

# 3D-Assisted Feature Synthesis for Novel Views of an Object

Hao Su\*, Fan Wang\*, Eric Yi, Leonidas Guibas  
Stanford University

## Abstract

Comparing two images from different views has been a long-standing challenging problem in computer vision, as visual features are not stable under large view point changes. In this paper, given a single input image of an object, we synthesize its features for other views, leveraging an existing modestly-sized 3D model collection of related but not identical objects. To accomplish this, we study the relationship of image patches between different views of the same object, seeking what we call surrogate patches — patches in one view whose feature content predicts well the features of a patch in another view. Based upon these surrogate relationships, we can create feature sets for all views of the latent object on a per patch basis, providing us an augmented multi-view representation of the object. We provide theoretical and empirical analysis of the feature synthesis process, and evaluate the augmented features in fine-grained image retrieval/recognition and instance retrieval tasks. Experimental results show that our synthesized features do enable view-independent comparison between images and perform significantly better than other traditional approaches in this respect.

## 1. Introduction

Comparing images of objects from different views is a classic and cornerstone task in computer vision. It is the core for many applications such as object instance recognition, image matching and retrieval, and object classification. In most scenarios, although the input is 2D images, the comparison between images is actually aimed at comparing the underlying 3D objects, regardless the different camera viewpoints from which they were captured. When the viewpoint difference is small, existing pipelines built upon robust local features [15, 7, 14] can perform the comparison well. However, these pipelines usually fail when the viewpoint difference is very large, since the content and relative locations of local features fail to persist.

Humans can do cross-view image comparisons very well, even if the viewpoint difference is large. Given a single image of an object, one can easily imagine the under-

\*Indicates equal contributions.

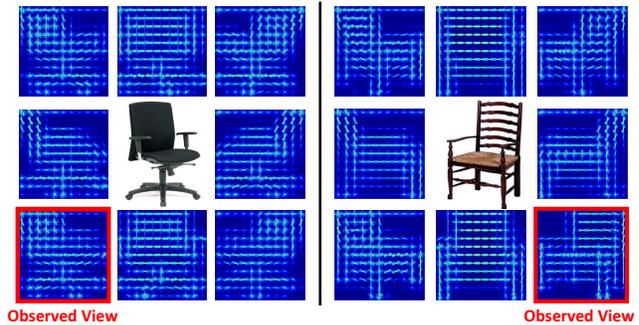


Figure 1: **Visualization of synthesized HoG features on 8 canonical views** Given the input image in the center, its HoG feature is shown in the red bounding box, and the synthesized features are visualized for the other view points.

lying 3D object, and infer the appearance in different views. This, however, is highly challenging for computers, due to two challenges: 1) estimating the 3D structure from a single image is physically under-determined: depth is missing for the observed parts, and all information is missing for the unseen parts; 2) synthesizing realistic details in novel views needs sophisticated geometric reasoning.

In this paper, we address the cross-view image comparison problem by *synthesizing features* of different views for an imaged object (Fig. 1), using a modestly-sized 3D model collection as a non-parametric prior. 3D models can provide strong prior information to help an algorithm “imagine” what the underlying 3D object should look like from novel views. Recently, more and more high-quality 3D models are available online, organized with category and geometric annotations such as alignment [2], making our proposed approach possible and effective. Moreover, we directly synthesize image features instead of synthesizing raw pixel values of novel view images. The motivation for doing so is that most computer vision techniques rely on image features as input. Furthermore, since features are more abstract forms of image appearance, they can be easier to transfer across views. Finally, by synthesizing features at a set of canonical viewpoints, we augment the original feature set and obtain a true multi-view representation of the object, effectively lifting the 2D image to 2-1/2D space [23, 5].

Our method is based upon two key observations. First,

features of an object from different views are correlated. This is because these images observe the same underlying 3D object, whose parts can be further correlated by 3D symmetries, repetitions, and other regularities. The nature of these intra-object correlations is typically consistent for objects in the same class. In fact, a remarkable feature of our approach is that it can exploit 3D symmetries of objects without any 3D analysis — by just learning these symmetries from patch observations in different views. Second, for similar objects, their features from the same view are correlated. In particular, the inter-object correlations are strong for features at the same spatial location. Therefore, we can approximate the features of an unknown 3D object via an existing collection of 3D models of similar objects.

