3-D Object Recognition Using 2-D Views

Wenjing Li, *Member, IEEE*, George Bebis, *Member, IEEE*, and Nikolaos G. Bourbakis, *Fellow, IEEE*

***Abstract*—We consider the problem of recognizing 3-D objects from 2-D images using geometric models and assuming different viewing angles and positions. Our goal is to recognize and localize instances of specific objects (i.e., model-based) in a scene. This is in contrast to category-based object recognition methods where the goal is to search for instances of objects that belong to a certain visual category (e.g., faces or cars). The key contribution of our work is improving 3-D object recognition by integrating Algebraic Functions of Views (AFoVs), a powerful framework for predicting the geometric appearance of an object due to viewpoint changes, with indexing and learning. During training, we compute the space of views that groups of object features can produce under the assumption of 3-D linear transformations, by combining a small number of reference views that contain the object features using AFoVs. Unrealistic views (e.g., due to the assumption of 3-D linear transformations) are eliminated by imposing a pair of rigidity constraints based on knowledge of the transformation between the reference views of the object. To represent the space of views that an object can produce compactly while allowing efficient hypothesis generation during recognition, we propose combining indexing with learning in two stages. In the first stage, we sample the space of views of an object *sparsely* and represent information about the samples using indexing. In the second stage, we build probabilistic models of shape appearance by sampling the space of views of the object *densely* and learning the manifold formed by the samples. Learning employs the Expectation-Maximization (EM) algorithm and takes place in a "universal," lower-dimensional, space computed through Random Projection (RP). During recognition, we extract groups of point features from the scene and we use indexing to retrieve the most feasible model groups that might have produced them (i.e., hypothesis generation). The likelihood of each hypothesis is then computed using the probabilistic models of shape appearance. Only hypotheses ranked high enough are considered for further verification with the most likely hypotheses verified first. The proposed approach has been evaluated using both artificial and real data, illustrating promising performance. We also present preliminary results illustrating extensions of the AFoVs framework to predict the intensity appearance of an object. In this context, we have built a hybrid recognition framework that exploits geometric knowledge to hypothesize the location of an object in the scene and both geometrical and intesnity information to verify the hypotheses.**

I. Introduction

THE ability to recognize objects and identify their positions in 3-D is one of the most fascinating skills of the human visual system. This skill allows people to navigate in both familiar and unfamiliar environments, interact with surrounding objects, avoid obstacles, and identify hazards. Although recognition is a spontaneous, natural activity for humans and other biological vision systems, building systems capable of recognizing relevant objects in their environment with accuracy and robustness has been a difficult and challenging task in computer vision [1]–[3]. This is mainly because the appearance of an object can have a large range of variation due to photometric effects, scene clutter, changes in shape (e.g., nonrigid objects), and most importantly, viewpoint changes. As a result, different views of the same object can give rise to widely different images.

Currently, there are two main research directions in object recognition: i) *geometrical* models, largely based on shape information and ii) *intensity* models, also known as *empirical* models [4], are based on direct representations of image intensity. Geometrical models employ descriptions based on the projected boundary of an object. Typically, geometric invariant descriptions and index functions are employed for recognition. An excellent review on geometric models appears in [5]. On the other hand, intensity models describe an object by a set of images which is acquired for a range of views and illumination conditions that are expected to be encountered in subsequent recognition. To account for clutter, occlusions, and viewpoint changes, invariant local photometric descriptors are computed from interest regions of an object [6], [7]. A representative collection of papers in this area can be found in [8]. We review both approaches in Section II.

In this paper, we consider the problem of recognizing and localizing 3-D objects from 2-D images using geometric information under different viewing angles and positions. Our approach is model-based, that is, we search for instances of specific objects in a scene. This is in contrast to category-based recognition where the goal is to search for instances of objects that belong to a certain visual category (e.g., faces or cars) [9]–[11]. Our approach employs AFoVs [12]–[14], a powerful framework for handling variations in the geometric appearance of an object due to viewpoint changes, which is also supported by psychophysical findings indicating that the human visual system works in a similar way [15], [16]. Given an image that might contain one or more objects in various positions and orientations, our goal is to identify and localize all the models present in the image. Pose information is only recovered implicitly by recovering the AFoVs parameters.

Simply speaking, AFoVs are functions that express a relationship among a number of views of an object in terms of their image coordinates alone. The main theoretical result indicates that "*the variety of 2-D views depicting the shape appearance of a 3-D object can be expressed as a combination of a small number of 2-D views of the object*." This suggests a simple but

Manuscript received June 3, 2007; revised June 21, 2008. First published September 30, 2008; current version published October 10, 2008. This work was supported by the Office of Naval Research (ONR) under grant N00014-01-0781. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Vicent Caselles.

W. Li was with the Computer Science and Engineering Department, University of Nevada, Reno, NV 89557 USA. She is now with STI Medical Systems, Honolulu, HI 96813 USA (e-mail: wli@sti-hawaii.com)

G. Bebis is with the Computer Science and Engineering Department, University of Nevada, Reno, NV 89557 USA (e-mail: bebis@cs.unr.edu).

N. G. Bourbakis is with the Information Technology Institute, Wright State University, Dayton, OH 45435 USA (e-mail: bourbaki@cs.wright.edu)

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2008.2003404

powerful framework for predicting shape appearance: "*novel 2-D views of a 3-D object can be predicted by combining a small number of known 2-D views of the object*." The main advantage of using the AFoVs framework for recognition is that it does not rely on invariants or full 3-D models. In fact, no camera calibration or 3-D scene recovery are required. Also, it is fundamentally different from multiple-view approaches which perform matching by comparing novel views to prestored views of the object (i.e., reference views). In contrast, AFoVs predict the geometric appearance of a 3-D object in an novel view by combining a small number of reference 2-D views of the object.

Although interesting and appealing, there are several restrictive assumptions behind the underlying theory that limit its practical use. For example, it is assumed that the feature correspondences between the novel and reference views are known or that the values of the AFoVs parameters are known. Moreover, it is assumed that all geometric features of an object are present in every view of the object (i.e., the objects are transparent). We have made some progress in our past work towards making AFoVs more practical for object recognition by: i) coupling AFoVs with indexing to bypass the correspondence problem [17], [18], ii) using groups of features to relax the requirement of transparent objects [17]–[19], and iii) estimating the ranges of values that the parameters of AFoVs can assume [17], [18], [20]–[22] using Singular Value Decomposition (SVD) [23], [24] and Interval Arithmetic (IA) [25], [26]. Using two reference views per object and assuming 3-D linear transformations, we have demonstrated the feasibility of our approach by recognizing novel object views from quite different viewpoints and locations.

This work builds upon our previous work on object recognition using AFoVs [17], [18], which is reviewed in Section III, with the goal of further improving its efficiency and performance. Although AFoVs allow us to compute the space of views that an object can produce, representing this space compactly and efficiently for recognition purposes is a critical issue. In the past, we simply sampled the space of views that an object can produce and represented information about the samples in an index table using hashing. Although this approach produces good results in practice, it has several drawbacks. First, it has high space requirements even for a moderate number of objects. Second, when generating the views of an object it is important to eliminate all unrealistic views (i.e., due to the assumption of 3-D linear transformations). Without imposing extra constraints on the generated views to identify and reject unrealistic views, certain hypothetical matches would be invalid, slowing down recognition. Third, implementing indexing using hashing is not very effective since it generates a large number of hypothetical matches, increasing recognition time. Finally, when predicting the geometric appearance of an object in the scene, it would help the recognition process if there was a scheme to estimate the likelihood of each prediction before applying formal verification.

We have addressed the above issues in this work, improving the AFoVs-based recognition framework in several important ways. *First*, when generating the space of views that an object can produce, we propose imposing a pair of rigidity constraints to identify and reject unrealistic views. This saves space and improves recognition time by reducing the number of invalid hypotheses. *Second*, when representing the space of views that an object can produce, we propose using a more powerful scheme that represents the space of views more compactly and efficiently for recognition purposes by combining indexing with learning in two stages. In the first stage, we sample the space of views of an object *sparsely* and represent the samples using indexing. In the second stage, we build probabilistic models of shape appearance by sampling the space of views of the object *densely* and learning the manifold formed by the samples. Learning employs the EM algorithm [27] and takes place in a "universal," lower-dimensional, space computed through RP [28], [29].

The purpose of the first stage is to generate rough hypothetical matches between the models and the scene fast, while keeping the space requirements of indexing low. To account for indexing a sparse number of views, we employ a powerful indexing scheme based on the Sproull k-d tree [30], [31] which performs nearest-neighbor search. The purpose of the second stage is to quickly filter out as many invalid hypotheses as possible avoiding explicit model verification which could be time consuming. It should be mentioned that once the probabilistic models of shape appearance have been built, we only need to store a few parameters per model (e.g., using mixtures of Gaussian, we need to store the mixing parameters as well as the mean and covariance of each component; see Section IV-C); therefore, the overall space requirements of the method depend mainly on indexing. *Finally*, when performing verification, we propose ranking the hypotheses and verifying the most likely hypotheses first. Ranking is a byproduct of representing the space of views of an object using probabilistic models of shape appearance and reduces recognition time significantly. An earlier version of this work has appeared in [32].

It is worth mentioning that using geometric models alone is a viable approach for recognition when color information does not provide sufficient discrimination information or when objects luck detailed texture. In general, geometric and intensity models could be combined in order to build more accurate and robust recognition systems [4]. In this context, we have extended the AFoVs recognition framework to predict both geometric and intensity object appearance. We present preliminary results where geometric information is used to establish a number of hypothetical matches between the models and the scene while both geometric and intensity information is used to verify the hypothetical matches.

The rest of the paper is organized as follows. In Section II, we provide a brief review of object recognition. Background information on AFoVs and our previous work on recognition using AFoVs is provided in Section III. The improved recognition framework using AFoVs is presented in detail in Section IV. Section V presents our experimental procedures and results using both artificial and real 3-D objects. Employing AFoVs to predict the intensity appearance of an object along with preliminary results showing how to employ both geomet-

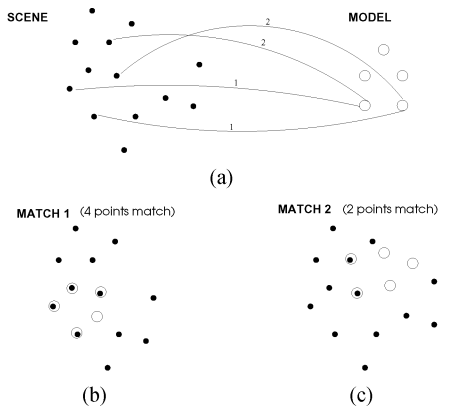

Fig. 1. (a) Hypothetical matches using two-point correspondences between a model and the scene. (b) *Match1* produces more matches (good hypothesis), while (c) *Match 2* fails to produce more matches (bad hypothesis).

rical and intensity models for recognition are presented in Section VI. Finally, our conclusions and plans for future work are given in Section VII.

II. Object Recognition Review

Geometrical models are based on the idea that much of the information about an object resides in its geometric properties. Thus, they use geometric models to represent the shape of the object and explore correspondences between the geometric model and image features during recognition. Given an unknown scene in this case, recognition implies the identification of a set of features from the unknown scene which approximately match a set of features from a known view of a model object. Fig. 1 shows an example where a set of point features are used for matching. In general, more robust and discriminative features can been considered for matching such as regions [33], [34] or convex groups of line segments [35].

Traditional object recognition systems often lack scalability when a large number of objects or extensive variations in object appearance are encountered [36]. Since usually there is no *a priori* knowledge of which model features correspond to which scene features, recognition can be computationally too expensive, even for a moderate number of models. To limit the possible number of matches, perceptual organization cues have been typically employed in order to extract salient structures from the scene [35]. Also, methods based on geometric constraints [37] or minimum possible number of feature correspondences [38] have been proposed. Indexing is an alternative paradigm which uses *a priori* stored information to quickly eliminate incompatible matches during recognition [39]–[43].

Accommodating variations due to viewpoint changes is a central problem in the design of any object recognition system. Typical strategies for dealing with the varying geometrical appearance of an object due to viewpoint changes include using geometric invariants [39], [41], explicit 3-D models [38], [44], [45], and multiple views [46], [36]. According to the first strategy, invariant properties of geometric features (i.e., properties that vary little or not at all as viewing conditions change) have been employed during recognition. In particular, indexing based on invariants is very efficient since only a single entry for each set of features needs to be stored, regardless of changes in the viewpoint. The main problem with this strategy is that it is difficult or even impossible to find general case geometric invariants (e.g., no general case invariants exist in the case of visual images under 3-D perspective projection) [47], [48].

The second strategy employs explicit geometric 3-D models (e.g., CAD models). When used with indexing, 2-D appearances of the 3-D model are generated and arranged in the index space [38], [44], [45]. During recognition, a model of the image formation process is applied on the 3-D model objects in order to generate predictions of the appearance of the objects. This strategy is not very practical since 3-D models are not always available. The last strategy models an object by a set of 2-D views (i.e., reference views), essentially showing how the shape of the object appears from different viewpoints. The aspect graph is probably the most well known multiview representation approach [49], [50]. The aspect graph of an object is a graph structure where every node in the graph represents a topologically distinct view of the object. Arcs between nodes represent visual events that occurs as we move from one view to another by changing viewpoints. Methods based on this strategy perform recognition by matching one of the reference views to some part of the novel view. This strategy is not very efficient because, in general, a large number of views must be stored for each model object.

