Untangling Object-View Manifold for Multiview Recognition and Pose Estimation

Amr Bakry and Ahmed Elgammal

Department of Computer Science, Rutgers University
Piscataway, NJ, USA

Abstract. The problem of multi-view/view-invariant recognition remains one of the most fundamental challenges to the progress of the computer vision. In this paper we consider the problem of modeling the combined object-viewpoint manifold. The shape and appearance of an object in a given image is a function of its category, style within category, viewpoint, and several other factors. The visual manifold (in any chosen feature representation space) given all these variability collectively is very hard and even impossible to model. We propose an efficient computational framework that can untangle such a complex manifold, and achieve a model that separates a view-invariant category representation, from category-invariant pose representation. We outperform the state of the art in the three widely used multiview dataset, for both category recognition, and pose estimation.

1 Introduction

Visual object recognition is a challenging problem. This is mainly due to the large variations in appearance of objects within a given category, as well as variation of the appearance of an object due to viewpoint, illumination, occlusion, articulation, clutter, *etc.*. Impressive work have been done in the last decade on developing computer vision systems for generic object recognition. Research has spanned a wide spectrum of recognition-related issues, however, the problem of multi-view/view-invariant recognition remains one of the most fundamental challenges to the progress of the computer vision.

The problems of object classification from multi-view setting (view-invariant recognition) and pose recovery are coined together. Inspired by Marr's 3D object-centric doctrine [14], traditional 3D pose estimation algorithms often solved the recognition, detection, and pose estimation problems simultaneously (*e.g.* [7,11,13,24]), through 3D object representations, or through invariants. However, such models were limited in their ability to capture large within-class variability and were mainly focused on recognizing instances of objects. In the last two decades the field has shifted to study 2D representations based on local features and parts, with encoding the geometry loosely (*e.g.* pictorial structure like methods [6,5]) or without encoding the geometry at all (*e.g.* bag of words methods [29,25].) Encoding the geometry and the constraints imposed by objects' 3D structure are essential. Most research on generic object recognition

D. Fleet et al. (Eds.): ECCV 2014, Part IV, LNCS 8692, pp. 434–449, 2014.

bundle all viewpoints of a category into one representation; or learn view-specific classifiers from limited viewpoints, *e.g.* frontal cars, side view cars, rear cars, *etc.*.

In the context of multiview recognition and pose estimation, there is a growing recent interest in developing representations that captures 3D geometric constraints in a flexible way to handle the categorization problem. The work of Savarese and Fei-Fei [21,22] was pioneering in that direction. In [21,22] a part-based model was proposed where canonical parts are learned across different views, and a graph representation is used to model the object canonical parts. Successful recent approaches have proposed learning category-specific detection models that is able to estimate object pose (*e.g.* [15,18,23,19]). This has an adverse side-effect of not being scalable to a large number of categories while dealing with high within-class variations. Typically papers on this area focus primarily on evaluating the detection, and secondarily on evaluating pose estimation performance, and do not evaluate the categorization performance. In contrast to category-specific representations, in this paper we focus on developing a common representation for recognition and pose estimation, which can scale up to deal with a large number of classes.

In this paper we consider the problem of modeling the combined object-viewpoint manifold. The shape and appearance of an object in a given image is a function of its category, style within category, viewpoint, and several other factors. Given all these variability collectively, the visual manifold (in any chosen feature representation space) is very hard and even impossible to model. The main goal of this paper is to find a computational framework that can untangle such a complex manifold. In particular, we aim at untangling the object-viewpoint manifold, to achieve a model that separates a view-invariant category representation, from category-invariant pose representation.

This paper is builds over the model introduced in [30], which mainly proposed to model the category as a "style" variable over the view manifold of objects. This unconventional way is motivated by three observations: 1) low-dimensionality of the manifold of different views for a given object; 2) the prior knowledge of the view-manifold topology; 3) view manifolds of different objects (under the same view setting) share the same topology (ignoring degeneracy) but differ in their geometry, i.e, view manifolds of different objects are deformed version of each other. In contrast, considering the inter-class and the intra-class variability, even from a give view point, the resulting visual manifold is expected to be quite challenging to model, and can be of infinite dimensions. In [30] a computational framework was introduced that capitalizes on these observations, and models the deformation of different objects' view manifolds. The deformation space is then parameterized to reach a latent view-invariant category space, which is used in recognition. The overall model in [30] is a generative model, where hypotheses about the category and pose were used, within a sampling-based inference approach to minimize the reconstruction error, given a test image.

There is a mounting evidence of a feedforward computation in the brian [3] for the immediate categorization task. This motivated us to seek a forward model, that capitalizes on the same manifold structure observations used in [30], however avoids the challenging inference problem. The sampling-based inference, in [30], constitutes a major limitation to the computational framework. Even though the pose space is very low in dimensionality (one or two depending on the view setting), the view-invariant category latent space is high in dimensionality, which makes sampling not effective with no guarantee of convergence to the correct answer. In contrast, the current work presents several realizations, which leads to feed-forward computational models that do not require sampling-based inference.

The organization of the paper is as follows. Sec 2 describes the framework. Sec 3 describes how sampling-free inference can be achieved. Sec 4 illustrates experimental validation of the approach.

2 Framework

This section explains the intuition behind the the proposed framework and introduces the mathematical framework.

2.1 Framework Overview

Consider collections of images containing instances of different object classes and different views of each instance. The shape and appearance of an object in a given image is a function of its category, style within category, viewpoint, besides other factors that might be nuisances for recognition. Our discussion do not assume any specific feature representation of the input, we just assume that the images are vectors in some input space. The visual manifold given all these variability collectively is impossible to model. Let us first simplify the problem. Let us assume that the object is detected in the training images (so there is no 2D translation or in-plane rotation manifold). Let us also assume we are dealing with rigid objects (to be relaxed), and ignore the illumination variations (assume using an illumination invariant feature representation). Basically, we are left with variations due to category, within category, and viewpoint, *i.e.* , we are dealing with a combined *view-object manifold*.