**Contribution** We propose a method for synthesizing object image features from unobserved views by exploiting inter-shape and intra-shape correlations. Given the synthesized image features for novel views, we are then able to compare two images of the same or different objects by comparing their augmented multi-view features. The resulting distance is view-invariant and achieves much better performance on fine-grained image retrieval and classification tasks when compared with previous methods.

## 2. Related Work

**View-invariant Image Comparison** Many papers in literature attempt to achieve view-invariance by designing robust features [16, 3, 22]. In general, they quantize gradients into small number of bins to tolerate viewpoint change. This strategy, however, is widely known to fail in handling large viewpoint motions.

Spatial pooling is usually employed to allow the movement of local feature points as the viewpoint changes. Bag-of-visual words [6], Pictorial structure [8], spatial pyramid [13], and HoG [7] representations are the most popular ones. How a feature point would move w.r.t viewpoint change is not explicitly modeled in these methods, while we explicitly relate local regions of different views, enabling precise localized comparison.

Recently, there have also been evidence that generic descriptors learned by CNNs [12] are robust to certain viewpoint variation, demonstrated in image correspondence task [14] and retrieval task [18, 4]. As experiments (Sec 5.3) demonstrate, our feature augmentation scheme can further boost the performance of CNN features. [31] learns to predict novel views of faces using a fully connected neural network. It is unclear about its ability for generic object classes, which are more complicated in structure.

**Novel-view Synthesis** There are recent works to synthesize novel views of objects from a single image. Su et al. [25] achieves the goal by first reconstructing the 3D geometry.

Rematas et al. [19] synthesize novel views of objects by directly copying RGB pixel values from the original view. These approaches work well when the variation of 3D object structure is limited. However, they still lack the ability to recover detailed information when the object structure is complicated, and tend to suffer in unseen area.

In a different direction, by running a CNN classifier backwards, [1] is able to synthesize views of novel objects by using a manually specified input vector encoding the object and view, or to interpolate between multiple views of a given 3D model.

**3D Model Collections** Recently, we witness the emergence of several large-scale online 3D shape repositories, including the Trimble 3D warehouse (over 2.5M models in total), Turbosquid (300K models) and Yobi3D (1M models). By manual or geometry processing approaches, these publicly available 3D models can be organized by category and geometric annotations. ModelNet [30] organized over 130K 3D models from 600 categories, 10 categories of which are manually orientated. We believe that the rich information in these 3D models are helpful to understand the 3D nature of objects in images.

## 3. Problem Formulation and Method Overview

**Problem Input** Our input contains two parts:

1) an image of an object  $O$  with bounding box and known class label. With recent advances in image detection and classification [21], obtaining object label and bounding box has become much easier. All following steps are performed on a cropped image which only contains the object.

2) a collection of 3D shapes (CAD models) from the same class. All 3D shapes are orientation-aligned in the world coordinate system during a preprocessing step. Each shape is stored as a group of rendered images from the predefined list of viewpoints. Each rendered image is also cropped around the object. The view for object  $O$  in the input image is estimated to be one of the predefined viewpoints (§5.1). Local features such as HoG are extracted for each patch.

**Problem Output** The output is an *augmented version* of the original feature of the input image, consisting of one descriptor per view. Without loss of generality, the subproblem is: given the object observed from viewpoint  $v_0$ , estimate its features from another viewpoint  $v_1$ .

**Method Overview** The proposed framework is shown in Fig. 2. For a specific patch in the novel view (the query patch), we seek to find those patches on the observed view which can best predict it (see *Surrogate Region Discovery* in Fig. 2), and then learn how the features in those “surrogate” patches at the observed view can be best synthesized from the 3D model views (see *Estimation of Synthesis Parameters* in the figure). We finally transfer the same synthesis