Intensity models do not need to recover object geometry and can learn the appearance characteristics of an object from training imagery. One idea to capture intensity appearance is by using grayscale or color histograms [51], [52]. To handle illumination changes, several robust measures have been proposed [53]. Another idea to capture intensity appearance involves enumerating many possible object appearances in advance, obtained under various viewpoints and possibly different lighting conditions. In this case, each model view is stored as a vector of image intensities, represented in a low dimensional space [54]. This approach was first used in [55] for face recognition using Principal Component Analysis (PCA) for dimensionality reduction. Later, this approach was extended for general object recognition by parameterizing the PCA space in terms of pose and illumination [56]. A hyper-surface in this space represents a particular object. Recognition was performed by projecting the image of an object onto a point in the eigenspace. The object was recognized based on the hyper-surface on which it lies. The exact location of the point determines the pose of the object. Employing feature selection schemes in PCA spaces to select subsets of features has shown to improve object detection results [57].

The most severe limitation of intensity-based approaches is that they require isolating the object of interest from the background. Although some of these methods have been demonstrated successfully on isolated objects or presegmented images, it is difficult to extend them to cluttered and partially occluded images due to their global features. To improve the robustness of intensity-based methods to occlusion and clutter, component-based object representation schemes have been proposed [58]. A key issue in this case is how to choose an appropriate set of component to represent an object. For certain object classes (e.g., faces or cars), this is quite intuitive and the components can be chosen manually [58], [59], [60]; however, a more systematic approach is needed in general. Lately, the emphasis has been on representing objects in terms of a large number of local descriptors which are computed by applying generic "interest" operators and computing invariant photometric and image transformation descriptions from a local region in a vicinity around the interest points [7], [61], [62]. Using local descriptors for recognition has several advantages including that they are more distinctive, robust to occlusion, less sensitive to local image distortions (e.g., scale changes, 3-D viewpoint changes) and do not require segmentation.

One way to compute a set of local descriptors is by placing a grid on the image and applying certain filters (e.g., Gabor) at the nodes of the grid [9]. However, centering the grid on the object when there is occlusion might be difficult. To deal with the issues of occlusion and clutter, first a number of landmarks are detected on the surface of the object by applying various "interest" operators such as the Moravec operator [63] or the Harris corner detector [64]. Then, local descriptors are computed from a small region around each interest point. During training, local descriptors are extracted from a set of reference images and stored in a database. During recognition, local descriptors are extracted from a new image; then, they are matched to the database to establish candidate matches. Fast indexing or nearest-neighbor algorithms are typically used to speed up this step. Consisted groups of matched features (i.e., groups of features belonging to a single object) can be found using transformation clustering. To eliminate false positives, each match is filtered by identifying subsets of features that agree on the object and its location, scale, and orientation in the new image. Finally, verification is performed, for example, using a least-squares approach, to find consistent pose parameters.

Using local descriptors for category-based object recognition relies more on powerful representation schemes and learning algorithms than recognizing specific objects. Due to the challenges involved in representing and learning the variability between objects in the same category as well as objects between different categories, issues related to viewpoint changes have not been treated explicitly. Instead, they have been considered as another factor contributing to the variability of objects in the same category. Therefore, the emphasis has been how to "learn" this variability using either generative or discriminative models. Examples of category-based object recognition methods include the "Bag of Keypoints" approach [10], [65], the "constellation" model approach [66], [67], and the "pictorial structures" approach [60], and the "boosting" approach [68].

Two important issues must be addressed when using using local descriptors for object recognition. First, how to extract "interest" points that are invariant to illumination changes and image transformations and second, how to compute a local descriptor for a small region around each interest point that is also invariant to illumination changes and image transformations. Both issues have been addressed extensively over the last few years. Regarding the issue of interest point detection, a lot of emphasis has been given on developing scale and affine invariant detectors while being also resistant to illumination changes. Examples include the Kadir–Brady detector [69], the Difference of Gaussians detector [62], the Harris–Laplace detector [70], and the affine extension of the Harris–Laplace detector [6]. A detailed comparison of various invariant detectors can be found in [6], [71]. Typically, interest point detectors return some additional information that allows for determining a scale or affine invariant region around each interest point. To describe the local region around each interest point, various photometric, scale and affine invariant descriptors have been proposed. Examples include SIFT features [62], PCA-SIFT features [72], moment invariants [73], and complex filters [74]. A detailed comparison of various descriptors can be found in [7].

In general, methods based on geometric models can handle viewpoint changes explicitly and are more appropriate when searching for instances of specific objects in the scene; however, they suffer from problems in extracting geometric features reliably and robustly, have difficulty in describing "nongeometric" objects, and are not quite capable in differentiating between a large number of objects. On the other hand, methods based on intensity models employ more powerful feature extraction schemes, allowing them to deal more effectively with clutter and occlusion, and can distinguish among a large numbers of objects. However, they cannot deal well with objects lacking detailed texture while viewpoint changes are typically handled implicitly by considering them as an extra factor of variation that must be "learned" into the model.

Using geometric or intensity models alone should not be expected to address very challenging recognition problems. A more promising direction is probably integrating geometric with intensity models [4]. The benefits of incorporating geometric information in intensity models has been demonstrated in a number of studies including [10], [60], [62], [66], [67], [75]. In some cases, a geometric model is used to confirm the presence of an object in the scene [62], [75], while in other cases, geometric information is integrated with intensity information in the same model [10], [60], [66], [67]. In our work, we have extended the AFoVs framework to predict both the geometric and intensity appearance of an object. We present preliminary results illustrating cases where geometric information is used to hypothesize the location of objects in the scene and both geometric and intensity information are used to verify the hypotheses.

III. Background

This section provides a brief review of the theory of AFoVs as well as our previous work on 3-D object recognition using AFoVs. Detailed information about AFoVs can be found in [12], [76], [14], [13], [77], and [78] and about our work in [17]–[22].

A. Review of AFoVs

Simply speaking, AFoVs are functions which express a relationship among a number of views of an object in terms of their image coordinates alone. In particular, it has been shown that if we let an object undergo 3-D rigid transformations and assume that the images of the object are obtained by orthographic projection followed by uniform scaling (i.e., a good approximation to perspective projection when the camera is far from the object), then novel views of the object can be expressed as a *linear* combination of three other views of the same object (i.e., reference views) [12]. This result can be simplified by removing the

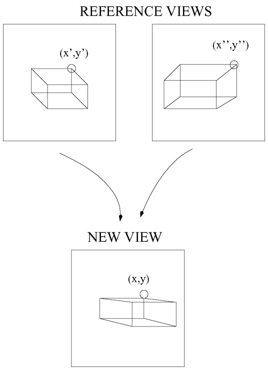

Fig. 2. Novel views can be obtained by combining a small number of reference views. This figure illustrates how to predict the coordinates in the new view from the coordinates in the reference views using (1).

orthonormality constraint associated with the rotation matrix. In this case, the object undergoes 3-D linear transformations in space and AFoVs become simpler, involving only two reference views.

Specifically, let us consider two reference views V_1 and V_2 of the same object which have been obtained by applying different linear transformations, and two points $p' = (x', y'), p'' = (x'', y'')$, one from each view, which are in correspondence. Then given a novel view V of the same object which has been obtained by applying another linear transformation and a point $p = (x, y)$ which is in correspondence with point p' and p'', the coordinates of p can be expressed as a linear combination of the coordinates of p' and p'' as

$$\begin{aligned} x &= a_1 x' + a_2 y' + a_3 x'' + a_4 \\ y &= b_1 x' + b_2 y' + b_3 x'' + b_4 \end{aligned} \tag{1}$$

where the parameters $a_j, b_j, j = 1, \ldots, 4$, are the same for all the points which are in correspondence across the three views (see Fig. 2). It should be noted that the above equations can be re-written using the y-coordinates of the second reference view instead.

The above results hold true in the case of objects with sharp boundaries; however, similar results exist in the case of objects with smooth boundaries [76] as well as nonrigid (i.e., articulated) objects [12] (i.e., more reference views are required in these cases). The extension of AFoVs to the case of perspective projection has been carried out in [13] and [14]. In particular, it has been shown that three perspective views of an object satisfy a trilinear function. Moreover, Shashua [14] has shown that a simpler and more practical pair of algebraic functions exist when the reference views have been obtained under scaled orthographic projection (one perspective and two orthographic views satisfy a bilinear function). In this work, we consider the case of orthographic projection assuming 3-D linear transformations.

B. Recognition Using AFoVs

AFoVs are quite attractive for object recognition since variations in the geometrical appearance of an object can be represented in a simple and compact form for the system to handle. Given a novel view of an object, AFoVs can be used to predict the image coordinates of object features in the novel view by appropriately combining the image coordinates of corresponding object features across the reference views. We have exploited this idea in our past work to recognize unknown views of a 3-D object from a small number of 2-D reference views of the same object, assuming orthographic projection and 3-D linear transformations which can handle affine transformations [17], [18], [19]. To bypass the correspondence problem between the novel and reference views, we proposed coupling AFoVs with indexing. Moreover, we relaxed the requirement of transparent objects by applying AFoVs on groups of features (i.e., we do not require that all the features of an object are visible from any viewpoint).

Using AFoVs for recognition involves two main phases: *training* and *recognition* as illustrated in Fig. 3. During training, we sample the space of views that groups of point features can produce and represent the sampled views in a hash table using a simple hash function that involves scaling and quantizing the image coordinates. During recognition, groups of points are extracted from the scene and used to retrieve from the hash table the model groups that might have produced them (i.e., hypothesis generation). Each hypothesis is then verified to confirm the presence of the hypothesized model in the scene. To sample the space of views that groups of model points can produce, we sample the space of AFoVs parameters. The range of values that the parameters of AFoVs can assume were estimated using SVD [23], [24] and IA [25], [26] (also, see Section III-C). To reduce space requirements without degrading recognition accuracy, we have showed that it is possible to represent in the hash table only the x-coordinates (or y-coordinates) of the views at the cost of making hypothesis generation slightly more complicated (also, see Section IV-D).

Indexing-based 3-D object recognition using AFoVs offers a number of advantages. First of all, the index table can be built using a small number of reference views per object. This is in contrast to common approaches that build the index table using either a large number of reference views or 3-D models. Second, recognition does not rely on the similarity of the novel views with the reference views; all that is required for the novel views is to contain some common groups of features with the reference views. Third, verification becomes simpler. This is because candidate models can be back-projected onto the scene by simply combining a small number of reference views of the candidate model group using the predicted AFoVs parameters. Finally, the proposed approach is more general and extendible since there exist AFoVs over a wide range of transformations and projections as mentioned in the previous section.

C. Estimating the Ranges of AFoVs Parameters

In this section, we present briefly the main ideas for estimating the ranges of values of the AFoVs parameters. Under the

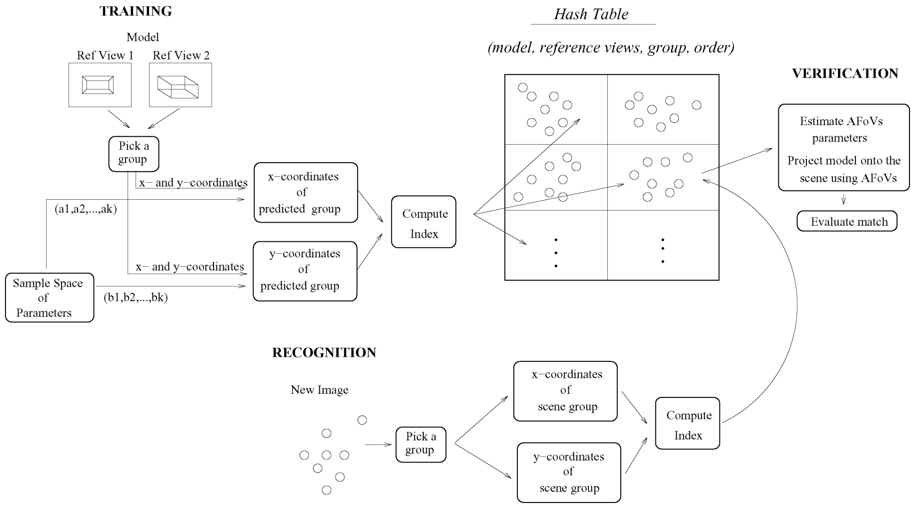
TRAINING
Model
Ref View 1
Ref View 2
Pick a group
x– and y–coordinates
(a1,a2,...,ak)
x–coordinates of predicted group
Compute Index
y–coordinates of predicted group
x– and y–coordinates
Sample Space of Parameters
(b1,b2,...,bk)
Hash Table
(model, reference views, group, order)
VERIFICATION
Estimate AFoVs parameters
Project model onto the scene using AFoVs
Evaluate match
RECOGNITION
New Image
Pick a group
x–coordinates of scene group
y–coordinates of scene group
Compute Index

Fig. 3. Original AFoVs-based recognition framework (from [17]). During training, we sample the space of views that groups of point features can produce and represent the sampled views in a hash table. During recognition, groups of points are extracted from the scene and used to retrieve from the hash table the model groups that might have produced them (i.e., hypothesis generation). Each hypothesis is then verified to confirm the presence of the hypothesized model in the scene.

assumption of orthographic projection and 3-D linear transformations, given the point correspondences between the reference views and the novel view, the following system of equations is satisfied from (1):

$$\begin{bmatrix} x'_1 & y'_1 & x''_1 & 1 \\ x'_2 & y'_2 & x''_2 & 1 \\ \cdots & \cdots & \cdots & 1 \\ x'_N & y'_N & x''_N & 1 \end{bmatrix} \begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \\ a_3 & b_3 \\ a_4 & b_4 \end{bmatrix} = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \cdots & \cdots \\ x_N & y_N \end{bmatrix} \tag{2}$$

where (x'_1, y'_1), (x'_2, y'_2), …, (x'_N, y'_N) and (x''_1, y''_1), (x''_2, y''_2), …, (x''_N, y''_N) are the coordinates of the points in the reference views V_1 and V_2 respectively, and (x_1, y_1), (x_2, y_2), …, (x_N, y_N) are the coordinates of the points in the novel view V. Dividing the above system of equations into two sub-systems, one involving the a_j parameters and one involving the b_j parameters, we have

$$Pc_1 = p_x \tag{3}$$

$$Pc_2 = p_y \tag{4}$$

where P is the matrix formed by the x and y coordinates of the reference views [same as the first matrix in the left part of (2)]. c_1 and c_2 are vectors corresponding to a_j's and b_j's, and p_x, p_y are vectors corresponding to the x and y coordinates of the novel view. Since both (3) and (4) are over-determined, they can be solved using SVD.