The underlying principle in our framework is that multiple views of an object lie on an intrinsically low-dimensional manifold (*view manifold*) in the input space. The view manifolds of different objects are distributed in that descriptor space. To recover the category and pose of a test image we need to know which manifold this image belongs to, and what is the intrinsic coordinate of that image within that manifold. This basic view of object recognition and pose estimation is not new, and was used in the seminal work of Murse and Nayar [16]. In that work, PCA was used to achieve linear dimensionality reduction of the visual data, and the manifolds of different objects were represented as parameterized curves in the embedding space. However, dimensionality reduction techniques, whether linear or nonlinear, will just project the data to a lower dimension, and will not be able to achieve the desired untangled representation.

The main challenge is how to achieve an untangled representation of the visual manifold. The key is to utilize the low-dimensionality and known topology of the view manifold of individual objects. To explain the point, let us consider the

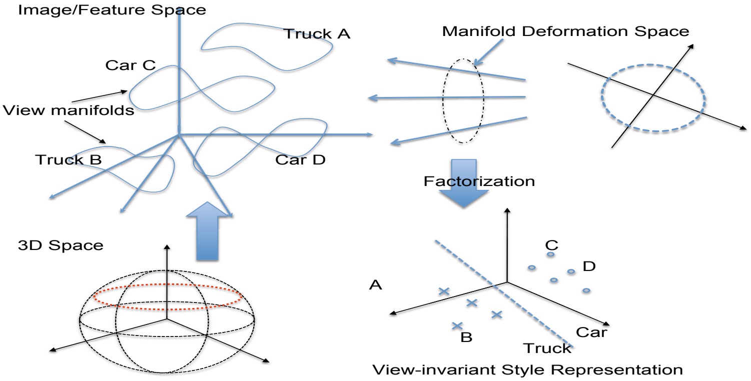

Fig. 1. Framework for untangling the view-object manifold.

simple case where the different views are obtained from a viewing circle, *e.g.* a camera looking at an object on a turntable. The view manifold of each object in this case is a one-dimensional closed manifold embedded in the input space. However, that simple closed curve deforms on the input space as a function of the object geometry and appearance. The visual manifold can be degenerate, for example, imaging a texture-less sphere from different views result in the same image, *i.e.* , the view manifold in this case is degenerate to a single-point.

Ignoring degeneracy, the view manifolds of all objects share the same topology but differ in geometry, and are all homeomorphic to each other. Therefore, capturing and parameterizing the deformation of a given object's view manifold tells us fundamental information about the object category and within category. *The deformation space* of these view manifolds captures a view-invariant signature of objects, analyzing such space provides a novel way to tackle the categorization and within-class parameterization. Therefore, a fundamental aspect in our framework, is that we use the view-manifold deformation as an invariant for categorization and modeling the within-class variations. If the views are obtained from a full or part of the view-sphere around the object, the resulting visual manifold should be a deformed sphere as well. In general, the dimensionality of the view manifold of an object is bounded by the dimensionality of viewing manifold (degrees of freedom imposed by the camera-object relative pose).

2.2 Manifold Parameterization

Here, we summarize the mathematical framework proposed in [30], which is the basic for our model, and highlight the challenges. The input are different views of each object instance, where the number views do not have to be same, and the views do not have to be aligned across objects.

Let us denote the view manifold of object instance s in the input space by $\mathcal{D}^s \subset \mathbb{R}^D$ where D is the dimensionality of the input space. Assuming that all manifolds $\mathcal{D}^s$ are not degenerate (we will discuss this issue shortly), then they

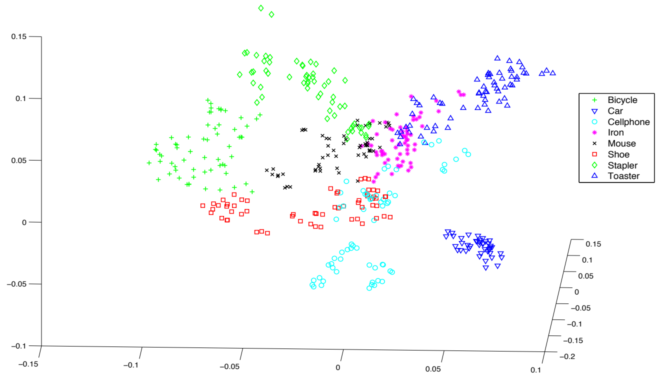

Fig. 2. Plotting of a three-dimensional unsupervised projection of the view-invariant style parameterization of 473 instances from 3DObjects dataset [21] (obtained from a training set of 3784 images from 8 views). Points of different categories show in different colors and point style. The plot clearly shows the separation between different objects, even in a three-dimensional projection.

are all topologically equivalent, and homeomorphic to each other[1]. Moreover, suppose we can achieve a common view manifold representation across all objects, denoted by $\mathcal{M} \subset \mathbb{R}^e$, in a Euclidean embedding space of dimensionality e. All manifolds $\mathcal{D}^s$ are also homeomorphic to $\mathcal{M}$. In fact all these manifold are homeomorphic to a unit circle in 2D for the case of a viewing circle, and a unit-sphere ($\mathbf{S^2}$) for the case of full view sphere.