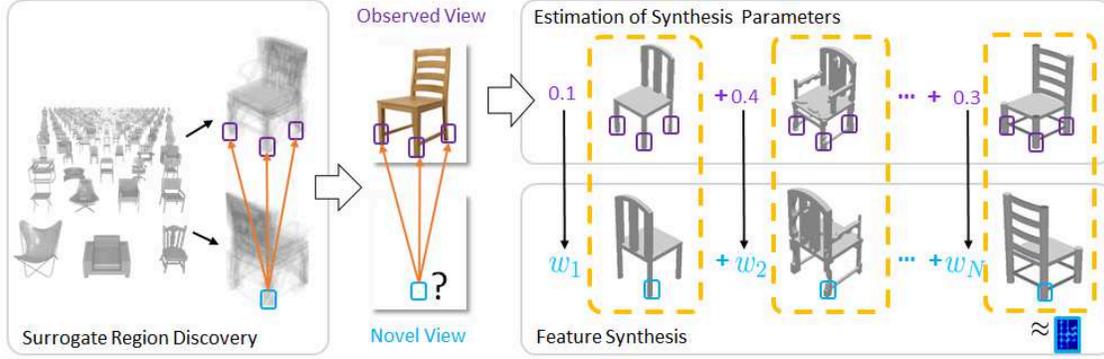


Figure 2: **Method overview.** Given a single object image, we synthesize image features for novel views of the latent underlying object. The synthesis is done patch-by-patch. To predict the feature in the blue patch of a new view, we first look for regions in the observed view which are most correlated with it — they are called the surrogate regions (purple patches). In a first stage, the surrogate regions are found by scanning the shape collection for such correlations that are robust across multiple shapes (**Surrogate Region Discovery**, §4.2). In a second stage, at the observed view, we learn how to reconstruct each surrogate region by a linear combination of the same region in the same view from all shapes in the shape collection (**Estimation of Synthesis Parameter**, §4.3). Finally, in the last stage, we transfer the linear combination coefficients back to the novel view to reconstruct the features in the blue patch, by linearly combining the features at the same patch on the novel view from all shapes in our collection (**Feature Synthesis**, §4.4).

method to the desired query patch (see *Feature Synthesis* in the figure) to generate the desired patch features. Please see the supplemental video for demonstration.

## 4. Novel View Image Feature Synthesis

### 4.1. Notation

The set of preselected viewpoints is indexed by  $\mathcal{V} = \{1, \dots, V\}$ . Each rendered image or the input real image is covered by  $G$  overlapping patches, indexed by  $\mathcal{G} = \{1, \dots, G\}$ . A patch-based feature set  $\mathbf{f} = [\mathbf{x}_1^T; \dots; \mathbf{x}_G^T] \in \mathbb{R}^{G \times D}$  is extracted for the image, where each  $\mathbf{x}_g \in \mathbb{R}^D$  is a feature vector for patch  $g$ . So the multi-view shape descriptor is represented by a tensor  $\mathbf{S} = [\mathbf{f}_1; \dots; \mathbf{f}_V] \in \mathbb{R}^{V \times G \times D}$ , in which each  $\mathbf{f}_v$  is a feature at view  $v$ . Finally, the 3D shape collection is denoted by  $\mathcal{S} = \{\mathbf{S}_1, \dots, \mathbf{S}_N\}$ , where  $\mathbf{S}_n$  denotes the multi-view descriptor of a shape  $n$ . For convenience, we further let  $\mathbf{S}_{n,v,g} \in \mathbb{R}^D$  denote the features of the  $g$ -th patch in the  $v$ -th view of the  $n$ -th shape.

### 4.2. Surrogate Region Discovery

To synthesize features from a novel view, we need to transfer information from the observed view, therefore, it is essential to understand and characterize the correlation of features at different locations of different views. Such correlations naturally exist because images from different views observe the same underlying 3D shape, whose parts may be further correlated by 3D symmetries, repetitions, and other factors. Fig. 3 shows some intuitive examples about patch relationships. Some patches in one view can well predict a certain patch in a novel view, because of the

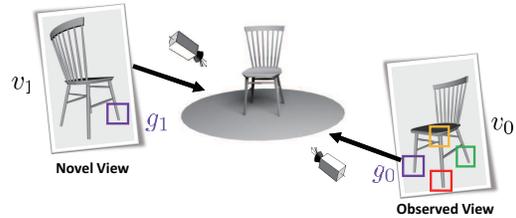


Figure 3: **Patch surrogate relationship** (§4.2). The surrogate relationship measures the predictability of patches across views ( $v_0$  and  $v_1$ ). In this example,  $g_0$  is a good surrogate of  $g_1$ , because  $g_0$  well predicts the appearance of  $g_1$ . The red patch and green patch in  $v_0$  can also well predict  $g_1$  because of symmetry and part membership (chair legs), respectively. On the other hand, the yellow patch at  $v_0$  will not be very helpful in determining  $g_1$ .

identity of underlying location in 3D, symmetry and part memberships. We call such patches as *surrogate patches*; the region they form is called a surrogate region  $\mathcal{R}$ .