To determine the possible range of values for c_1 and c_2, we assume first that the novel view has been scaled such that the x and y coordinates assume values in a specific interval. This can be done, for example, by mapping the novel view to the unit square. In this case, the x and y coordinates would assume values in [1]. To determine the range of values for c_1 and c_2, we need to consider all possible solutions of (3) and (4), assuming that p_x and p_y belong to the interval [1]. We have used IA (i.e., Interval Arithmetic) [25], [26] to find the interval solutions of this problem (see [17] and [18] for details). It should be noted that since both (3) and (4) involve the same matrix P and p_x and p_y assume values in the same interval, the c_1 and c_2 interval solutions are the same.

IV. Improving Recognition Using AFoVs

The main advantage of AFoVs is that they allow us to compute the space of views that a 3-D object can produce by combining a small number of reference views of the object. We have exploited this idea for recognition by sampling the space of 2-D views that an object can produce (i.e., by sampling the space of the AFoVs parameters) and representing appropriate information about the samples in a hash table. This information is used during recognition to hypothesize the presence of certain models in the scene. This work builds upon our previous work on object recognition using AFoVs with the goal of further improving its performance and efficiency. In particular, we have addressed a number of important issues which are discussed below.

First, when sampling the space of AFoVs parameters to generate the 2-D views of an object, it is important to ensure that the generated views are realistic. However, since we assume the case of 3-D linear transformations and the ranges of AFoVs parameters are estimated using IA [25], [26], certain parameter values will not yield valid views. Although this issue does not have a serious effect on recognition performance, it does increase space requirements (i.e., invalid views are represented) and recognition time (i.e., invalid hypotheses are generated). In this work, we propose imposing a pair of rigidity constraints on the AFoVs parameters to avoid representing information about unrealistic views and reduce the number of invalid hypotheses during recognition.

Second, despite the well-known advantages of indexing for efficient hypothesis generation, recognition performance depends on the number of sampled views indexed. This is because general case invariants do not exist in 3-D [47], [48]; therefore, achieving high recognition rates requires indexing a large number of sampled views. However, this increases space requirements as well as recognition time due to an expected increase in the number of hypothetical matches. To take

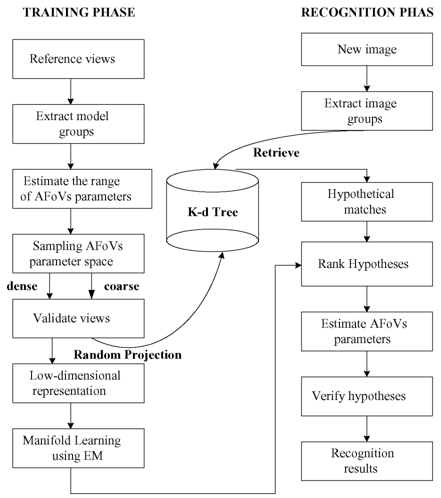

Fig. 4. Improved AFoVs-based recognition framework (see text for details).

advantage of efficient hypothesis generation using indexing while keeping space requirements low, we propose a two-stage scheme that combines indexing with learning.

The first stage involves indexing a *sparse* number of sampled views; therefore, space requirements are lower compared to the original approach that uses dense sampling. To account for indexing a small number of views, we have replaced the original indexing scheme based on hashing, which simply performs range search, with a more efficient scheme based on the Sproull k-d tree [30], [31], which performs nearest-neighbor search. The second stage involves building probabilistic models of shape appearance by sampling the space of views that an object can produce *densely* and learning the manifold formed by the samples. Learning employs the EM algorithm [27] and takes place in a "universal," lower-dimensional, space computed through RP (i.e., Random Projection) [28], [29]. Once the probabilistic models of shape appearance have been built, we only need to store a few parameters per model; therefore, space requirements are minimal and overall space requirements depend on indexing.

Specifically, the main purpose of the first stage is to provide rough hypothetical matches fast, while keeping storage requirements low. The main purpose of the second stage is to evaluate each hypothesis fast without resorting to explicit verification which might be time consuming. In fact, it should be expected that a fairly large number of neighbors needs to be retrieved from the k-d tree to ensure correct recognition results due to indexing a "sparse" number of views only. To avoid verifying each hypothesis explicitly, we "filter" them by computing their likelihoods using the probabilistic models of shape appearance built at the second stage.

Finally, we take advantage of representing shape appearance probabilistically to enhance the verification stage by ranking the hypotheses based on their likelihoods and verifying the most likely hypotheses first. Verification is performed as before by back-projecting the hypothesized model(s) onto the scene by combining a small number of reference views of the model(s), using the predicted AFoVs parameters [17]–[19]. Hypothetical matches are evaluated by computing the back-projection error between the model(s) and the scene.

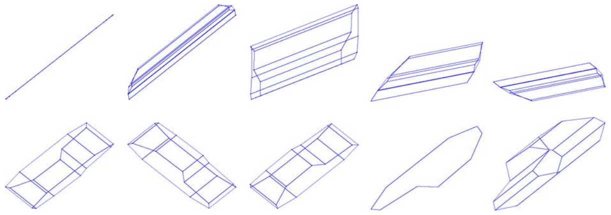

Fig. 5. (Top) Unrealistic views using a car model. (Bottom) Realistic views using rigidity constraints (threshold = 0.0001).

A. Eliminating Unrealistic Views

When sampling the AFoVs parameters using (1) in order to sample the space of views that an object can produce, it is possible to generate views that are not realistic in practice. There are two main reasons for this. First, the equations in (1) correspond to the case of 3-D linear transformations, a superset of 3-D rigid transformations. Second, the interval solutions for the AFoVs parameters are not tight in general [17], [18] which implies that certain solutions would not satisfy (2). Fig. 5 (top) shows several examples of unrealistic views generated using a car model.

We can identify and eliminate such views by imposing a pair of rigidity constraints. This, however, requires knowledge of the transformation between the reference views. Specifically, if we assume that the two reference views are related by some rotation R, then the AFoVs parameters in (1) must satisfy the following two constraints [12]:

$$a_1b_1 + a_2b_2 + a_3b_3 + (a_1b_3 + a_3b_1)r_{11} + (a_2b_3 + a_3b_2)r_{12} = 0 \quad (5)$$

$$a_1^2 + a_2^2 + a_3^2 - b_1^2 - b_2^2 - b_3^2 - 2(b_1b_3 - a_1a_3)r_{11} - 2(b_2b_3 - a_2a_3)r_{12} = 0 \quad (6)$$

where r_{11} and r_{12} are the first two elements of the rotation matrix R. By applying these two constraints, the sampled views can be effectively refined. In our experiments, we obtain the reference views by placing the model objects on a turn table. Therefore, the reference views are related to each other by a rotation about the y-axis. In practice, the rotation matrix could be estimated using structure from motion techniques [12]. To check the rigidity constraints, we test whether the expressions on the left hand-side of (5) and (6) are less than a small threshold. Fig. 5 (bottom) shows some of the views obtained with the rigidity constraints enforced.

Table I summarizes the main steps of the view generation process with the rigidity constraints imposed. Compared to our previous work, there is one more modification; we have reduced the space of AFoVs parameters from eight to six by disregarding the translation parameters a_4 and b_4. This is possible by normalizing the groups such as their centroid lies at the origin. By applying the same normalization on the scene groups during recognition, the values of a_4 and b_4 become zero; therefore, they do not need to be considered when generating the space of views of an object. Step 5.2 is the same as before and its purpose is to

TABLE I
MAIN STEPS ILLUSTRATING HOW VALID VIEWS ARE GENERATED

```
1. Select reference views and model groups;
2. Normalize the groups such as their centroids lie at $(0, 0)$;
(to eliminate the translation AFoVs parameters $a_4$ and $b_4$)
3. Estimate the range of AFoVs parameters using SVD and IA;
4. Set a sampling step;
5. Sample the space of AFoVs parameters;
    for each sample value $a_1$ in the interval $[a_1^{min}, a_1^{max}]$
    for each sample value $a_2$ in the interval $[a_2^{min}, a_2^{max}]$
    for each sample value $a_3$ in the interval $[a_3^{min}, a_3^{max}]$
    for each sample value $b_1$ in the interval $[b_1^{min}, b_1^{max}]$
    for each sample value $b_2$ in the interval $[b_2^{min}, b_2^{max}]$
    for each sample value $b_3$ in the interval $[b_3^{min}, b_3^{max}]$
    {
        5.1 Generate the view predicted by the
            sampled parameters $(a_1, a_2, a_3, b_1, b_2, b_3)$;
        5.2. If the generated view lies within the unit square
        5.3 If rigidity constraints are satisfied
            Accept the view as a valid view;
    }
```

verify whether the generated views lie within the unit square as it is required by the normalization step discussed in Section III-C (i.e., novel views should be normalized such that p_x and p_y lie in [1]). In general, this constraint is not satisfied by certain sets of parameters due to the fact that interval solutions are not tight [25], [26].

B. Representing the Space of Views Using Indexing

Coupling AFoVs with indexing is critical in making AFoVs more practical for recognition. Indexing is a mechanism which, when provided with a key value, allows rapid access to some associated data. It is based on the idea of using *a priori* stored information about the model objects in order to quickly eliminate incompatible model-scene matches during recognition. As a result, only the most feasible matches are considered for verification, that is, the matches where the model features could have produced the scene features.

During a preprocessing step, groups of model features are used to index appropriate information in a data structure. The locations indexed are filled with entries containing references to the model objects and some additional information that later can be used to recover the transformation between the model and the scene. During recognition, image features are used to retrieve information from the data structure. The models listed in the indexed entries are collected into a list of candidate models and the most often indexed models are selected for verification.

We have employed indexing in the past to represent information about the views that an object can produce in an index table. As described in the previous section, the views that an object can produce are computed by sampling the space of AFoVs parameters. Given a novel view of an object, we use information stored in the index table to estimate the AFoVs parameters that predict the appearance of the object in the scene. Thus, instead of having to search the space of all possible appearances and explicitly reject invalid predictions through verification, indexing inverts the process so that only the most feasible predictions are considered. This step essentially bypasses the correspondence problem between the novel view and the references views.

Coupling AFoVs with indexing offers significant advantages, however, recognition performance depends on the number of views indexed. As a result, space requirements could be expected to be high even for a moderate number of models. To take advantage of efficient hypothesis generation based on indexing while keeping storage requirements low, we propose indexing only a *sparse* number of views per object. The goal is to generate rough hypothetical matches very fast during recognition. Further hypothesis validation would be required as described in the next section. Generating a sparse number of views for a given object is done the same way as before except that the space of AFoVs parameters is sampled coarser this time.

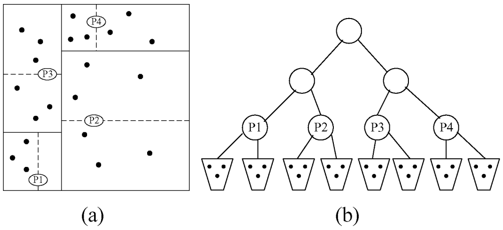

Fig. 6. 2-D k-d tree is a data structure which partitions the space using hyper-planes as shown in (a). The partitions are arranged hierarchically to form a tree as shown in (b).

In our past work, we used hashing to store information about the views of an object in a table. The same mechanism was used to retrieve the closest model views to a given novel view during recognition. Hashing, however, performs range search which would be very inappropriate to use when storing a sparse number of views like in our case. In contrast, it would be more appropriate to employ a more powerful indexing scheme, for example, a scheme that would be capable of performing nearest-neighbor search. Perhaps the most widely used algorithm for searching in multiple dimensions is a static space partitioning technique based on a k-dimensional binary search tree, called the k-d tree [30]. The k-d tree is a data structure (see Fig. 6) which partitions the space using hyper-planes. The partitions are arranged hierarchically to form a tree.

In its simplest form, a k-d tree is constructed as follows. A point in the database is chosen to be the root node. Points lying on one side of a hyperplane passing through the root node are added to the left child and points on the other side are added to the right child. This process is applied recursively on the left and right children until a small number of points remain. The resulting tree of hierarchically arranged hyper-planes induces a partition of space into hyper-rectangular regions, termed buckets, each containing a small number of points. The k-d tree can be used to search for nearest neighbors as follows. The k coordinates of a novel point are used to descend the tree to find the bucket which contains it. An exhaustive search is performed to determine the closest point within that bucket.