We can achieve a parameterization of each manifold deformation by learning object-dependent regularized mapping functions $\gamma_s(\cdot) : \mathbb{R}^e \to \mathbb{R}^D$ that map from $\mathcal{M}$ to each $\mathcal{D}^s$. Given a Reproducing Kernel Helbert Space (RKHS) of functions and its corresponding kernel $K(\cdot, \cdot)$, from the representer theorem [8,20] it follows that such functions admit a representation in the form

$$\gamma_s(\mathbf{v}) = \boldsymbol{C}^s \cdot \psi(\mathbf{v}) \ , \tag{1}$$

where $\boldsymbol{C}^s$ is a $D \times N_\psi$ mapping coefficient matrix, and $\psi(\cdot) : \mathbb{R}^e \to \mathbb{R}^{N_\psi}$ is a nonlinear kernel map, i.e. $\psi(\mathbf{v}) = [K(\mathbf{v}, \mathbf{v}_1), \cdots, K(\mathbf{v}, \mathbf{v}_{N_\psi})]^T$, defined using a set basis of points $\{\mathbf{v}_i \in \mathbb{R}^e\}_{i=1 \cdots N_\psi}$ on $\mathcal{M}$ (The basis points can be arbitrary and does not need to correspond to actual data points [20]).

In the mapping in Eq. 1, the geometric deformation of manifold $\mathcal{D}^s$, from the common manifold $\mathcal{M}$, is encoded in the coefficient matrix $\boldsymbol{C}^s$. Therefore, the space of matrices $\mathbb{C} = \{\boldsymbol{C}^s\}$ encodes the variability between manifolds of different objects, and can be used to parameterize such manifolds. Notice that

[1] A function $f : X \to Y$ between two topological spaces is called a homeomorphism if it is a bijection, continuous, and its inverse is continuous. In our case the existence of the inverse is assumed but not required for computation, *i.e.*, we do not need the inverse for recovering pose. We mainly care about the mapping in a generative manner from $\mathcal{M}$ to $\mathcal{D}^s$.

the dimensionality of these matrices ($D \times N_\psi$) does not depend on the number of views available each object. We can parameterize the variability across different manifolds in a subspace in the space of coefficient matrices.

Of course the visual manifold can be degenerate or it can be self intersecting, because of the projection from 3D to 2D and lack of visual features, *e.g.* images of a textureless sphere. In such cases the homeomorphic assumption does not hold. The key to tackle this challenge is in learning the mapping in a generative manner from $\mathcal{M}$ to $\mathcal{D}^s$, not in the other direction. By enforcing the known non-degenerate topology on $\mathcal{M}$, the mapping from $\mathcal{M}$ to $\mathcal{D}^s$ still exists, still is a function, and still captures the manifold deformation. In such cases the recovery of object pose might be ambiguous and ill-posed. In fact, such degenerate cases can be detected by rank-analysis of the mapping matrix $\boldsymbol{C}^s$.

The space of manifold deformation functions, encoded by the coefficient matrices $\boldsymbol{C}^s$ is a high-dimensional rich space. Note that all the views of a given object is represented by a single point in that space, parameterizing the geometry of the view manifold of that object, and hence encoding information about its 3D geometry. By projecting the coefficient matrices to a low-dimensional latent space, we can reach a view-invariant representation. Such a representation can be achieved in an unsupervised way or in a supervised way using class labels; in a linear or nonlinear way. In the simplest case, using linear projection, we can achieve a generative model of the data in the form

$$z = \gamma(\mathbf{v}, \mathbf{s}) = \mathcal{A} \times_2 \mathbf{s} \times_3 \psi(\mathbf{v}), \tag{2}$$

where $\mathcal{A}$ is a third order tensor of dimensionality $D \times d \times N_\psi$, $\times_i$ is the mode-i tensor product as defined in [12].The variable $\mathbf{v}$ is a representation of the viewpoint that evolves around the common manifold $\mathcal{M}$, which is explicitly modeled. In this model, the variable $\mathbf{s} \in \mathbb{R}^d$ is a parameterization of manifold $\mathcal{D}^s$ that encodes the variation in category/instance of an object in a view-invariant way. We denote that space by “style” space. Therefore, that space can be used to train category classifiers in a view-invariant way. In this model, both the viewpoint and object/style latent representations, $\mathbf{v}$ and $\mathbf{s}$, are continuous.

Given features from a single test image, denoted by $\mathbf{z}$, recovering the pose and category reduces to an inference problem, where the goal is to find $\mathbf{s}^*$ and viewpoint $\mathbf{v}^*$ that minimize a reconstruction error, i.e.,

$$\arg\min_{\mathbf{s},\mathbf{v}} \|\mathbf{z} - \mathcal{A} \times_2 \mathbf{s} \times_3 \psi(\mathbf{v})\| \tag{3}$$

Once $\mathbf{s}^*$ is recovered, a category classifier trained on the style space can be used for categorization. There are different ways to do inference here, for example typical MCMC sampling, or gradient-based optimization can be used.

While the view variable is constrained to a 1D or 2D manifold for the cases of a viewing circle or a viewing sphere, respectively, inference in the style space is very challenging if its dimensionality is high. There is a fundamental tradeoff here: Lowering the dimensionality can lead to efficient inference, on the expense of losing the discriminative power of the space; in contrast, keeping the dimensions of the style space high will make the inference unlikely to converge. This is a fundamental limitation of the model, which we try to resolve by avoiding sampling all together, and investigating feed-forward solutions.

3 From Inference to Feed Forward

We propose a feedforward realization of the model that does not involve inference of the latent variables, yet still capitalizes on the advantages of the model. There are three motivations behind investigating such a feedforward realization of the model. First, biologically motivated, inspired by the extensive evidence of a cascade of feedforward computation in the brain for solving the immediate categorization problem [2], we would like to capitalize on the view-invariant property of the style space to achieve a realization of the model that can be implemented in a feedforward manner. Second, computationally, solving the inference problem in Eq 3 requires a sampling or a gradient-based search, which might not be desired for real-time applications. Third, from accuracy point of view, there is a tradeoff in choosing the dimensionality of the style space, (recall the style space is a achieved using linear or nonlinear projection of the high-dimensional manifold deformation space). Inference in high-dimensional spaces is notoriously not efficient nor effective. Reducing the dimensionality would lead to efficient inference, on the expense of losing discriminative power in categorization.