This relationship between patches across views can possibly be inferred by analyzing shape geometry, but this is non-trivial and requires reliable object part segmentation, symmetry detection, etc. Therefore, we use a probabilistic framework to quantitatively measure such correlations, aiming to estimate the “surrogate suitability” of one image patch in one view to predict another patch in another view. We first introduce the concept of **perfect patch surrogate**:

**Definition 1.** Patch  $g_0$  at view  $v_0$  is a **perfect patch surrogate** for patch  $g_1$  at view  $v_1$  if  $\mathbf{S}_{i,v_0,g_0} = \mathbf{S}_{j,v_0,g_0}$  implies  $\mathbf{S}_{i,v_1,g_1} = \mathbf{S}_{j,v_1,g_1}$  for **any** shape pair  $\mathbf{S}_i$  and  $\mathbf{S}_j$ .

Intuitively, this definition means that, for a pair of 3D shapes, the similarity of patch  $g_0$  at view  $v_0$  implies the

similarity of patch  $g_1$  at view  $v_1$ . Usually patches cannot be perfect surrogates for each other, so we seek for a probabilistic version of Definition 1:

**Definition 2.** For a given patch  $g_1$  at  $v_1$ , the **surrogate suitability** of patch  $g_0$  at view  $v_0$  is defined as

$$\gamma(g_0; g_1) = \log P(\mathbf{S}_{i,v_1,g_1} = \mathbf{S}_{j,v_1,g_1} | \mathbf{S}_{i,v_0,g_0} = \mathbf{S}_{j,v_0,g_0}),$$

where  $P$  can be a probability for the discrete case and a density for the continuous case.

The quantity  $\gamma(g_0; g_1)$  is a measure of how suitable patch  $g_0$  is as a surrogate for patch  $g_1$ . Intuitively, larger  $\gamma(g_0; g_1)$  indicates a stronger correlation (Fig. 4). Therefore, the surrogate region  $\mathcal{R}(g_1)$  can either consist of the top  $k_p$  patches with highest  $\gamma(g_0; g_1)$ , or  $\mathcal{R}(g_1) = \{g_0 : \gamma(g_0; g_1) > \tau\}$ , where  $k_p$  or  $\tau$  is determined empirically.

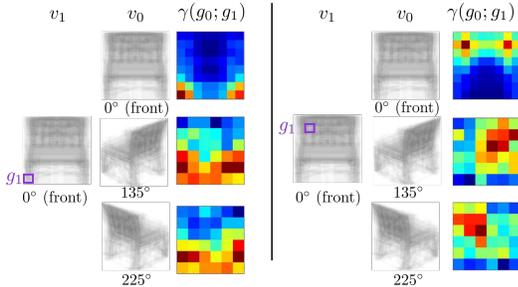


Figure 4: **Visualization of patch surrogate suitability.** Two examples of the surrogate suitability from  $g_1$  in  $v_1$  to patches in view  $v_0$ . Red means large  $\gamma$ . For example, in the left figure,  $g_1$  corresponds to the tip of right-front leg at  $v_1$  (front view). At the front view itself, the left-front and right-front leg tips have higher surrogate suitability for  $g_1$  because of symmetry; at the 225° view, the left-back, right-back and right-front leg tips have higher surrogate suitability because of symmetry and part membership.

#### 4.2.1 Estimation of Patch Surrogate Suitability

With the large-scale shape collection at hand, we adopt a learning based approach to estimate the (probabilistic) patch surrogate suitability in a data-driven manner.

Estimating  $\gamma(g_0; g_1)$  is a non-parametric density estimation problem. As image features are high-dimensional continuous variables, theoretical results indicate that the sample complexity for reliable estimation is very high and infeasible in practice. To overcome the difficulty, we quantize features into a vocabulary  $\mathcal{D}$  containing  $D$  visual words. For notation convenience, we denote the codeword of features  $\mathbf{S}_{i,v_0,g_0}$  by  $A_{g_0}^i$  and  $\mathbf{S}_{i,v_1,g_1}$  by  $A_{g_1}^i$ , then

$$\gamma(g_0; g_1) = \log P(A_{g_1}^i = A_{g_1}^j | A_{g_0}^i = A_{g_0}^j) \quad (1)$$

where  $P$  is the probability measure.