In a typical k-d tree [30], the partition of a hyperplane is perpendicular to the coordinate axes. In this work, we use the Sproull k-d tree [31] which is a radical refinement to the traditional one. The choice of the partition plane is not orthogonal or "coordinate based." Instead, it is chosen by computing the principal eigenvector of the covariance matrix of the points. Although k-trees can retrieve data fast assuming low-dimensional data, retrieval time increases significantly with an increase in the data dimensions [45]. This is not an issue in our work since the dimensionality of the data used for indexing is low (i.e., six).

C. Representing the Space of Views by Learning Shape Appearance

Although AFoVs allow us to generate the views that an object can produce efficiently, representing this information compactly would be important from a practical point of view. In this work, we propose combining indexing using statistical models of shape appearance, yielding an attractive scheme for model-based recognition. Specifically, the purpose of the indexing step is to generate rough hypothetical matches between the models and the scene fast. Due to the sparseness constraint and nearest-neighbor search at indexing, many hypothetical matches would be expected to be invalid. Evaluating potential matches without resorting to expensive verification (i.e., see Section IV-F) would be important in keeping recognition time low.

In this work, we propose evaluating the likelihood of the hypothetical matches generated by indexing using probabilistic models of shape appearance. In particular, the views that an object can produce form a manifold in a "universal" lower dimensional space which can be learned efficiently using mixture models and the EM algorithm. It should be mentioned that the structure of this manifold will not be linear in general due to the rigidity constraints imposed (see Section IV-A). Since we can generate a large number of views using AFoVs, we can build effective models of shape appearance by revealing and learning the true structure of this manifold. This is in contrast to similar approaches in the literature where a large number of images is required to ensure good results [56].

A mixture model is a type of density model which consists of a number of component functions, usually Gaussian. These component functions are combined to provide a multimodal density. Specifically, let the conditional density for the sample data ξ belonging to an object O be a mixture of M component densities

$$p(\xi|O) = \sum_{j=1}^{M} p_j(\xi)\pi_j \tag{7}$$

where a mixing parameter π_j corresponds to the prior probability that data ξ was generated by component j and where $\sum_{j=1}^{M} \pi_j = 1$. Each mixture component is a Gaussian with mean μ and covariance matrix Σ in a case of N-dimensional space, i.e.,

$$p_j(\xi) = \frac{1}{(2\pi)^{N/2}|\Sigma_j|^{1/2}} e^{-1/2(\xi-\mu_j)^T\Sigma_j^{-1}(\xi-\mu_j)}. \tag{8}$$

Mixture models provide great flexibility and precision in modelling the underlying statistics of sample data. They are able to smooth over gaps resulting from sparse sample data and provide tighter constraints in assigning object membership. In the past, they have been used to model the color appearance of objects for tracking and segmentation [79]. They have also been applied to learn the distribution of various object classes such as human faces [80].

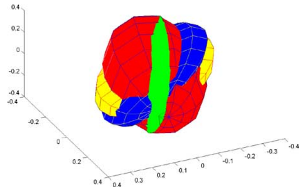

Fig. 7. Four manifolds, shown in different colors, each corresponding to the space of views that a different group of points can produce. As it can be observed, the manifolds overlap since although the groups are different, they might look quite the same if projected from specific viewpoints. RP was used to project the data to a three-dimensional space for visualization purposes.

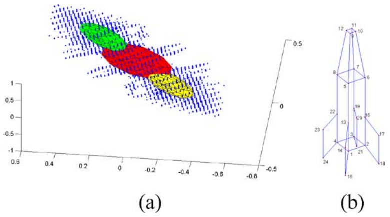

Fig. 8. Mixture model shown in (a) corresponds to a group of 8 point features (i.e., 9 to 16) from the artificial object shown in (b).

EM is a well established maximum likelihood estimation algorithm for fitting a mixture model to a set of training data [27], [81]. It is iterative with the mixture parameters being updated in each iteration. It has been shown that it monotonically increases the likelihood with each iteration, converging to a local maximum. However, EM suffers from singularities in the covariance matrix when the dimensionality of the data is high.

We have encountered similar problems in our experiments using large groups of point features. To avoid these problems, we have used RP to project the sampled views into a low-dimensional space before running the EM algorithm [28], [29]. The same RP was used for each object, we we refer to this common low-dimensional space as the "universal" object space. Fig. 7 shows an example of four manifolds, each corresponding to the space of views that a particular group of point features can produce. The four groups come from the set of real objects used in our experiments (see Section V-B).

RP has several important properties that allow EM to converge without problems, for example, it preserves separation between clusters and makes eccentric clusters more spherical [28], [29]. In our case, the space of views that groups of model points can produce is generated by involving six parameters (i.e., a_1, a_2, a_3, b_1, b_2, and b_3). Since we represent information only about the x-coordinates of the views (see Section IV-D), the effective dimensionality of our data is three. Therefore, we have used RP to reduce the dimensionality of the x-coordinates of the groups down to three. It should be noted that the EM algorithm requires providing the number M of mixture components. Here, we have determined the number of components automatically using mutual information [82]. Fig. 8 shows the mixture model obtained for a group of 8 points from an artificial objects (i.e., rocket).

D. Decoupling Image Coordinates

We have shown in our previous work that i) the process that generates the x coordinates of the sampled views is exactly the same to the process that generates the y coordinates as shown in (1) and ii) the interval solutions of the a_j and b_j parameters are exactly the same (see [17] and [18]). Therefore, similarly to our previous work [17], [18], we use only the x-coordinates of the sampled views to represent information in the k-d tree as well as to build the probabilistic models of shape appearance (i.e., Section IV-C).

This simplification offers significant advantages (e.g., we need to sample the space of the a_j parameters only), however, it adds a slight overhead to the hypothesis generation step since the k-d tree must be accessed twice per scene group. First, the x coordinates of the scene groups are used to form hypotheses predicting the a_j parameters, then, the y coordinates of the scene groups are used to form hypotheses predicting the b_j parameters. The likelihood of consistent hypotheses is then estimated by combining the likelihoods from the x and y coordinates of the groups using the statistical models of shape appearance (see Section IV-E).

E. Hypothesis Generation and Ranking

The hypothesis generation step starts by extracting groups of scene features from the scene and retrieving the closest model groups from the k-d tree. Since we do not assume knowledge of point correspondences between the models and the scene, we consider all possible circular shifts of the points in the scene groups. Assuming convex groups and that we traverse the points in the group in the same order, the maximum number of circular shifts is equal to the number of points in the group. In general, we can avoid considering all possible shifts by identifying certain starting points in the group (e.g., see Section V-B). For each query scene group, the number of nearest neighbors to be retrieved is controlled by a parameter (see Section V).

We test two simple constraints to evaluate each hypothesis before computing its likelihood. First, we test whether both the x and y coordinates of a scene group predict the same model group. Second, we test whether the model group predicted is similar enough to the scene group by requiring that the mean square error (MSE) between the groups is below a threshold (i.e., 0.3 pixels). Only hypotheses satisfying both constraints are considered for further processing. These hypotheses are ranked by computing their probability using the mixture models described in the previous subsection.

For each scene group, we compute two probabilities, one from the x-coordinates of the group and the other from the y-coordinates of the group The overall probability for a particular hypothesis j is then computed as follows:

$$p_o(\xi|O_j) = \frac{log(p(\xi_x|O_j) * p(\xi_y|O_j))}{log(\max_i(p(\xi_x|O_i)) * \max_i(p(\xi_y|O_i)))} \quad (9)$$

where $i, j = 1 \ldots, H$. H the number of hypotheses generated by the k-d tree search, $p(\xi_x|O_j)$ and $p(\xi_y|O_j)$ are the probabilities from the x- and y-coordinates of the current hypothesis, and $p_o(\xi|O_j)$ is the overall probability of the current hypothesis. It should be noted that each object is assumed to have the same probability of being present in the scene. Only hypotheses ranked high enough are considered for further verification with the most likely hypotheses verified first.

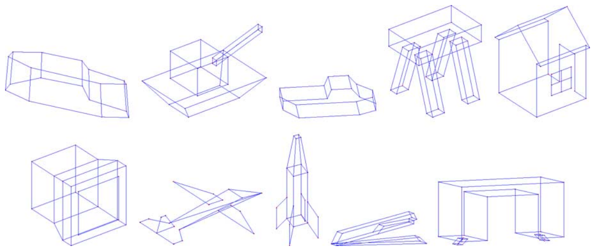

Fig. 9. Set of artificial objects used in our experiments.

F. Hypothesis Verification

Hypothesis verification takes advantage of hypothesis ranking to verify the most likely hypothesis first. For each hypothesis, the AFoVs parameters that predict the model in the test view are estimated by solving an over-constrained system of equations [i.e., (2)] using SVD. Then, the model is back-projected onto the scene to evaluate the quality of the prediction. Back-projecting the model onto the scene is extremely simple in the case of AFoVs and involves combining the reference views of the predicted model using the estimated AFoVs parameters.

V. Experimental Results

We describe below a set of experiments to evaluate and demonstrate the proposed approach. To enable robust feature extraction, we consider objects having sharp edges. Obviously, this assumption can be relaxed by using "interest" [6] operators as discussed in Section VII. Each object view is represented by a set of point features corresponding to intersections of line segments comprising the boundary of the object. To account for occlusion, we use subsets (i.e., groups) of point features as opposed to using all point features. In practice, we can select salient groups of point features, for example, corresponding to intersections of perceptually important groups of lines (e.g., convex groups of [35]). In this case, each point feature has a certain ordering in the group which can facilitate matching as discussed in Section IV-E.

A. Artificial 3-D Objects

First, we used a set of 10 artificial 3-D objects (see Fig. 9) to evaluate the performance of the proposed approach. Each model was represented by two reference views which were obtained by applying different orthographic projections on the 3-D models. For each model, we considered all possible groups having eight (8) point features (i.e., 22 groups on average for each model). In general, a value between 6 and 8 works well. Groups with less than 6 points might not have good discrimination power. On the other hand, groups with more than 8 points might not always be available and would be prone to occlusions. The space of views that the model objects can produce was computed according to the procedure shown in Table I. First, a coarse k-d tree was built by storing information about a sparse set of views. A total of 2,242 views were sampled and represented in the k-d tree. Then, a dense number of views was generated for each model group and its manifold was learned in a common random space using the EM algorithm.

TABLE II
PROBABILISTIC RANKING FOR THE CAR QUERY

Query Group	shift	Model Candidates	Un-normalized Likelihoods	Ranking
Car-g1	shift 0	Car-g1	(99.11, 34.07)	0.9992
	shift 6	Bench-g5	(29.59, 28.89)	0.8303
	shift 4	Car-g1	(99.77, 0.73)	0.5281
	shift 7	Car-g2	(0, 0),	0
Car-g2	shift 0	Car-g2	(164.65, 50.85)	1
	shift 4	Rocket-g2	(0.48, 0.22)	0
Tank-g1	shift 0	Tank-g1	(74.35, 38.73)	1
	shift 3	Monitor-g1	(18.54, 2.10)	0.4598
	shift 4	Monitor-g1	(0.00, 22.46)	0
	shift 4	Bench-g1	(0, 0)	0
Tank-g2	shift 0	Tank-g2	(227.30, 85.29)	1
Tank-g3	shift 0	Tank-g3	(1158.0, 905.8)	1
	shift 3	Truck-g1	(179.73,263.93)	0.7767
	shift 3	Rocket-g3	(39.43, 60.15)	0.5606
	shift 4	Rocket-g2	(43.72, 54.30)	0.5606
	shift 2	Car-g1	(22.49, 5.8191)	0.3516
	shift 6	Car-g1	(18.54, 4.22)	0.3145
	shift 7	House-g1	(0, 0)	0
Rocket-g1	shift 0	Rocket-g1	(539.1, 1922.9)	0.9428
	shift 4	Rocket-g1	(674.4, 3562.1)	1
Rocket-g2	shift 0	Rocket-g2	(32.66, 171.94)	1
	shift 4	Bench-g2	(0, 0)	0
Rocket-g3	shift 0	Rocket-g3	(21.45, 137.07)	1
	shift 4	Bench-g4	(0, 87.22)	0

The test views were generated by applying random orthographic projections on the 3-D models. We added 3 pixels random noise to the point features of the test views to make the experiments more realistic. We did not assume any knowledge of the point correspondences between model and scene groups; however, we did assume that point features have certain ordering in the group as discussed in Section IV-E. If noise affects the ordering, then recognition can not be accomplished using this group and a different group must be chosen. Assuming that there is no easy way to select the initial point feature in a group, we considered all possible circular shifts (i.e, eight (8) in this case) of point features when searching the k-d tree. For each query, we retrieved the 10 closest neighbors as this value worked well experimentally.

The query results for three of our models (i.e, car, tank and rocket) are shown in Table II, as well as their ranking, computed by the mixture models. The first column in Table II indicates the query group and the model it comes from, the second column indicates the circular shift applied on the scene group (i.e, "shift 0" always corresponds to the correct correspondence), and the third column shows the model candidates retrieved by the scene query and corresponding circular shift. It should be mentioned the third column shows only those hypotheses that satisfy the two simple constraints discussed in Section IV-E. The fourth column shows the un-normalized likelihoods computed for the x– and y–coordinates respectively while the overall probabilities, computed using (9), are shown in the last column. The overall probabilities indicate the level of confidence for each hypothesis and are used to rank them.