View-Invariant Category Manifolds: Let the set of view manifold parameterization matrices be $\{\mathbf{C}^i\}$, where $i = 1, \cdots, M$, is the index of the instances in the training data. Let us assume the case where the factorization in Eq 2 is achieved in an unsupervised way, by finding the subspace spanning these matrices. In that case, the factorization is achieve by SVD of the matrix

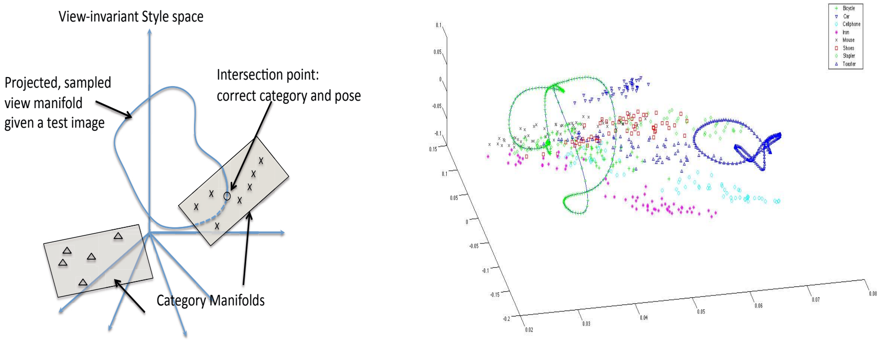
View-invariant Style space
Projected, sampled view manifold given a test image
Intersection point: correct category and pose
Category Manifolds
Bicycle
Car
Cellphone
Iron
Mouse
Shoes
Stapler
Toaster

Fig. 3. Left: Illustration of recovering pose and category by manifold intersection in a view-invariant space. Right: Example of Style-projected Inconsistent View Manifold for two images

$[\mathbf{c}_1 \cdots \mathbf{c}_M] = \mathbf{U\Sigma V}^\top$, where $\mathbf{c}_i$ is a vectorization of $\mathbf{C}^i$. The columns of $\mathbf{V}^\top$, corresponding to the styles of all training instances. Let us denote these style vectors by $\{\mathbf{s}_i \in \mathbb{R}^d\}_1^M$. Instances of the same category lie on a linear manifold (subspace) in the style space; we call that the *view-invariant category manifold*, and denote it by $\mathcal{C}^k$, where k denotes the category index. Such manifolds capture the within-category variability and also facilitate modeling other variabilities, hence relaxing the rigidity assumption. Figure 2 shows an example of the view-invariant space, with different category clearly separated. For the case where no dimensionality reduction take place, *i.e.* $d = M$, the style vectors for the instances of each category would provide orthonormal basis for that category's subspace.

Style-projected Inconsistent View Manifold: The key to achieve a feedforward realization is, again, in utilizing the low-dimensionality and known topology of the view manifold. Given a test image $\mathbf{z}$ we need to solve the inference problem in Eq 3 for the view $(\mathbf{v})$ and style $(\mathbf{s})$ variables. If we know the viewpoint, the problem reduces to solving a least-squares problems for the style variable, which can be achieved by solving the linear system $(\mathcal{A} \times_3 \psi(\mathbf{v}))\mathbf{s} = \mathbf{z}$. Suppose we have a sequence of images of the same object from different viewpoints, $\{\mathbf{z}_i\}_1^n$, and we know the corresponding latent view representation $\{\mathbf{v}_i\}_1^n$, the solutions for the linear system above for every pair $(\mathbf{z}_i, \mathbf{v}_i)$ should all coincide in a single point $\mathbf{s}^*$, since the style-space is view-invariant. However, we only have a single test image, and we do not know the corresponding latent view representation. Instead, if we sample the latent view manifold $\{\hat{\mathbf{v}}_i\}_1^n$ and solve the linear systems $(\mathcal{A} \times_3 \psi(\hat{\mathbf{v}}_\mathbf{i}))\hat{\mathbf{s}}_\mathbf{i} = \mathbf{z}$, we get a sequence of solutions $\{\hat{\mathbf{s}}_i\}$, which constitutes a projection of the view manifold into the style space, using inconsistent pairs $\{(\mathbf{z}, \hat{\mathbf{v}}_i)\}$. Such projection will also constitute a manifold, we call that *style-projected inconsistent view manifold*, denote it by $\hat{\mathcal{M}}_\mathbf{z}$, formally define it as

$$\hat{\mathcal{M}}_\mathbf{z} = \{\hat{\mathbf{s}_i} = \mathbf{V}_i^\dagger \mathbf{z}\}_1^n$$

where $\mathbf{V}_i = \mathcal{A} \times_3 \psi(\hat{\mathbf{v}}_\mathbf{i})$ is a $d \times D$ matrix, and † denotes the Moore-Penrose pseudoinverse. Note that each image will have its own inconsistent view manifold, hence the use of the subscript. Figure 3 shows examples of these manifolds for sample images.

Ideally the correct style $\mathbf{s}^*$ will be a point on that projected view manifold, corresponding to the solution for the pair $(\mathbf{z}, \mathbf{v}^*)$, where $\mathbf{v}^*$ is the closest sampled view to the correct viewpoint. Ideally also the correct style will be the intersection point between $\hat{\mathcal{M}}_\mathbf{z}$ and the correct category's manifold $\mathcal{C}^k$. Notice that finding the intersection point directly corresponds to finding the correct viewpoint as well. Figure 3 illustrates this process. Realistically, these manifolds might not intersect, especially since we are using sparse sampling of views. Moreover, the category manifolds are hard to model, given the sparse data available at training anyway. Therefore, we need to investigate different ways to achieve an approximate solution. The brute-force method would be a nearest neighbor search between $\{\hat{\mathbf{s}}_i\}_1^n$ and the set of style vectors of the all training instances.

Instead we can parameterize $\hat{\mathcal{M}}$ and/or $\mathcal{C}$ and use interpolation to find closest points between them.

Based on the concept explained above, in what follows we propose four different solutions to solve for pose, instance, and category, given image $\mathbf{z}$.