Estimating (1) by an empirical conditional distribution still requires a large number of samples. However, we show

that (1) can be cast as a Rényi entropy estimation problem. We can prove that the *optimal* sample complexity needed for estimating (1) is  $\Theta(D)$  (Theorem 2 in supplementary material). Roughly speaking, with  $N = \Theta(D)$  shapes, we can accurately estimate (1) with high probability. The proof also suggests an algorithm to estimate Eq (1) as below:

$$\hat{\gamma}(g_0; g_1) = \log \sum_{(A_{g_0}, A_{g_1}) \in \mathcal{D} \times \mathcal{D}} \hat{P}^2(A_{g_0}, A_{g_1}) - \log \sum_{A_{g_1} \in \mathcal{D}} \hat{P}^2(A_{g_1}).$$

Here, probabilities  $P^2(x)$  should be estimated by  $\hat{P}^2(x) = \frac{N_x(N_x-1)}{N^2}$ , where  $N_x$  is the total number of times value  $x$  appears in samples and  $N = \sum_x N_x$ .

#### 4.3. Estimation of Synthesis Parameters

The global shape space for multi-view representation is non-linear and high-dimensional. Our assumption, however, is that shapes in a local neighborhood can be well approximated by a locally linear and low-dimensional subspace [27]. This allows us to synthesize novel shapes through linear interpolation, so as to approximate the latent image object. Since the multi-view representation is actually a concatenation of features from all patches of all views, this local linearity not only holds for the whole shape, but also for each view of the shape, for each patch of the view, or even for a subset of patches of the view. In other words, features for the patches from the same location(s) on the same view of all shapes also lie in a locally linear subspace. The key point for capturing this relationship is to estimate appropriate coefficients for the interpolation, and we use an approach derived from locally linear embedding (LLE) methods [20].

For any patch  $g$  in view  $v$ , its feature is denoted as  $\mathbf{x}_{v,g} \in \mathbb{R}^D$ . We use  $\mathbf{S}_{:,v,g} \in \mathbb{R}^{D \times N}$  to denote the feature matrix collecting patch  $g$  of view  $v$  of all 3D shapes, then local linearity tells us that

$$\mathbf{x}_{v,g} \approx \mathbf{S}_{:,v,g} \mathbf{w}_{v,g}, \quad (2)$$

where  $\mathbf{w}_{v,g} \in \mathbb{R}^N$  is the reconstruction coefficient.

Given a surrogate region  $\mathcal{R}$  on the observed view, its features should be a linear combination of the same region across different 3D shapes. So  $\mathbf{w}_{v_0,\mathcal{R}}$  can be estimated by solving an Locally Linear Embedding (LLE) problem:

$$\begin{aligned} & \underset{\mathbf{w}_{v_0,\mathcal{R}}}{\text{minimize}} \quad \sum_{g_0 \in \mathcal{R}} \|\mathbf{x}_{v_0,g_0} - \mathbf{S}_{\mathcal{N},v_0,g_0} \mathbf{w}_{v_0,\mathcal{R}}\|^2, \\ & \text{subject to} \quad \mathbf{w}_{v_0,\mathcal{R}} \geq \mathbf{0}; \quad \mathbf{w}_{v_0,\mathcal{R}}^T \mathbf{1} = 1, \end{aligned} \quad (3)$$

where  $\mathcal{N}$  denotes the  $k$ -nearest shapes by comparing the rendered images on  $v_0$  with the input image, thus  $\mathbf{S}_{\mathcal{N},v_0,g_0} \in \mathbb{R}^{D \times k}$  and  $\mathbf{w}_{v_0,\mathcal{R}} \in \mathbb{R}^k$ .

Note that our reconstruction coefficient  $\mathbf{w}_{v_0,\mathcal{R}}$  is specific to the choice of view  $v_0$  and patch(es)  $\mathcal{R}$ , unlike previous locally linear reconstruction methods assuming uniform  $\mathbf{w}$  for the whole image descriptor [28].

#### 4.4. Feature Synthesis

Now that we have the synthesis coefficients estimated for  $\mathcal{R}$  on view  $v_0$ , we have to decide how to transfer it back to  $v_1$ , so that we can synthesize  $\mathbf{x}_{v_1, g_1}$  by apply the coefficients on features of  $g_1$  on  $v_1$  from all shapes.