Only hypotheses ranked high enough (i.e., 0.9 or above) are considered for further verification. In this case, the parameters of the AFoVs are estimated accurately from the hypothetical match using a least squares approach such as SVD. Using the estimated AFoVs parameters, we then predict the appearance of the candidate model using (1) and compare it with the scene. Computing the MSE between the predictions and the scene provides a measure of similarity for deciding the presence of a candidate model in the scene. Fig. 10 shows the verification results for the hypotheses listed in Table II. We received extremely small MSE errors in all of our experiments using artificial data sets.

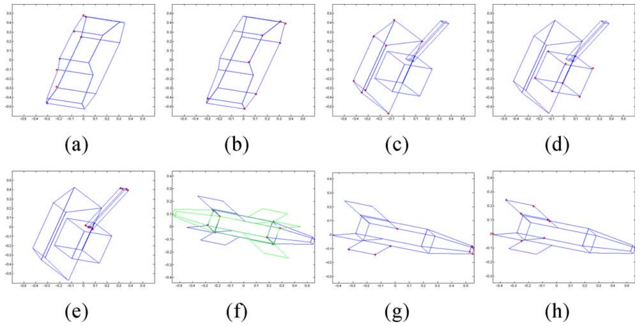

Fig. 10. Example verification results using different query groups: (a)–(b) two groups of from the car model, (c)–(e) three groups of the tank model, (f)–(h) three groups of the rocket model.

Table II shows that the hypotheses with the highest likelihoods were also the correct hypotheses in all cases except in one case (i.e, Rocket-g1). In that case, the first group of the rocket model was matched to the model assuming two different solutions due to symmetry, as shown in Fig. 10(f). We denote the test group of point features using "+," while the blue lines indicate

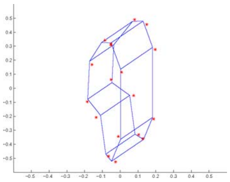

Fig. 11. Example of a query object affected by 5% noise.

TABLE III
ACCURACY OF k-d TREE QUERY FOR DIFFERENT NUMBER OF *SPARSE* VIEWS ($k = 10$)

Query Set	2242 (2.76%)	22852 (28.13%)	81236 (100%)
Set 1	100%	100%	100%
Set 2	86%	96%	96%
Set 3	96%	93%	100%
Accuracy	94%	96%	98.7%

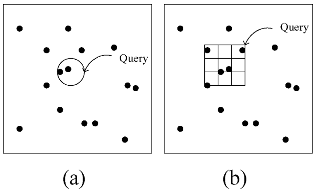

Fig. 12. Illustration of: (a) nearest neighbor search, (b) range search.

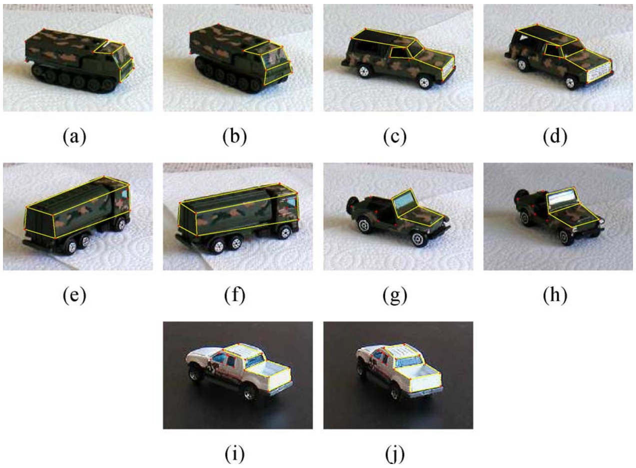

Fig. 13. Real objects used in our experiments, each represented using two reference views; the red dots show the features (i.e., points) used to represent the objects while the yellow lines show the corresponding groups of features. The points and lines have been drawn for visualization purposes only.

the predicted views. Such symmetric solutions can be resolved later during verification where more model point features are used.

To evaluate the effect of indexing a sparse number of views on recognition performance, we performed several experiments by indexing different numbers of views. Specifically, we created 3 test sets with each set containing 28 test views coming from all the models in our database. The test views were obtained by applying random orthographic projections on the 3-D models. Moreover, we added 5% random noise to the point coordinates of the test views. Fig. 11 shows a example of a test view. The blue lines indicate the test view without noise while the red "stars" show the locations of the noisy point features.

Three different k-d trees were generated by indexing different number of sampled views each time (i.e., 2242, 22852, and 81236). For each query, we retrieved the 10 nearest neighbors from the k-d tree. Table III shows the query accuracy for each of the three test sets where "accuracy" is defined as the probability that the correct model appears in the set of hypotheses retrieved from the k-d tree. Obviously, if the correct model does not appear in the list of hypotheses retrieved from the k-d tree, it can not be identified anyway during the subsequent steps of the algorithm. As the table illustrates, indexing more sampled views does not improve in general the accuracy significantly. In fact, we can index 97% less views by sacrificing accuracy only by 4.7%.

Comparing indexing based on hashing (i.e., used in our previous work) with k-d trees in terms hypothetical matches, hashing is expected to generate more hypotheses in general. Since hashing performs range search instead of nearest neighbor search, it recovers all points within a given distance from the query point, as shown in Fig. 12. The fact that we index a sparse number of views implies that we would have to use a fairly large neighborhood from the query point to ensure that the correct model is always retrieved. This, however, would increase the number of hypothetical matches. For example, let us consider a hash table of size 10×10. Assuming that we index 2,000 views, each hash table entry should store 20 entries on the average, assuming that the data is distributed uniformly on the table which might not always be easy [83]. Assuming that we search a 3 by 3 neighborhood for each query, there would be $3 * 3 * 20 = 180$ hypotheses generated for each query. In contrast, the number of hypothesis generated using k-d trees depends on the number of nearest neighbors retrieved (10–20 in our experiments) which should be expected to be much lower as shown in our experimental results.

B. Real 3-D Objects

In this section, we demonstrate the proposed approach using several real 3-D objects shown in Fig. 13. In these experiments, each object was represented using two reference views which are shown in Fig. 13. In each case, the second reference view was obtained by rotating the object about the y–axis by a small angle (e.g., 10–20 degrees). Having knowledge of the rotation between the reference views allow us to enforce the rigidity constraints discussed in Section IV-A.

In our experiments, we used groups containing six (6) point features. These groups were formed by combining two convex subgroups [35] of size four (4) each, having two point features in common. The groups extracted are shown by yellow lines in Fig. 13. To order the points in a given group during recognition, we choose the common points between the subgroups of size four as starting points and trace the rest of points counterclockwise. A sparse set of 3118 sampled views of the groups were represented in a k-d tree. The manifold of each group was then learned using a dense number of views and the EM algorithm.

TABLE IV
PROBABILISTIC RANKING FOR THE REAL DATA (PART I)

Query Group	Shift	Model Candidates	Un-normalized Likelihoods	Ranking
Fig.14(a)-g1	shift 0	tank1-g1	(3.34, 20.92)	0.9602
	shift 0	truck5-g3	(2.48, 14. 44)	0.8096
	permute 1	truck2-g1	(3.16, 24.94)	0.9876
Fig.14(a)-g2	shift 0	tank1-g2	(9.66, 28.98)	1
Fig.14(b)-g1	shift 0	truck1-g1	(22.09, 25.69)	0.9488
	shift 0	truck1-g3	(17.62, 33.06)	0.9527
	shift2	truck5-g2	(12.14, 36.17)	0.9105
Fig.14(b)-g2	shift 0	truck1-g2	(16.66, 24.42)	0.9511
	shift 0	truck5-g2	(12.15, 26.85)	0.9170
	shift 1	truck1-g3	(10.01, 33.07)	0.9194
	shift 1	truck3-g1	(14.01, 28.30)	0.9479
Fig.14(b)-g3	shift 0	truck1-g3	(6.70, 25.15)	0.8492
	shift 0	truck3-g1	(6.64, 29.46)	0.8740
	shift 2	tank1-g2	(14.21, 9.45)	0.8117
Fig.14(b)-g4	shift 0	truck1-g3	(11.23, 69.55)	0.9885
	shift 0	truck1-g4	(8.37, 50.70)	0.8979
	shift 0	truck2-g1	(5.33, 45.53)	0.8149
	shift 1	truck5-g3	(12.14, 49.98)	0.9510
Fig.14(c)-g1	shift 0	truck1-g3	(10.80, 10.62)	0.9912
	shift 0	truck2-g1	(8.27, 6.08)	0.8189
Fig.14(c)-g2	shift 0	truck2-g2	(3.38, 12.76)	0.8704
Fig.14(c)-g3	shift 0	truck2-g3	(5.14, 6.28)	0.8479
	shift 1	truck2-g1	(3.55, 9.41)	0.8563
Fig.14(d)-g1	shift 0	truck2-g2	(4.63, 17.47)	0.8691
	shift 0	truck3-g1	(6.49, 24.16)	1
Fig.14(e)-g1	shift 0	truck2-g3	(4.60, 21.87)	0.8201
	shift 0	truck5-g1	(12.66, 17.93)	0.9647
Fig.14(e)-g2	shift 0	truck5-g2	(13.58, 24.64)	0.9456
	shift 1	truck1-g4	(6.44, 34.41)	0.8762
Fig.14(e)-g3	shift 0	truck2-g3	(4.43, 13.55)	0.8258
	shift 0	truck5-g3	(7.77, 14.98)	0.9592

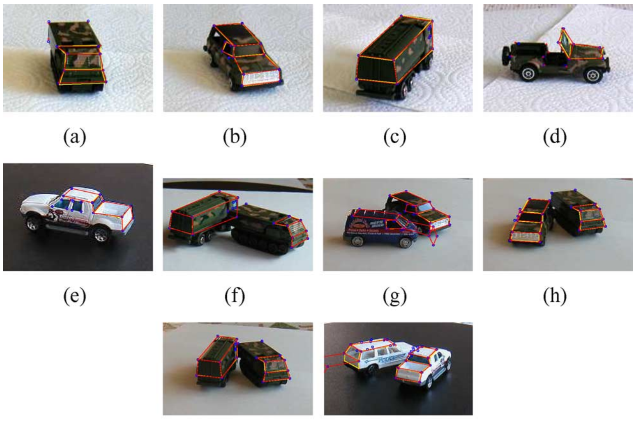

Fig. 14. Example recognition results illustrating the capability if the proposed methods to handle viewpoint changes. Of particular interest are the results shown in (g) and (j) as some of the input objects are not among the models stored in the database. It should be mentioned that the lines have been drawn on top of the objects to visualize the matches found (i.e., they are not given as input to the recognition algorithm).

Fig. 14 shows some of the test views used in our experiments. These scenes are simpler than those typically used to demonstrate intensity-based object recognition methods; however, the extraction of geometric features robustly and reliably in cluttered scenes is a challenging issue. Therefore, our emphasis in these experiments is to demonstrate the capability of the proposed methodology in handling viewpoint changes, which is the main strength of AFoVs, rather than dealing with clutter.

As before, we extract groups of point features from the scene and we use them to retrieve hypothetical matches from the k-d tree. For each query, we retrieve the 20 closest neighbors which was found to work well experimentally since real objects contain more noise. Hypotheses satisfying the two simple constraints mentioned in Section IV-E were then ranked using the mixture models of the model groups. The query results for each of the test groups shown in Fig. 14 with yellow lines are shown in Tables IV–VI. As before, the first column in each table indicates the query group and the model it comes from, the second column indicates the circular shift applied on the scene group (i.e, "shift 0" always corresponds to the correct correspondence), and the third column shows the model candidates retrieved by the scene group and the corresponding circular shift. As before, the third column of each table shows only the hypotheses satisfying the two simple constraints discussed in Section IV-E. The fourth column of each table shows the un-normalized likelihoods computed for the x– and y–coordinates respectively while the overall probabilities, computed using (9), are shown in the last column. The overall probabilities indicate the level of confidence for each hypothesis and are used to rank them.