Manifold Intersection: Parametrizing the projected view manifold is easy since its topology and dimensionality is known. The category manifolds are linear in the style space. A simple way to find an approximate solution is to find the point on $\hat{\mathcal{M}}_{\mathbf{z}}$ closest to each category subspace, This can be achieved by

$$\underset{i,k}{\operatorname{argmin}} \|\mathbf{V}_i^{\dagger}\mathbf{z} - \mathbf{A}_k\mathbf{A}_k^{\top}\mathbf{V}_i^{\dagger}\mathbf{z}\| \tag{4}$$

where $\mathbf{A}_k$ is the matrix of orthonormal basis for category k. Unlike the optimization in Eq 3, where the search was over continuous spaces for style and view, here the problem reduces to discrete search over categories and sample views. The trade-off in choosing the style dimensionality is no-longer an issue here. The main trade-off here comes from sampling the viewpoint/pose space, however, in most pose estimation applications, only coarse estimation of the viewpoint is needed anyway. However, dense sampling might be necessary to obtain good approximation of the intersection with category manifold, which directly impact the categorization accuracy. This leads to the following three alternative solutions.

View-specific Projections: Given a test image $\mathbf{z}$, the correct style $\mathbf{s}^*$ will be a point on the projected view manifold for that image $\hat{\mathcal{M}}_{\mathbf{z}}$, which is most consistent with the correct view $\mathbf{v}^*$, *i.e.* minimizes the reconstruction error. The problem then reduces to minimizing

$$\|\mathbf{z} - \mathcal{A} \times_2 (\mathbf{V}_i^{\dagger}\mathbf{z}) \times_3 \psi(\hat{\mathbf{v}}_i)\|$$

Since $\mathbf{V}_i = \mathcal{A} \times_3 \psi(\hat{\mathbf{v}}_\mathbf{i})$, the above equation reduces to

$$i^* = \underset{i}{\operatorname{argmin}} \|\mathbf{z} - \mathbf{V}_i\mathbf{V}_i^{\dagger}\mathbf{z}\| \equiv \underset{i}{\operatorname{argmax}} \|\mathbf{V}_i\mathbf{V}_i^{\dagger}\mathbf{z}\| \tag{5}$$

Basically, this marginalizes the instance/category and provides a way to find the best viewpoint, among the sampled latent viewpoints, that is most consistent with test image. Once the best view, i^*, is found, the style can be directly obtained as $\mathbf{s}^* = \mathbf{V}_{i^*}^{\dagger}\mathbf{z}$. The geometric interpretation of this solution relies on noticing that the each of the matrices $\mathbf{V}_i\mathbf{V}_i^{\dagger}$ is an orthogonal projection operator into a view-dependent object-invariant subspace spanned by the columns of $\mathbf{V}_i$. Eq 5 is equivalent to finding the view-dependent subspace (spanned by the columns of $\mathbf{V}_i$) where $\mathbf{z}$ is closest to. In that sense, the images in the training data are used to learn these view-dependent object-invariant operators.

One important aspect that we should highlight is that the number of view-specific projector in this model is not restricted by the number of views in the training data. Since manifold parameterization is used to learn the view manifold for each instance, we can sample the view manifold at any arbitrary points $\{\hat{\mathbf{v}}_i\}_1^n$, and hence we can reach any desired number of view-specific projectors.

Instance-specific Projections: Using the same rational above, we can also obtain instance-specific view-invariant projectors by marginalizing out the view. Given a test image $\mathbf{z}$, and hypothesizing its corresponding style $\mathbf{s}$, an encoding of the view can be obtained by solving the linear system $(\mathcal{A} \times_2 \mathbf{s})\psi = \mathbf{z}$. Recall that $\psi(\mathbf{v})$ is a vector of nonlinear RBF kernels on $\mathbf{v}$, hence we can not obtain $\mathbf{v}$ directly, instead an encoding in an empirical kernel space. Given the set of style vectors $\{\mathbf{s}_i\}_1^M$ obtained from the instances in training data, let us define $D \times N_\psi$ instance-specific matrices $\{\mathbf{B}_i = \mathcal{A} \times_2 \mathbf{s}_i\}_1^M$. The solution for the view representation can be written as $\psi(\mathbf{v}) = \mathbf{B}_i^\dagger \mathbf{z}$. Substituting in the reconstruction error equation, we can reach

$$i^* = \underset{i}{\operatorname{argmin}} \|\mathbf{z} - \mathbf{B}_i \mathbf{B}_i^\dagger \mathbf{z}\| \equiv \underset{i}{\operatorname{argmax}} \|\mathbf{B}_i \mathbf{B}_i^\dagger \mathbf{z}\| \tag{6}$$

This marginalizes the viewpoint and provides a set of instance-specific view-invariant orthogonal projectors $\{\mathbf{B}_i \mathbf{B}_i^\dagger\}_1^M$. Eq 6 is equivalent to finding the instance-specific view-invariant subspace (spanned by the columns of $\mathbf{B}_i$) where $\mathbf{z}$ is closest to. Once the closest instance subspace is obtained, the pose can be recovered by finding the closest view in the empirical kernel map space

$$\underset{j}{\operatorname{argmin}} \|\mathbf{B}_{i*}^\dagger \mathbf{z} - \psi(\mathbf{v}_j)\| \tag{7}$$

Notice that, if the full dimensions of the style space is retained, *i.e.* d=M, the matrices $\mathbf{B}_i$'s reduce to the original coefficient matrices $\mathbf{C}^i$'s. In terms of scalability, the instance-specific solution will not scale well since one projection has to be computed for every instance in the training data, a problem that we will discuss next, to reach category-specific projections

Category-specific Projections: The scalability issues highlighted above motivates finding category-specific view-invariant projections, rather than instance-specific ones. The goal is to find a good category representation from the set of matrices $\mathcal{B}_k = \{\mathbf{B}_i | i \in \text{class } k\}$. Equivalently, each of these instance-specific matrices can be represented by an orthonormal basis matrix $\mathbf{U}_i \in \mathbb{R}^D \times N_\psi$. In other words, each instance corresponds to a point on a Grassmann manifold $G(D, N_\psi)$ (the subspace spanned by its column). This put into our disposal all the tools available for Grassmann manifold analysis [4] to obtain a good category-specific representations. For example k-means clustering on Grassmann manifold [28] can be used to achieve a representative category-specific subspace.