We make the following assumption to connect the weight across views: if a patch  $g_0$  can surrogate  $g_1$  very well (with high  $\gamma(g_0; g_1)$ ), then their reconstruction weights are the same, i.e.  $\mathbf{w}_{v_0, g_0} \equiv \mathbf{w}_{v_1, g_1}$ . Intuitively, this assumption implies that, if you interpolate (the multi-view representation of) a set of shapes by linear reconstruction, then the interpolation coefficient estimated at one view is the same as the one estimated from another view. It can be derived from the surrogate relationship.

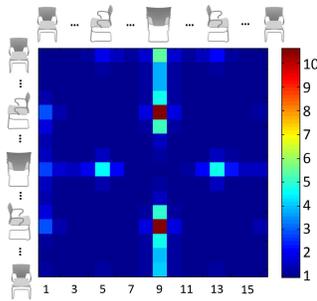


Figure 5: **Evaluation of weight transferability.** Smaller value means better transferability between corresponding two views. This matrix is asymmetric, since some views of an object may be more informative than others. For example, it is easy to guess the back view given a left-front view of a chair, since most chair parts are visible. However, it is difficult to do the opposite. Please read supplemental material for how to obtain this matrix.

Empirical verification of this assumption is shown in Fig. 5. The  $(j, i)$ -th element in the matrix shows the transferability from view  $v_i$  to  $v_j$ . It measures how close the synthesized feature on view  $v_j$  is to its ground truth version when using coefficients estimated on  $v_i$ . Each entry could range from 1 to the size of shape collection (5,057 in this experiment). The closer the value is to 1, the better the transferability is between  $v_i$  and  $v_j$ . The average value of the whole matrix is only 1.39, meaning that the weights transferred across views can reconstruct the features very well. Note that there are some entries indicating bad transferability between specific views. For example, view 5 and 9, which are the side view and back view respectively, cannot be transferred to each other very well because they share less common information.

Therefore,  $\mathbf{w}_{v_1, g_1}$  can be replaced by  $\mathbf{w}_{v_0, \mathcal{R}}$  if  $\mathcal{R}$  is the appropriate surrogate region on  $v_0$  for  $g_1$ . We can reconstruct the feature by  $\mathbf{x}_{v_1, g_1} = \mathbf{S}_{\mathcal{N}, v_1, g_1} \mathbf{w}_{v_0, \mathcal{R}}$ . Fig. 1 shows two examples of our synthesized image features.

#### 4.5. Method Summary

To fully exploit the information from 3D shape collection, we explore two kinds of relationships — *intra-shape* relationship that relates the novel view and the observed view (§4.2), and *inter-shape* relationship that relates the image and the shape collection (§4.3). To summarize, for each patch in the novel view, the intra-shape relationships allows us to find which patches in the observed view are its best surrogates, and the inter-shape relationships teach us how the feature of the new patch should be synthesized from those of its surrogates. In this way we can populate with features for all views of the latent object in our image, effectively creating its representation in our shape space.

### 5. Experiments

#### 5.1. Data Preparation

**Large-scale 3D Shape Dataset** We use 3D shapes from ShapeNet [26], a large-scale shape collection of 3D meshes. It contains 55 man-made object categories and 57,386 3D models in total. Models are categorized by WordNet structure, and those from each category are jointly aligned by orientation. The number of models per class varies from 20 (purse) to over 8000 (table).

**Shape Collection Preprocessing** Each shape is rendered from 16 predefined view points along a circle unless specified otherwise. The patch configuration is as below: each rendered image is resized to  $112 \times 112$  and partitioned into patches of  $32 \times 32$  which overlap with each other by 16 pixels, forming  $6 \times 6$  patches in total; local features are extracted for each patch. HoG is the default local features.

**Image Preprocessing** Object bounding box and class label are provided by R-CNN. A random forest classifier trained by the rendered images of aligned 3D models is used to estimate the view of the cropped object. Image features are extracted similarly as the rendered images.