Table VII lists the number of hypotheses generated from the k-d tree (i.e., satisfying the two simple constraints of

TABLE V
PROBABILISTIC RANKING FOR THE REAL DATA (PART II)

Query Group	Shift	Model Candidates	Un-normalized Likelihoods	Ranking
Fig.14(f)-g1	shift 0	tank1-g1	(10.28, 17.71)	0.8836
	shift 0	truck5-g3	(12.22, 13.36)	0.8651
	shift 1	truck2-g1	(5.82, 29.57)	0.8741
Fig.14(f)-g2	shift 0	tank1-g1	(31.00, 18.89)	1
	shift 0	truck5-g3	(19.26, 9.58)	0.8187
Fig.14(f)-g3	shift 0	truck1-g4	(3.71, 32.81)	0.8689
	shift 0	truck2-g1	(7.65, 20.80)	0.9175
Fig.14(g)-g1	shift 0	truck1-g1	(23.44, 44.94)	1
	shift 0	truck2-g1	(9.65, 34.14)	0.8329
Fig.14(g)-g2	shift 0	truck1-g2	(16.03, 26.39)	0.8627
	shift 1	truck1-g1	(19.95, 25.28)	0.8879
	shift 1	truck3-g1	(9.03, 55.49)	0.8869
Fig.14(g)-g3	shift 0	truck1-g3	(14.47,32.06)	0.9149
Fig.14(g)-g4	shift 0	truck1-g4	(7.77, 19.65)	0.9957
	shift 0	truck2-g2	(3.32, 18.64)	0.8168
	shift 2	truck5-g3	(5.26, 20.08)	0.9227
Fig.14(h)-g1	shift 0	truck1-g1	(15.03, 8.46)	0.8719
	shift 0	truck1-g3	(14.16, 17.24)	0.9891
Fig.14(h)-g2	shift 0	truck1-g2	(16.68, 10.38)	0.8053
	shift 1	truck3-g1	(4.67, 36.09)	0.8011
Fig.14(h)-g3	shift 0	truck1-g3	(3.07, 21.64)	1
	shift 2	truck5-g3	(2.87, 1.97)	0.8426
Fig.14(h)-g4	shift 0	tank1-g1	(7.97, 22.99)	0.8607
	shift 1	truck2-g3	(5.57, 26.22)	0.8232
	shift 1	truck5-g3	(3.00, 42.51)	0.8006
Fig.14(h)-g5	shift 0	tank1-g2	(7.58, 24.03)	1
	shift 1	truck3-g1	(4.65, 29.43)	0.9096

TABLE VI
PROBABILISTIC RANKING FOR THE REAL DATA (PART III)

Query Group	Shift	Model Candidates	Un-normalized Likelihoods	Ranking
Fig.14(i)-g1	shift 0	truck2-g1	(9.92, 3.1717)	0.9015
Fig.14(i)-g2	shift 0	truck1-g3	(0.72, 17.03)	0.8165
	shift 0	truck2-g1	(1.26, 12.47)	0.8992
	shift 0	truck2-g2	(3.41, 6.28)	1
	shift 2	truck2-g3	(2.63, 5.01)	0.8412
Fig.14(i)-g3	shift 0	truck2-g3	(6.97, 4.40)	0.8719
	shift 1	truck2-g1	(4.21, 7.28)	0.8715
Fig.14(i)-g4	shift 0	tank1-g1	(7.82, 21.16)	0.8948
	shift 1	truck2-g1	(4.64, 30.81)	0.8691
	shift 1	truck2-g3	(5.35, 24.85)	0.8563
Fig.14(i)-g5	shift 0	tank1-g2	(7.40, 22.64)	0.9813
	shift 1	truck3-g1	(5.16, 24.96)	0.9308
Fig.14(j)-g1	shift 0	truck5-g1	(12.14, 15.19)	0.8439
	shift 0	truck5-g2	(8.28, 28.63)	0.8847
	shift 1	truck1-g3	(16.90, 27.40)	0.9929
Fig.14(j)-g2	shift 0	truck5-g2	(13.56, 34.81)	0.9923
	shift 0	truck5-g3	(8.34, 26.80)	0.8718
	shift 1	truck1-g3	(7.50, 36.57)	0.9049
Fig.14(j)-g3	shift 0	truck5-g2	(8.71, 26.21)	1
	shift 0	truck5-g3	(6.84, 17.87)	0.8852
Fig.14(j)-g4	shift 0	truck5-g1	(14.66, 40.33)	0.9952
	shift 0	truck5-g3	(7.88, 37.47)	0.8869
Fig.14(j)-g5	shift 0	tank1-g2	(20.99, 17.86)	0.8844
	shift 1	truck3-g1	(35.65, 22.83)	1

Section IV-E) versus the hypotheses ranked high enough and considered for further verification. It is clear that hypothesis ranking reduces the number of hypotheses considered for verification substantially. On the average, the number of hypotheses generated from the k-d tree was eight (8), and the number of hypothesis ranked high enough was two (2).

Due to similarities among the groups considered, some of the correct models were not always ranked the highest. However, the correct model was always among the first two or three highest ranked hypotheses. For example, Fig. 7 shows the manifolds corresponding to four different groups. It can be seen that the manifolds overlap significantly with each other. That means

TABLE VII
NUMBER OF HYPOTHESES TO BE VERIFIED BASED ON HYPOTHESIS RANKING (RANKING THRESHOLD: 0.8)

Queries	No. of groups	No. of Hypoth. retrieved (k-d tree)	No. of Hypoth. to be verified
Fig. 14(a)	2	18	4
Fig. 14(b)	4	48	14
Fig. 14(c)	3	28	5
Fig. 14(d)	1	8	2
Fig. 14(e)	3	17	6
Fig. 14(f)	3	21	7
Fig. 14(g)	4	18	9
Fig. 14(h)	5	43	11
Fig. 14(i)	5	42	12
Fig. 14(j)	5	19	12
Average per group		8	2

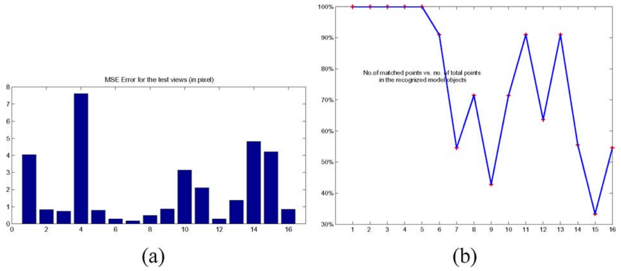

Fig. 15. (a) MSE error for each of the test groups shown in Fig. 14; (b) number of matched points over total number of model points.

that certain groups look similar from certain viewpoints. Therefore, it is difficult to distinguish two groups using the mixture models alone and further verification is required using mode model point features. In general, however, combining the k-d tree with the probabilistic models allow us to reject most invalid hypotheses quickly.

During verification, we estimate the parameters of the AFoVs using SVD, and compute the MSE between the predicted model view and the test view. The verification results can be seen in Fig. 14 where the yellow lines correspond to the scene groups and the red lines correspond to the predicted models. During this stage, we back-project all the model points onto the scene to provide more evidence about the presence of the model in the scene. The MSE error for each recognized object is always below eight (8) pixels (see Fig. 15).

It should be noted that some of the objects considered for testing were not among our models such as the cars in Fig. 14(g) and (j). Although, these objects had very similar local structures with the model objects (i.e., certain groups of point features extracted from the two objects were similar to groups of point features extracted from the models), overall similarity was pretty low as shown in Fig. 14(g) and (j).

By computing the ratio between the number of matched points between the model and the scene and the total number of points in the model, we can quantify the accuracy of each match more clearly. Fig. 14(b) shows these ratios for the each of the 15 test groups shown in Fig. 14(a). The two minima of the curve shown in Fig. 14(b) correspond to the unknown objects.

VI. EXTENDING AFOVS TO PREDICT EMPIRICAL APPEARANCE

Employing AFoVs for 3-D object recognition represents a powerful framework for predicting geometrical appearance. However, using geometrical information only does not provide enough discrimination for objects having similar geometrical structure but probably different empirical appearance. To deal with this issue, we have amended the proposed system by integrating geometrical and empirical representations during hypothesis verification to improve discrimination power and robustness. This is a promising step towards making the AFoVs-based recognition framework more general and effective.

As discussed in Section III, methods based on geometric models are more efficient in segmenting objects from the scene and more robust to occlusion. However, they can only handle changes in geometrical (shape) appearance of the object. On the other hand, methods based on empirical models are more successful in handling the combined effects of shape, pose, reflection and illumination but have serious difficulties in segmenting the objects from the complex background and dealing with occlusion. Fig. 7 shows the geometric manifolds corresponding to the space of views that four rather simple objects can produce. As it can be observed, the manifolds overlap significantly with each other. That means that there is much similarity in terms of geometric features in different objects and that certain objects look similar from certain viewpoints. In such cases, it may still be possible to distinguish them using empirical appearance.

To develop a viable recognition framework using AFoVs, a more powerful model of appearance is required, entailing geometric based recognition for shape and empirical based recognition for surface details. Here, we demonstrate how to extend AFoVs to predict the empirical appearance of an object. The main idea is using geometric information to segment the objects from the scene and generate the hypotheses, and both geometric and empirical information for hypothesis verification. This gives rise to a hybrid recognition framework, allowing for more realistic predictions of object appearance than geometry alone, thereby improving the performance of the system.

A. Establish Dense Correspondences

For each group of corresponding points, we apply triangulation recursively to get dense correspondences. Note that, for

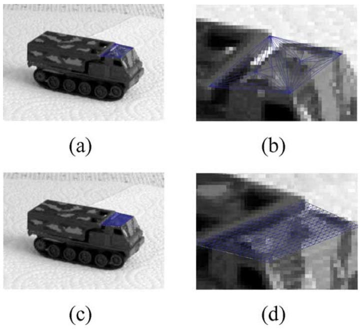

Fig. 16. Examples illustrating: (a), (b) the first triangulation scheme, and (c), (d) the second triangulation scheme. As it can be observed, the second scheme provides a more uniform triangulation which allows predicting intensity appearance more consistently.

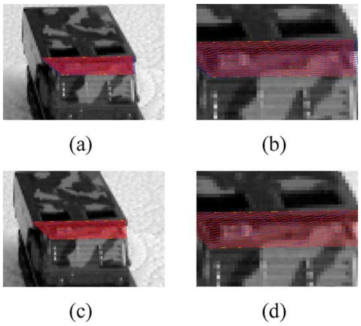

Fig. 17. Illustrating the effect of refining the AFoVs parameters: (a), (b) before refinement, (c), (d) after refinement. As it can be observed, refinement leads to more accurate predictions.

any set of points under orthographic projection, the projection of its centroid is the centroid of its projection. Two triangulation schemes could be possible based on different ways to partition a triangle. The first one divides the current triangle into three sub-triangles using the centroid of the current triangle. An example of this type of triangulation scheme can be seen in Fig. 16(a) and (b). The second ones divides the current triangle into four sub-triangles by considering the middle point of each side of the current triangle. An example can be seen in Fig. 16(c) and (d). As it can be seen from the figures, the second scheme produces more canonical triangles (i.e., triangles that are consistent both in shape and size). In both cases, we use the ratio of the resulting triangle areas as a criterion to stop the iterative triangulation process. Therefore, both schemes are scale independent. This is important in order to get the same number of triangles in different views of the same object.

B. Refine AFoVs' Parameters

The parameters of AFoVs can be refined using the dense correspondences obtained through triangulation as shown in Fig. 17. To see the effects of parameter refinement, we compute the AFoVs parameters in two ways, first using sparse correspondences (i.e., using only the points that comprise the groups in the reference views) and second, using dense correspondences (i.e., using all the points generated through triangulation). In each case, we predict the locations of the triangulated points in the scene using AFoVs (red lines) and compare them with the actual locations of the points in the scene (blue lines), obtained through explicit triangulation. Fig. 17(a) and (b) shows the results using the nonrefined AFoVs parameters while Fig. 17(c)and (d) shows the results using the refined AFoVs parameters. Obviously, the predictions using the refined AFoVs parameters are much more accurate (i.e., red and blue triangles overlap much better).

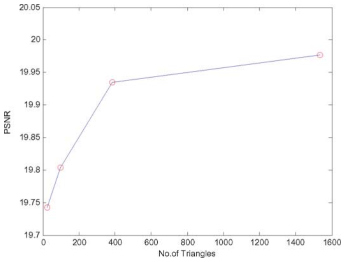

Fig. 18. PSNR versus number of triangles. As it can be observed, PSNR increases up to a certain point with using more triangles.

C. Peak Signal to Noise Ratio (PSNR)

Empirical appearance can be predicted from the reference images using a simple scheme. Specifically, the predicted coordinates can be determined using (1) as before. The intensity value at each predicted location can be determined by combining the intensity values at the corresponding locations in the reference images, for example, by averaging. The predicted image can then be compared with the query image. The quality of the prediction can be evaluated using the peak-signal-to-noise ratio (PSNR)

$$\mathrm{PSNR} = 10 \log \left[\frac{255 \times 255}{\frac{1}{N \times M} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} (u_{ij} = v_{ij})^2} \right] \tag{10}$$

where u_{ij} and v_{ij} are the actual and predicted pixel intensities at location (i, j). N and M are the width and height of the image. Fig. 18 shows how PSNR varies with the number of triangles. Obviously, increasing the number of triangles improves the quality of the prediction as expected. If the empirical appearance of the scene object is very different from the empirical appearance of the predicted object, then the PSNR will become very low, even though the two objects might have very similar geometric structure. Therefore, incorporating empirical appearance information could be very useful in distinguishing similar geometric structures having different empirical appearances. It should be mentioned that the above scheme for predicting empirical appearance is rather simple and assumes that the scene and reference images were captured under similar lighting conditions. Obviously, more sophisticated schemes would be necessary assuming arbitrary lighting conditions [84] or moment invariants [73].

D. Preliminary Results

Figs. 19 and 21 show some preliminary results. Fig. 19(a) and (b) shows two reference images of an object. Fig. 19(c) shows a test image of the same object, taken from a very different viewing angle. The yellow lines indicate the extracted groups of

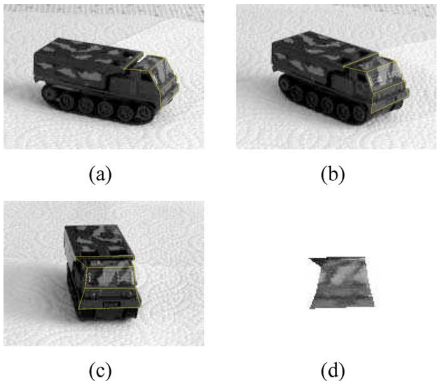

Fig. 19. Example results illustrating the idea of predicting intensity appearance using AFoVs: (a), (b) reference views, (c) input image, and (d) prediction results.