Given the set of instance-specific matrices $\mathcal{B}_k$ for the k-th category, we can reach a representation of that category's subspace by merging the subspaces of all its instances. Let $\mathbb{B}_k$ be a $D \times (N_\psi M_k)$ matrix constructed by stacking all the matrices in $\mathcal{B}_k$, where M_k is the number of instances of class k. The column span of this matrix is the union of all the column spans of the instance-specific matrices for this class. Therefore, a category-specific view-invariant projector can be achieved by $\mathbb{B}_k \mathbb{B}_k^\dagger = \mathbf{U}_k \mathbf{U}_k^\top$, where $\mathbb{B}_k = \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k$ is the truncated SVD of $\mathbb{B}_k$. Category and pose can be recovered in the same way as in Eq 6 and 7, by replacing the instance-specific matrices with the category-specific ones.

Discussion: At this point, it is important to contrast the solutions based on the view-specific, instance-specific, and category-specific projections. In terms of scalability, the instance-specific solution will not scale well since one projection has to be computed for every instance in the training data. In contrast the view-specific solution provides a more scalable solution, since the number of views can always be restricted. The view-specific projection also allows the use of discriminative classifiers, *e.g.* SVM in the style space, since it provides a solution for the $\mathbf{s}^*$, in contrast, the instance-specific and the category-specific just find the closest instance or category subspace. Another advantage of the view-specific solution, is that it allows expanding the model to add new objects, even with a single image from a single view point. This can be achieved by computing the corresponding style representation, as mentioned above. A reader might question, why this solution would yield a feedforward computational model. Notice that all projectors are learned offline during training. Finding the best point, whether using nearest neighbor search, or svm classifiers, is also a feedforward computation. Although we do not address detection in this paper, it can be achieved through a sliding window approach. However, the challenge is to learn a model for clutter. This can be achieved by projecting clutter training patches using the view-specific projectors, and learning a clutter/object classifier in the style space.

4 Experiments

We validated our framework using three multiview datasets: 3DObjects [21], U-Washigton-RGBD datasets [9], and Multi-View Car Dataset [17]. Since we target categorization, instance recognition and pose estimation, in all reported experiments we use ground-truth localizations of objects.

Results on 3DObjects

3DObjects dataset contains objects from 10 different categories: car, stapler, iron, shoe, monitor, computer mouse, head, bicycle, toaster and cellphone. Each object is imaged from 24 poses on a viewing sphere (8 azimuth angles × 3 zenith angles), and from 3 scales. We used the entire (all classes) 3DObjects dataset to evaluate the performance of the proposed framework on both object categorization and viewpoint estimation. Similar to [21,22] we test our model on an 8-category classification task (excluding heads and monitors). However, unlike [21,22], we do not exclude the farthest scale (which is more challenging). Figure 2 shows the learned view-invariant "style" vectors of each object instance, which clearly shows separation between different classes, even in a three-dimensional projection. Because of the limited number of zenith angles (3), we treat each zenith angle as a different viewing circle; *i.e.* all viewing manifolds are considered homeomorphic to a unit circle. To compare to published results, we used a train/test split similar to [21]; we randomly selected 7 object instances out of 10 in each category to build the proposed model, and the rest 3 instances for testing. We used HOG [1] features (20x20x31) as the input space representation. For parameterizing the view manifold, we used 8 RBF centers, (*i.e.* $N_\psi = 8$).

Table 1. 3DObjects: Category recognition and pose estimation results (%) for several configurations

Categorization Accuracy								Pose Estimation		
	View-specific				Instance-specific	Category-specific	Manifold intersection	View specific	Instance-specific	Manifold intersection
# v	SVM	5NN	7NN	S-Dists						
8		81.86	83.07	79.73	89.65	90.01	76.46	81.86	70.08	63.83
16		82.46	83.74	79.67	89.65	90.01	76.21	80.67	70.08	60.32
20	90.53	82.1	83.34	79.55	89.65	90.01	69.30	80.34	70.08	46.19

Table 1 shows the categorization and pose estimation accuracies using the different setting explained in Sec 3. Different rows show the results with different number of sampled views along the view manifold latent space, which is the number of view-specific projectors. For the case of view-specific projectors, after recovering the pose and the style, we evaluated four different classifiers on the style space: one-vs-all linear SVM, 5NN, 7NN, and the distance to the different category subspaces (similar to Eq 4 after choosing the best view, *i.e.* minimizing over categories only), denoted as S-Dists. For the view-specific case, the SVM classifier yields the best results. Interestingly, the three types of projectors gave very similar results ($\approx 90\%$). Notice, by construction, that changing the number of sampled views has no effect on the recognition accuracy of the instance-specific or the category-specific projectors. For the pose estimation, we estimate the azimuth angle. Given that the ground truth only has 8 azimuth viewpoints, for the cases where we sample more than 8 views, we approximate the result to the nearest 8 bin case. Not surprisingly, the view-specific projector gave the best results for pose estimation. Overall, the view-specific projector give the best results for both category recognition and pose estimation. Table 4-I shows comparison to some of the published results on this dataset[2].

In a machine with *2.3 GHz Intel Core i7 CPU and 16 GB 1600 MHz DDR3 memory*, each frame of this dataset takes about 4.6 microseconds to be processed (using MATLAB code), excluding the HOG feature extraction, for the instance-specific case.

