Vision (in press)

54. Savarese, S., Fei-Fei, L.: 3D generic object categorization, localization and pose estimation. In: Proceedings of the International Conference on Computer Vision, pp. 1–8 (2007)
55. Savarese, S., Fei-Fei, L.: View synthesis for recognizing unseen poses of object classes. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS, vol. 5304, pp. 602–615. Springer, Heidelberg (2008)
56. Schneiderman, H., Kanade, T.: A statistical approach to 3D object detection applied to faces and cars. In: Proceedings of the International Conference on Computer Vision and Pattern Recognition, pp. 746–751 (2000)
57. Seitz, S., Dyer, C.: View morphing. In: Proceedings of the ACM SIGGRAPH, pp. 21–30 (1996)
58. Shimshoni, I., Ponce, J.: Finite-resolution aspect graphs of polyhedral objects. IEEE Transaction on Pattern Analysis Machine Intelligence 19(4), 315–327 (1997)
59. Stewman, J., Bowyer, K.: Learning graph matching. In: Proceedings of the International Conference on Computer Vision, pp. 494–500 (1988)
60. Su, H., Sun, M., Fei-Fei, L., Savarese, S.: Learning a dense multi-view representation for detection, viewpoint classification and synthesis of object categories. In: Proceedings of International Conference on Computer Vision (2009)
61. Sun, M., Su, H., Savarese, S., Fei-Fei, L.: A multi-view probabilistic model for 3d object classes. In: Proceedings of the International Conference on Computer Vision and Pattern Recognition (2009)
62. Tangelder, J.W.H., Veltkamp, R.C.: A survey of content based 3d shape retrieval methods. In: Proceedings of Shape Modeling Applications, pp. 145–156 (2004)
63. Thomas, A., Ferrari, V., Leibe, B., Tuytelaars, T., Schiele, B., Van Gool, L.: Towards multi-view object class detection. In: Proceedings of the International Conference on Computer Vision and Pattern Recognition, pp. 1589–1596 (2006)
64. Torralba, A., Murphy, K., Freeman, W.: Sharing features: efficient boosting procedures for multiclass object detection. In: Proceedings of the International Conference on Computer Vision and Pattern Recognition (2004)
65. Ullman, S., Basri, R.: Recognition by linear combination of models. Technical Report, Cambridge, MA, USA (1989)
66. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proceedings of the International Conference on Computer Vision and Pattern Recognition, pp. 511–518 (2001)
67. Weber, M., Einhäuser, W., Welling, M., Perona, P.: Viewpoint-invariant learning and detection of human heads. In: Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition (2000)
68. Weber, M., Welling, M., Perona, P.: Unsupervised learning of models for recognition. In: Vernon, D. (ed.) ECCV 2000. LNCS, vol. 1842, pp. 101–108. Springer, Heidelberg (2000)
69. Xiao, J., Chen, J., Yeung, D.Y., Quan, L.: Structuring visual words in 3d for arbitrary-view object localization. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS, vol. 5304, pp. 725–737. Springer, Heidelberg (2008)
70. Yan, P., Khan, D., Shah, M.: 3d model based object class detection in an arbitrary view. In: Proceedings of the International Conference on Computer Vision (2007)
71. Yan Li Leon Gu, T.K.: A robust shape model for multi-view car alignment. In: Proceedings of the International Conference on Computer Vision and Pattern Recognition (2009)
72. Zhang, Z.: Floatboost learning and statistical face detection. IEEE Transaction on Pattern Analysis Machine Intelligence 26(9), 1112–1123 (2004)