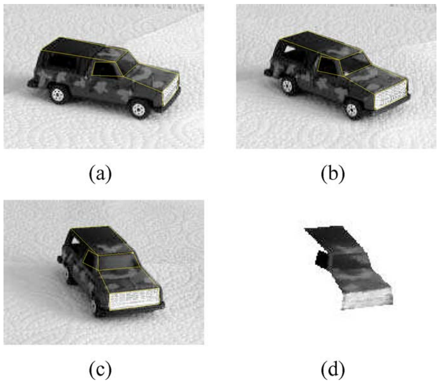

Fig. 20. Example results illustrating the idea of predicting intensity appearance using AFoVs: (a), (b) reference views, (c) input image, and (d) prediction results.

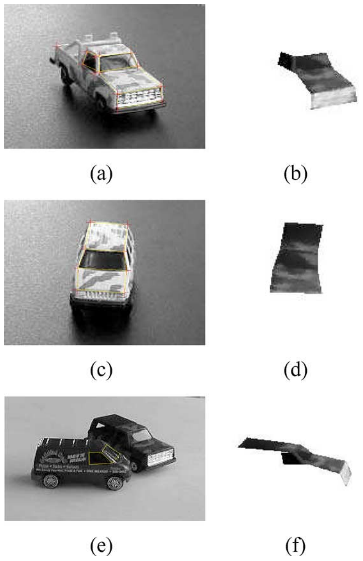

Fig. 21. Prediction results assuming the wrong models have been used to generate the predictions: (a)–(c) input images, (b)–(f) prediction results. As it can be observed, although an input object and a model might have similar geometric appearances, the predictions of intensity appearance would be quite different if the objects are different.

TABLE VIII
VERIFICATION RESULTS USING GEOMETRICAL AND EMPIRICAL APPEARANCE

Images	PSNR	Epirical-based Verification	Ground Truth
Fig.19(d)	16.1884	Good	Correct
Fig.20(d)	18.9688	Good	Correct
Fig.21(d)	5.0829	Bad	Wrong
Fig.21(f)	17.2585	Good	Wrong
Fig.21(b)	7.9193	Bad	Wrong

point features. Using the methodology outlined above, we have predicted the appearance of the groups in the scene. Fig. 19(d) shows the results. Fig. 20(a) and (b) shows two reference images of a jeep. Fig. 20(c) shows a test image while Fig. 19(d) shows the predicted appearance of the groups. In both examples, the predictions are quite reasonable.

Fig. 21(a) shows a vehicle that was not among our models. This is an interesting example since the vehicle shown in Fig. 21(a) has similar local geometric structure to one of our models [i.e., the jeep shown in Fig. 20(a) and (b)]. Due to the local geometrical similarity, the groups extracted match the "jeep" model, however, this hypothesis can be easily rejected by predicting the empirical appearance of the groups as shown in Fig. 21(b) and (c)–(f) shows additional examples. In the case of Fig. 21(e), both incorrect geometrical and empirical appearances have been predicted [see also Fig. 14(g)] which can make rejecting this hypothesis with much higher confidence. Please, note that some of the group points (i.e., yellow lines) are not shown clearly in Fig. 21(e) due to shrinking the figure for printing purposes.

The groups extracted from the test images in Figs. 19–21 [except the case shown in Fig. 21(e)] have all passed the geometrical appearance verification and predication by our initial system. However, further verification using empirical appearance yields different results, as shown in Table VIII. In this table, the second column shows the PSNR values, and the third column shows the verification results based on empirical appearance. The last column indicates ground truth information.

VII. Conclusion

We have presented an improved approach for 3-D object recognition using AFoVs. Compared to our earlier work, the new approach has been strengthened in several ways by (1) eliminating unrealistic views using rigidity constraints, (2) representing the space of views that an object can produce more compactly and efficiently using a two-stage scheme based on indexing and learning, and (3) improving verification by employing hypothesis ranking. We have also presented preliminary results illustrating how to extend the AFoVs framework in order to predict both the geometric and intensity empirical appearance of an object. In this context, we have have built a hybrid system that exploits geometric information to hypothesize the location of objects in the scene and both geometrical and intensity information to verify hypothetical matches.

The number of objects used in our experiments is comparable to the number of objects used in other, geometric-based, object recognition studies (e.g., [35], [38], [39], and [41]). However, compared to the size of the datasets used in recent, intensity-based, studies (e.g., [62], [70], and [72]), our dataset is very small. In general, geometric-based methods have difficulties dealing with large numbers of objects since they use much simpler features compared to the powerful descriptors used in intensity-based methods. On the other hand, geometric methods can handle viewpoint changes explicitly whether intensity-based methods consider viewpoint changes as another factor or variability that must be "learned." Moreover, intensity-based methods are not applicable for objects lacking texture. One way to demonstrate geometry-based methods on large datasets is by using more powerful features. However, we view geometry- and intensity-based methods as methods that complement each other rather than methods that compete with each other.

For future research, we plan to extend the proposed recognition framework in several ways. First of all, we plan to use more robust feature extraction methods as well as more powerful features for matching. One possibility is using more powerful perceptual grouping strategies such as the Iterative Multiscale Tensor Voting (IMTSV) scheme [85], [86] which has shown to tolerate significant amounts of noise and clutter. Alternatively, we plan to investigate state of art "interest" operators and local descriptors [6], [7], [71]. Using more powerful feature detectors would enable us to deal with more challenging scenes, both in terms of occlusion and clutter. Moreover, since AFoVs can handle viewpoint changes very efficiently, it would be more promising to handle viewpoint changes explicitly using AFoVs instead of treating them as an additional source of in-class variability. Related systems cannot handle a wide range of viewpoint changes [75]. Moreover, it might be possible to build simpler generative models by using AFoVs to "explain" viewpoint changes instead of modeling both intensity and geometric variations using a single model.

Second, we plan to investigate the issue of how to select a set of reference views that would allow recognition from any aspect. In the current implementation, although our system can handle novel views that are very different from the reference views, both novel and reference views must have been obtained from the same aspect. Therefore, the key question is how to select a small but sufficient number of reference views to allow recognizing novel views from any aspect. Past work on aspect graphs [49], [50] would be useful in this context although the objective of aspect graph theory is to represent an object in terms of its "characteristic" views. This might not be necessarily the same to finding the smallest number of views that would allow recognition from any aspect which is the objective of the AFoVs theory. They key issue is that some of the characteristic views of an object might be redundant. This is because the only requirement for recognizing a novel view is not how similar it is to the reference views but how much information the novel view has in common with the reference views. Therefore, it might be possible to choose a subset of an object's characteristic views for recognition purposes. In any case, methods to construct the aspect graph of an object [50] or view clustering and selection methods [87]–[89] would be useful in selecting a set of reference views covering all aspects.

Third, we plan to investigate more efficient ways to sample the space of views that an object can produce. In the current implementation, we sample the space of views uniformly which is quite inefficient. However, it would be more efficient to use an adaptive sampling step by taking into consideration that the appearance of an object might more or less stable from certain viewpoints. The idea is using a higher sampling rate in areas where object appearance changes fast and a lower sampling rate in areas where object appearance changes slowly. Past work on aspect graph theory [49], [50] would be very useful again in addressing this issue. For example, a typical approach for computing the aspect graph of an object involves tessellating the unit sphere and computing a view corresponding to each tessellation grid. Then, view clustering can be applied to group together similar views and identify viewpoints that produce topological similar views. The size of each cluster would provide some good indication about choosing the sampling step. In our case, the views corresponding to different viewpoint directions can be computed off-line using the estimated AFoVs parameters.

Fourth, we plan to employ more robust photometric features, such as the photometric moment invariants [73] to deal more effectively with illumination differences between novel views and the reference views. In our current implementation, we assume that novel views have been obtained under similar illumination conditions as the reference views which is not very realistic. Finally, we plan to extend the proposed recognition framework using AFoVs to category-based object recognition. Past work on recognizing object prototypes using AFoVs [78] would be useful in this context.

REFERENCES

[1] T. Binford, "Survey of model based image analysis systems," *Int. J. Robot. Res.*, vol. 1, no. 1, pp. 18–63, 1982.

[2] R. Chin and C. Dyer, "Model-based recognition in robot vision," *Comput. Surv.*, vol. 18, no. 1, pp. 67–108, 1986.

[3] P. Suetens, P. Fua, and A. Hanson, "Computational strategies for object recognition," *Comput. Surv.*, vol. 24, no. 1, pp. 5–61, 1992.

[4] J. Mundy and T. Saxena, "Towards the integration of geometric and appearance-based object recognition," *Lecture Notes Comput. Sci.*, vol. 1681, pp. 234–245, 1999.

[5] J. Mundy, "Object recognition in the geometric era: A retrospective," *Toward Category-Level Object Recognition*, J. Ponce, Ed. *et al.*, vol. 4170, pp. 2–28, 2006, LNCS.

[6] K. Mikolajczyk and C. Schmid, "Scale and affine invariant interest point derectors," *Int. J. Comput. Vis.*, vol. 60, no. 1, pp. 63–86, 2004.

[7] K. Mikolajczyk and C. Schmidt, "A performance evaluation of local descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, pp. 1615–1630, Oct. 2005.

[8] J. Ponce, M. Herbert, C. Schmid, and A. Zisserman, Eds., *Toward Category-Level Object Recognition* New York: Springer-Verlag, 2006, vol. 4170, LNCS.

[9] Z. Sun, G. Bebis, and R. Miller, "On-road vehicle detection using evolutionary gabor filter optimization," *IEEE Trans. Intell. Transport. Syst.*, vol. 6, pp. 125–137, 2005.

[10] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 4, pp. 594–611, Apr. 2006.

[11] B. Heisele, T. Serre, and T. Poggio, "A component-based framework for face detection and identification," *Int. J. Comput. Vis.*, vol. 74, no. 2, pp. 167–181, 2007.

[12] S. Ullman and R. Basri, "Recognition by linear combinations of models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 10, pp. 992–1005, Oct. 1991.

[13] O. Faugeras and L. Robert, "What can two images tell us about a third one ?," in *Proc. 3rd Eur. Conf. Computer Vision*, 1994, pp. 485–492.

[14] A. Shashua, "Algebraic functions for recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 8, pp. 779–789, Aug. 1995.

[15] S. Edelman, H. Bulthoff, and D. Weinshall, "Stimulus familiarity determines recognition strategy for novel 3D objects," Memo 1138, Mass. Inst. Technol., Cambridge 1989.

[16] H. Bulthoff and S. Edelman, "Psychophysical support for a 2D view interpolation theory for object recognition," in *Proc. Nat. Acad. Sci.*, 1992, vol. 89, pp. 60–64.

[17] G. Bebis, M. Georgiopoulos, M. Shah, and N. da Vitoria Lobo, "Algebraic functions of views for model-based object recognition," in *Proc. Int. Conf. Computer Vision*, 1998, pp. 634–639.

[18] G. Bebis, M. Georgiopoulos, M. Shah, and N. da Vitoria Lobo, "Indexing based on algebraic functions of views," *Comput. Vis. Image Understand.*, vol. 72, pp. 360–378, Dec. 1998.

[19] G. Bebis, S. Louis, Y. Varol, and A. Yfantis, "Genetic object recognition using combinations of views," *IEEE Trans. Evol. Comput.*, vol. 6, no. 4, pp. 132–146, Apr. 2002.

[20] G. Bebis, M. Georgiopoulos, N. V. Lobo, and M. Shah, "Learning affine transformations of the plane for model-based object recognition," in *Proc. Int. Conf. Pattern Recognition*, 1996, vol. IV, pp. 60–64.

[21] G. Bebis, M. Georgiopoulos, and S. Bhatia, "Learning orthographic transformations for object recognition," in *Proc. IEEE Int. Conf. Systems, Man, and Cybernetics*, Oct. 1997, vol. 4, pp. 3576–3581.

[22] G. Bebis, M. Georgiopoulos, N. V. Lobo, and M. Shah, "Learning affine transformations," *Pattern Recognit.*, vol. 32, pp. 1783–1799, 1999.

[23] G. Forsythe, M. Malcolm, and C. Moler*, Computer Methods for Mathematical Computations*. Englewood Cliffs, NJ: Prentice-Hall, 1977, ch. 9.

[24] W. Press *et al., Numerical Recipies in C: The Art of Scientific Programming*. Cambridge, U.K.: Cambridge Univ. Press, 1990.

[25] R. Moore*, Interval Analysis*. Englewood Cliffs, NJ: Prentice-Hall, 1966.

[26] E. Hansen and R. Smith, "Interval arithmetic in matrix computations: Part II," *SIAM J. Numer. Anal.*, vol. 4, no. 1, 1967.

[27] R. Redner and H. F. Walker, "Mixture densities, maximum likelihood and the em algorithm," *SIAM Rev.*, vol. 26, no. 2, pp. 195–239, 1984.

[28] S. Dasgupta, "Experiments with random projection," presented at the 16th Conf. Uncertainty in Artificial Intelligence, 2000.

[29] E. Bingham and H. Mannila, "Random projection in dimensionality reduction: application to image and text data," in *Proc. 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug. 2001, pp. 245–250.

[30] J. Friedman, J. L. Bentley, and R. A. Finkel, "An algorithm for finding best matches in logarithmic expected time," *ACM Trans. Math. Softw.*, vol. 3, pp. 209–226, Sep. 1977.