Results on RGBD

We evaluated the different setting with the RGB-D dataset [9], which is the largest available multi-view dataset, consisting of 300 instances of 51 tabletop object categories. Each object is rotated on a turn-table and captured using an Xbox Kinect, providing synchronized RGB and depth images. For each object three pitch angles are used: 30,45,60 degrees. Training is done on using 30 and 60 degrees sequences and testing is done on the 45 degree sequences. We use HOG descriptors [1] in both RGB and depth. Unlike the 3DObject dataset, which include completely different objects, the RGB-D is challenging because it has large number of objects, with high appearance similarity among them. Also

[2] We mainly compared to approaches that perform categorization and pose estimation. We do not compare to approaches that perform category-pacific detection and pose estimation, since such a comparison will not be fair.

Table 2. RGB-D: Instance, Category, and Pose recognition results (%) using several configurations

	View-Specific					Instance-Specific-I			Instance-Specific-II		
	SVM (classes)		S-Dists (Instances)						(Height-mean)		
Features	Category	Pose	Instance	Category	Pose	Instance	Category	Pose	Instance	Category	Pose
Setting I											
RGB			60.36	76.95	72.51	66.48	85.66	72.24	80.10	94.84	76.63
RGB+D	88.31	73.23	63.80	82.36	73.23	66.19	89.62	71.93	78.63	95.77	75.44
Setting II											
RGB	83.23	72.69	66.24	82.49	74.13	68.24	86.71	73.13			
Depth	51.87	59.02	17.88	39.80	59.02	34.42	71.55	61.30	38.86	76.04	61.65
RGB+D			62.09	82.04	73.36				79.73	96.01	76.01

many objects are almost textureless with symmetric geometry, which makes the pose estimation ill-posed in such cases (*e.g.* an apple or an orange)

Table 2 shows the results over different configuration. We use two different setting for manifold parameterization: Setting I uses 11 RBF centers, while Setting II used 20 RBF centers. In both settings we samples 32 viewpoints on the view latent space to generate the view-specific projectors. The description of the different classifiers/metrics is similar to the case of 3D Objects. For the instance-specific projectors we compared two settings: in the first we used the two different heights for each instance to construct a different projector, while in the second setting, we combined the two heights to obtain one instance-specific projector (taking the average of the two style vectors for each instance). We report the instance, category, and pose estimation accuracies. The best results is achieved using the instance-specific projectors.

Table 3 summarizes the results, and compares to the state-of-the-art results [10,30]. Comparison to [30] is particularly important since our approach is based on the same formulation. The percentage evaluation metric used is the same as [10]. Following from [10], Average Pose (C) are computed only on test images whose categories were correctly classified. We report the results of our instance-specific projector-II from Table 2. We compared the results using different features (RGB and/or Depth). For all feature settings, our instance-specific projector outperforms both [10,30] for instance, category, and pose estimation.

Although our framework is based on [30], and it might be considered as an approximation of it, however we outperforms [30] in all settings. The reason, as we hypothesized in Sec 3, is that our approach avoids the sampling-based inference, which has a fundamental dimensionality-accuracy tradeoff, which we do not have. Moreover, our approach is much more efficient. Using Matlab code, on Dell PRECISION 490 with *Intel(R) Xeon (5160@ 3.00GHz 3.00 GHz) CPU - 8 GB memory and 64-bits Windows-7 os* machine (this configuration is far from powerful), we find that the average running time using Instance-Specific approach in this dataset is about 9.2 milliseconds. While the running time of the View-specific approach (with K-NN classifier) is about 0.279 microseconds on the same machine, which shows the power and speed of our framework. This is compared to less than two seconds per frame reported in [30], *i.e.* , our approach much faster and more accurate.

Table 3. Instance and Category recognition, and pose estimation accuracy (%) on the RGBD dataset. Compared to the state of the art [30] and [10].

Method	Instance	Category	Avg. Pose	Avg. Pose (C)
Ours (RGB)	80.10	94.84	76.63	79.78
[30] (RGB)	74.36	92.00	61.59	80.01
Ours (Depth)	38.86	76.04	61.65	70.79
[30] (Depth)	36.18	74.49	26.06	66.36
ours (RGB+Depth)	79.73	96.01	76.01	78.42
[30] (RGB+Depth)	74.79	93.10	61.57	80.01
[10] (RGB+Depth)	78.40	94.30	53.50	56.80

Table 4. Categorization and Pose estimation - comparison with state-of-the-art

Table 4-I Categorization - 3DObjects

	View-Spec	Instance-Spec	Zhang et al	Savarese et al
	Projectors	Projectors	[30]	[21]
Average	90.53%	89.56%	80.07%	75.65%
Bicycle	99.54%	99.54%	99.79%	81.00%
Car	99.31%	100.00%	99.03%	69.31%
Cellphone	98.15%	96.29%	66.74%	76.00%
Iron	86.11%	90.74%	75.78%	77.00%
Mouse	52.58%	44.60%	48.60%	86.14%
Shoe	94.07%	92.59%	81.70%	62.00%
Stapler	98.10%	96.21%	82.66%	77.00%
Toaster	98.15%	99.54%	86.24%	74.26%

Table 4-II Pose Estimation - Multiview Cars

Method	Split	16 views	8 views
Ozuysal *et al.*[17]	50% split	41.69	71.20
Teney and Piater [26]	50% split	78.10	79.70
Torki and Elgammal [27]	50% split	70.31	80.75
Zhang *et al.* [30]	50% split	87.77	88.48
proposed- 16 views	50% split	**93.94**	**94.13**
proposed- 20 views	50% split	**94.64**	**94.73**
proposed- 32 views	50% split	**94.84**	**94.84**
Torki and Elgammal [27]	leave one out	63.73	76.84
Zhang *et al.* [30]	leave one out	90.34	90.69
proposed -32 views	leave one out	**95.38**	**95.38**

Results on EPFL-CARS

The Multi-View Car Dataset [17], is a challenging dataset, which captures 20 rotating cars in an auto show. It provides finely discretized viewpoint groundtruth, that can be calculated using the time of capturing assuming a constant velocity. Table 4-II shows the view estimation results in comparison to the state of the art. All results are generated using view-specific projectors. We build the parameterizations using 15 Gaussian-RBF centers, and the input space is HOG features. We compared the results using 50% splits and leave-one-out splits, which are the typical splits reported in other papers, we report the average over different splits. More detailed experiments available at the supplementary material.

5 Conclusion

We presented a framework for untangling the object-viewpoint visual manifold. We described different approaches based on the framework which learn view-specific object-invariant, instance-specific view-invariant, or category-specific view-invariant projectors from the input space, and described how to solve for the pose and category in each case. Experiment on three multi-view dataset showed the potentials of our proposed approach, we outperform the reported state-of-the-art approaches for recognition and pose estimation on these datasets. Moreover, the approach is shown to be very efficient. The view-specific projectors are the most promising and most scalable approach. We did not target detection in this paper, however, detection can be achieved by running the approach in a sliding window manner, which is a subject of our future research.

References

1. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR (2005)
2. DiCarlo, J.J., Cox, D.D.: Untangling invariant object recognition. Trends in Cognitive Sciences 11(8), 333–341 (2007)
3. DiCarlo, J.J., Zoccolan, D., Rust, N.C.: How does the brain solve visual object recognition? Neuron 73(3), 415–434 (2012)
4. Edelman, A., Arias, T.A., Smith, S.T.: The geometry of algorithms with orthogonality constraints. SIAM J. Matrix Anal. Appl. 20(2), 303–353 (1998)
5. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI (2010)
6. Felzenszwalb, P.F., Huttenlocher, D.P.: Pictorial structures for object recognition. IJCV 61(1), 55–79 (2005)
7. Grimson, W., Lozano-Perez, T.: Recognition and localization of overlapping parts from sparse data in two and three dimensions. In: Proceedings of the1985 IEEE International Conference on Robotics and Automation, vol. 2, pp. 61–66. IEEE (1985)
8. Kimeldorf, G.S., Wahba, G.: A correspondence between bayesian estimation on stochastic processes and smoothing by splines. The Annals of Mathematical Statistics 41, 495–502 (1970)
9. Lai, K., Bo, L., Ren, X., Fox, D.: A large-scale hierarchical multi-view rgb-d object dataset. In: 2011 IEEE International Conference on Robotics and Automation (ICRA), pp. 1817–1824. IEEE (2011)
10. Lai, K., Bo, L., Ren, X., Fox, D.: A scalable tree-based approach for joint object and pose recognition. In: Twenty-Fifth Conference on Artificial Intelligence, AAAI (2011)
11. Lamdan, Y., Wolfson, H.: Geometric hashing: A general and efficient model-based recognition scheme (1988)
12. Lathauwer, L.D., de Moor, B., Vandewalle, J.: A multilinear singular value decomposiiton. SIAM Journal on Matrix Analysis and Applications 21(4), 1253–1278 (2000)
13. Lowe, D.G.: Three-dimensional object recognition from single two-dimensional images. Artificial Intelligence 31(3), 355–395 (1987)
14. Marr, D.: Vision: A computational investigation into the human representation and processing of visual information. W.H. Freeman (1982)
15. Mei, L., Liu, J., Hero, A., Savarese, S.: Robust object pose estimation via statistical manifold modeling. In: 2011 IEEE International Conference on Computer Vision (ICCV), pp. 967–974. IEEE (2011)
16. Murase, H., Nayar., S.: Visual learning and recognition of 3d objects from appearance. International Journal of Computer Vision 14, 5–24 (1995)
17. Ozuysal, M., Lepetit, V., Fua, P.: Pose estimation for category specific multiview object localization. In: CVPR (2009)
18. Payet, N., Todorovic, S.: From contours to 3d object detection and pose estimation. In: ICCV (2011)
19. Pepik, B., Stark, M., Gehler, P., Schiele, B.: Teaching 3d geometry to deformable part models. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3362–3369. IEEE (2012)

20. Poggio, T., Girosi, F.: Network for approximation and learning. Proceedings of the IEEE 78(9), 1481–1497 (1990)
21. Savarese, S., Fei-Fei, L.: 3d generic object categorization, localization and pose estimation. In: ICCV (2007)
22. Savarese, S., Fei-Fei, L.: View synthesis for recognizing unseen poses of object classes. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS, vol. 5304, pp. 602–615. Springer, Heidelberg (2008)
23. Schels, J., Liebelt, J., Lienhart, R.: Learning an object class representation on a continuous viewsphere. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3170–3177. IEEE (2012)
24. Shimshoni, I., Ponce, J.: Finite-resolution aspect graphs of polyhedral objects. IEEE Transactions on Pattern Analysis and Machine Intelligence 19(4), 315–327 (1997)
25. Sivic, J., Russell, B.C., Efros, A.A., Zisserman, A., Freeman, W.T.: Discovering objects and their location in images. In: ICCV (2005)
26. Teney, D., Piater, J.: Continuous pose estimation in 2d images at instance and category levels. In: 2013 International Conference on Computer and Robot Vision, pp. 121–127 (2013)
27. Torki, M., Elgammal, A.: Regression from local features for viewpoint and pose estimation. In: Proceedings of International Conference on Computer Vision, ICCV (2011)
28. Turaga, P., Veeraraghavan, A., Srivastava, A., Chellappa, R.: Statistical computations on grassmann and stiefel manifolds for image and video-based recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 33(11), 2273–2286 (2011)
29. Willamowski, J., Arregui, D., Csurka, G., Dance, C.R., Fan, L.: Categorizing nine visual classes using local appearance descriptors. In: IWLAVS (2004)
30. Zhang, H., El-Gaaly, T., Elgammal, A., Jiang, Z.: Joint object and pose recognition using homeomorphic manifold analysis. In: AAAI (2013)