[31] R. Sproull, "Refinements to nearest-neighbor searching in k-dimensional trees," *Algorithmica*, vol. 6, pp. 579–589, 1991.

[32] W. Li, G. Bebis, and N. Bourbakis, "Integrating algebraic functions of views with indexing and learning for 3D object recognition," presented at the IEEE Workshop on Learning in Computer Vision and Pattern Recognition, Jun. 2004.

[33] R. Basri and D. Jacobs, "Recognition using region correspondences," *Int. J. Comput. Vis.*, vol. 25, no. 2, pp. 145–166, 1997.

[34] D. Jacobs and R. Basri, "3D to 2D pose determination with regions," *Int. J. Comput. Vis.*, vol. 34, no. 2/3, pp. 123–145, 1999.

[35] D. Jacobs, "Robust and efficient detection of salient convex groups," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 1, pp. 23–37, Jan. 1996.

[36] T. Binford and T. Levitt, "Model-based recognition of objects in complex scenes," in *Proc. Image Understanding Workshop*, 1996, pp. 89–100.

[37] E. Grimson and T. Lozano-Perez, "Localizing overlapping parts by searching the interpretation tree," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 9, no. 4, pp. 469–482, Apr. 1987.

[38] D. Huttenlocher and S. Ullman, "Recognizing solid objects by alignment with an image," *Int. J. Comput. Vis.*, vol. 5, no. 2, pp. 15–212, 1990.

[39] Y. Lamdan, J. Schwartz, and H. Wolfson, "Affine invariant model-based object recognition," *IEEE Trans. Robot. Autom.*, vol. 6, no. 10, pp. 578–589, Oct. 1990.

[40] T. Breuel, "Indexing of visual recognition from large model base," AI Memo 1108, AI Lab., Mass. Inst. Technol., Cambridge, 1990.

[41] C. Rothwell, A. Zisserman, D. Forsyth, and J. Mundy, "Planar object recognition using projective shape representation," *Int. J. Comput. Vis.*, vol. 16, pp. 57–99, 1995.

[42] A. Califano and R. Mohan, "Multidimensional indexing for recognizing visualshapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 4, pp. 373–392, Apr. 1994.

[43] C. Olsen, "Probabilistic indexing for object recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 5, pp. 518–522, May 1995.

[44] D. Jacobs, "Mathcing 3D models to 2D images," *Int. J. Comput. Vis.*, vol. 21, no. 1/2, pp. 123–153, 1997.

[45] J. Beis and D. Lowe, "Indexing without invariants in 3D object recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 10, pp. 1000–1015, Oct. 1999.

[46] Y. Lamdan, J. Schwartz, and H. Wolfson, "On recognition of 3D objects from 2D images," in *Proc. IEEE Int. Conf. Robotics and Automation*, 1988, pp. 1407–1413.

[47] D. Clemens and D. Jacobs, "Space and time bounds on indexing 3D models from 2D images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 10, pp. 1007–1017, Oct. 1991.

[48] J. Burns, R. Weiss, and E. Riseman, "View variation of point-set and line-segment features," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 1, pp. 51–68, Jan. 1993.

[49] J. Koenderink and A. van Doorn, "The singularities of the visual mapping," *Biol. Cybern.*, pp. 24–51, 1976.

[50] K. Bowyer and C. Dyer, "Aspect graphs: an introduction and survey of recent results," *Int. J. Imag. Syst. Technol.*, vol. 2, pp. 315–328, 1990.

[51] B. Schiele and J. Crowley, "Object recognition using multi-dimensional receptive field histograms," in *Proc. Eur. Conf. Computer Vision*, 1996, pp. 610–619.

[52] M. Swain and D. Ballard, "Color indexing," *Int. J. Comput. Vis.*, vol. 7, no. 1, pp. 11–32, 1991.

[53] D. Slater and G. Healey, "The illumination-invariant recognition of 3D objects using color invariants," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 2, pp. 206–210, Feb. 1996.

[54] I. Fodor, "A survey of dimension reduction techniques," Tech. Rep. UCRL, ID-148494, LLNL, 2002.

[55] M. Turk and A. Pentland, "Eigenfaces for recognition," *J. Cogn. Neurosci.*, vol. 3, no. 1, pp. 71–86, 1991.

[56] H. Murase, "Visual learning and recognition of 3-d objects from appearance," *Int. J. Comput. Vis.*, vol. 14, pp. 5–24, 1995.

[57] Z. Sun, G. Bebis, and R. Miller, "Object detection using feature subset selection," *Pattern Recognit.*, vol. 37, pp. 2165–2176, 2004.

[58] B. Heisele, I. Riskov, and C. Morgenstern, "Components for object detection and identification," in *Toward Category-Level Object Recognition*, J. Ponce, Ed. *et al.* New York: Springer-Verlag, 2006, vol. 4170, LNCS, pp. 225–237.

[59] A. Mohan, C. Papageorgiou, and T. Poggio, "Example-based object detection in images by components," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, pp. 349–361, 2001.

[60] P. Felzenszwalb and D. Huttenlocher, "Pictorial structures for object recognition," *Int. J. Comput. Vis.*, vol. 61, no. 6, pp. 55–79, 2005.

[61] C. Schmid and R. Mohr, "Local grayvalue invariants for image retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 530–534, Jul. 1997.

[62] D. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–100, 2004.

[63] H. Moravec, "Rover visual obstacle avoidance," in *Proc. Int. Joint Conf. Artificial Intelligence*, 1981, pp. 785–790.

[64] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. 4th Alvey Vision Conf.*, 1988, pp. 147–151.

[65] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," presented at the Eur. Conf. Computer Vision, 2004.

[66] M. Weber, M. Welling, and P. Perona, "Unsupervised learning of models for recognition," presented at the Eur. Conf. Computer Vision, 2000.

[67] R. Fergus, P. Perona, and A. Zisserman, "A visual category filter for google images," presented at the Eur. Conf. Computer Vision, 2004.

[68] A. Opelt, A. Pinz, and M. Fussenegger, "Generic object recognition with boosting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 3, pp. 416–431, Mar. 2006.

[69] T. Kadir and M. Brady, "Scale, saliency and image description," *Int. J. Comput. Vis.*, vol. 45, no. 2, pp. 83–105, 2001.

[70] K. Mikolajczyk and C. Schmidt, "Indexing based on scale invariant interest points," in *Proc. Computer Vision and Pattern Recognition Conf.*, 2001, pp. 525–531.

[71] K. Mikolajczyk, T. Tuytelaars, C. Schmidt, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool, "A comparison of affine region detectors," *Int. J. Comput. Vis.*, vol. 65, no. 1/2, pp. 43–72, 2005.

[72] Y. Ken and R. Sukthankar, "Pca-sift: A more distinctive representation for local image descriptors," in *Proc. Computer Vision and Pattern Recognition Conf.*, 2004, pp. 511–517.

[73] L. Van Gool, T. Moons, and D. Ungureanu, "Affine/photometric invariants for planar intensity patterns," in *Proc. Eur. Conf. Computer Vision*, 1996, pp. 642–651.

[74] F. Schaffalitzky and A. Zisserman, "Multi-view matching for unordered image sets," in *Proc. Eur. Conf. Computer Vision*, 2002, pp. 414–431.

[75] B. Leibe, A. Leonardis, and B. Schiele, "Combined object categorization and segmentation with an implicit shape model," presented at the Workshop on Statistical Learning in Computer Vision, 2004.

[76] R. Basri and S. Ullman, "The alignment of objects with smooth surfaces," *Comput. Vis., Graph., Image Process.: Image Understand.*, vol. 57, no. 3, pp. 331–345, 1993.

[77] R. Basri and E. Rivlin, "Localization and homing using combinations of model views," *Artif. Intell.*, vol. 78, pp. 327–354, 1995.

[78] R. Basri, "Recognition by prototypes," *Int. J. Comput. Vis.*, vol. 19, no. 2, pp. 147–167, 1996.

[79] S. McKenna, Y. Raja, and S. Gong, "Tracking colour objects using adaptive mixture models," *Image Vis. Comput.*, vol. 17, pp. 225–231, 1999.

[80] B. Moghaddam and A. Pentland, "Probabilistic visual learning for object representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 696–708, Jul. 1997.

[81] C. Bishop, *Neural Networks for Pattern Recognition*. Oxford, U.K.: Oxford Univ. Press, 1995.

[82] Z. R. Yang and M. Zwolinski, "Mutual information theory for adaptive mixture models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 4, pp. 396–403, Apr. 2001.

[83] G. Bebis, M. Georgiopoulos, and N. La Vitoria Lobo, "Using self-organizing maps to learn geometric hashing functions for model-based object recognition," *IEEE Trans. Neural Netw.*, vol. 9, pp. 560–570, 1998.

[84] P. Belhumeur and D. Kriegman, "What is the set of images of an object under all possible lighting conditions?," in *Proc. Computer Vision and Pattern Recognition Conf.*, 1996, pp. 270–278.

[85] L. Loss, G. Bebis, M. Nicolescu, and A. Skourikhine, "An automatic framework for figure-ground segmentation in cluttered backgrounds," in *Proc. Brit. Machine Vision Conf.*, 2007, vol. 1, pp. 202–211.

[86] L. Loss, G. Bebis, M. Nicolescu, and A. Skourikhine, "An iterative multi-scale tensor voting scheme for perceptual grouping of natural shapes in cluttered backgrounds," *Comput. Vis. Image Understand.*, 2008, to be published.

[87] T. Werner *et al.,* in *Proc. Int. Conf. Pattern Recognition*, 1996, pp. 73–77.

[88] R. Basri, D. Roth, and D. Jacobs, "Clustering appearances of 3D objects," in *Proc. Computer Vision and Pattern Recognition Conf.*, 1998, pp. 414–420.

[89] D. Weinshall and M. Werman, "On view likelihood and stability," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 2, pp. 97–108, Feb. 1997.

Wenjing Li (M'02) received the B.E. and M.E. degrees in computer science and engineering from the Hebei University of Technology, China, in 1995 and 1998, respectively, and the Ph.D. degree in electronic engineering from the Chinese University of Hong Kong in 2002.

From 2002 to 2004, she was a postdoctoral fellow in the Department of Computer Science, University of Nevada, Reno. During that time, she worked on the project "Automatic Target Recognition Using Algebraic Functions of Views" funded by the Office of Naval Research. She is now an image processing scientist and project leader at STI Medical Systems, Honolulu, HI. Her current research interests are best summarized as computer-aided detection/diagnosis and medical image analysis. She is also interested in pattern recognition, machine learning, and computer vision technologies. She has published over 30 technical papers in these areas and is a primary inventor of seven pending patents.

Dr. Li is a member of SPIE.

George Bebis (M'90) received the B.S. degree in mathematics and the M.S. degree in computer science from the University of Crete, Greece, in 1987 and 1991, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Central Florida, Orlando, in 1996.

Currently, he is an Associate Professor with the Department of Computer Science and Engineering, University of Nevada, Reno, and Director of the UNR Computer Vision Laboratory (CVL). His research interests include computer vision, image processing, pattern recognition, machine learning, and evolutionary computing. His research is currently funded by NSF, NASA, ONR, and Ford Motor Company.

Dr. Bebis is an Associate Editor of the *Machine Vision and Applications* journal and serves on the Editorial Board of the *Pattern Recognition* journal and the *International Journal on Artificial Intelligence Tools*. He has served on the program committees of various national and international conference and has organized and chaired several conference sessions. In 2002, he received the Lemelson Award for Innovation and Entrepreneurship. He is a member of the IAPR Educational Committee.

Nikolaos G. Bourbakis (F'96) received the B.S. degree in mathematics from the National University of Athens, Athens, Greece, and the Ph.D. degree in computer engineering and informatics from the University of Patras, Patras, Greece, in 1983.

He currently is the Associate Dean for Engineering Research, a Distinguished Professor of Information Technology, and the Director of the ATR Center at WSU, and a Research Professor at TUC. He has directed several research projects (Applied AI, Image Processing and Machine Vision, Visual Autonomous Navigation, Information Security, Bio-Informatics, Biomedical Engineering, Assistive Technologies) funded by government and industry, he has graduated 14 Ph.D. and 30 M.S. students, and he has published 300 papers in international refereed journals, conference proceedings, and book chapters. Previously, he was with SUNY, TUC, IBM, GMU, and UP.

Dr. Bourbakis is actively involved as an Associate Editor of several IEEE and international journals and Founder/General/Program Chair in numerous International IEEE Conferences (ICTAI, BIBE, IIS, INBS, NLP, IRS, JIS, RAT, etc). He is the EIC of the *Artificial and Biological Intelligence Tools International Journal* (WSP). He is an IEEE Computer Society Distinguished Speaker, a NSF University Research Programs Evaluator, and an IEEE Computer Society Golden Core Member. He has received several high prestigious awards, including: the IBM Author recognition Award 1991, the IEEE Computer Society Outstanding Contribution Award 1992, the IEEE Outstanding Paper Award ATC 1994, the IEEE Computer Society Technical Research Achievement Award 1998, the IEEE I&S Outstanding Leadership Award 1998, the IEEE ICTAI 10 years Research Contribution Award 1999, the PRS Best Selection Papers Recognition 1999, the IEEE BIBE Leadership Award 2003, the ASC Recognition Award 2005, the SETN Honorary Membership 2006, and the University of Patras Honorary Recognition Degree 2007.