#### 5.2. Applications

**Part-based Image Retrieval** Our approach can enable a new application of part-based image retrieval. The user can specify a region on the query image, and our approach can synthesize the features of related patches on novel views. The distance between images will only be evaluated on these patches instead of the whole images. Fig. 6 shows examples of part-based image retrieval. The rectangles on query images are the input specified by users. Although the algorithm can only see the provided patch on the view of query image, it returns images with similar appearance in the corresponding regions from other viewpoints. This



Figure 6: **Part-based retrieval results**

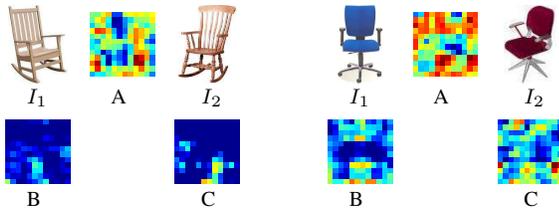


Figure 7: **Two examples of localized comparison of two images.** Heat map A visualizes the direct L2 distance of original HoG feature at different locations of two images. B and C visualize the localized difference by synthesizing HoG of  $I_2$  on the view of  $I_1$  (B), and synthesizing HoG of  $I_1$  on the view of  $I_2$  (C). Red color means larger difference.

part-based search can be useful in product search by image, allowing users to express preferences for product parts.

As there is no existing dataset to benchmark, we built a small-scale dataset of 100 chair images and conducted a user study with 5 people. In each experiment, a user draws an ROI on a query image and then marks the images with matching parts among the top-10 returned images. Each user performed 20 rounds of experiments. Our proposed part-based augmented HoG feature has an average accuracy of 67%, as opposed to 63% for global augmented HoG and 55% for vanilla HoG.

**Localized Cross-View Image Comparison** Traditional localized comparison between images is usually done by directly comparing image parts at the same location. This does not make sense when two images are of an object in different view points. Fig. 7 shows two examples of the localized comparison of two images. When two objects are similar but with different view points, directly comparing their features at each location yields a meaningless results as shown in heat map A of each example. If we synthesize the feature of one image at the same view point as the other image, the two objects are actually compared under the same view point, thus the feature distance at each location reflect the true difference of the two objects at each part.

**Fine-grained Image Retrieval on 55 Classes** We collect images of 55 classes with bounding boxes from ImageNet and verify their fine-grained labels within each class us-

ing AMT. Performance of fine-grained image retrieval is evaluated on these sets. Each image is taken as query once. All other images are ranked according to their distance to the query, and images having at least one fine-grained label overlapping with the query are regarded as correct. Precision-recall curves are generated, and the area-under-curve (AUC) is obtained to evaluate the retrieval performance. On average, the baseline L2 distance of HoG descriptor can achieve average AUC of 0.631, and our augmented HoG feature can achieve an AUC of **0.695**. Fig. 8 shows some examples of retrieval results for comparison.

**Fine-Grained Object Categorization** We also evaluate our method on fine-grained object categorization. For this experiment, we synthesize features at novel views for each training image. The newly synthesized features at each novel view are added to the training set separately. In this way, we augment the training set with more viewpoint diversity. We use the FGVC-aircraft dataset [17], which contains 10,000 images with 100 different aircraft model variants. The 3D airplane models are rendered at 200 viewpoints evenly distributed on the view sphere. We use the non-linear SVM on a  $\chi^2$  kernel and replicate the SPM feature setting in [17] to build the original feature, i.e. 600 k-means bag-of-visual words dictionary, multi-scale dense SIFT features, and  $1 \times 1$ ,  $2 \times 2$  spatial pyramid features. Our augmented feature is a view-invariant version of SPM feature. We also use bounding boxes predicted by R-CNN [9] and random forest for pose estimation (§5.1) on test data. Table 1 shows that our method significantly outperforms the baseline. Note that the baseline method in [17] does not use object bounding boxes in testing. To be fair, we also provide the baseline performance with bounding boxes provided.

**Instance Retrieval on Stanford Car** Our method can be naturally applied for instance-level recognition or retrieval. Since category-level class label for the object is required as input, most of existing instance-level data sets do not apply here because they mainly focused on instances of different classes, thus we create a new data set based on a fine-grained benchmark data set – Stanford Car [11]. Stanford Car contains car images classified into fine-grained categories defined by the car make, model and year. We randomly choose a subset of its categories, and the selected images are verified manually by AMT to see if they visually belong to the same instance. Besides the make, model and year information, two car images are regarded as the same instance if they also have the same color, texture, decoration (i.e. door exterior trim), and accessories (i.e. top rack), meaning that human cannot differentiate them without any outside information. In total, the created instance-level car data set contains 315 images in 20 instances.

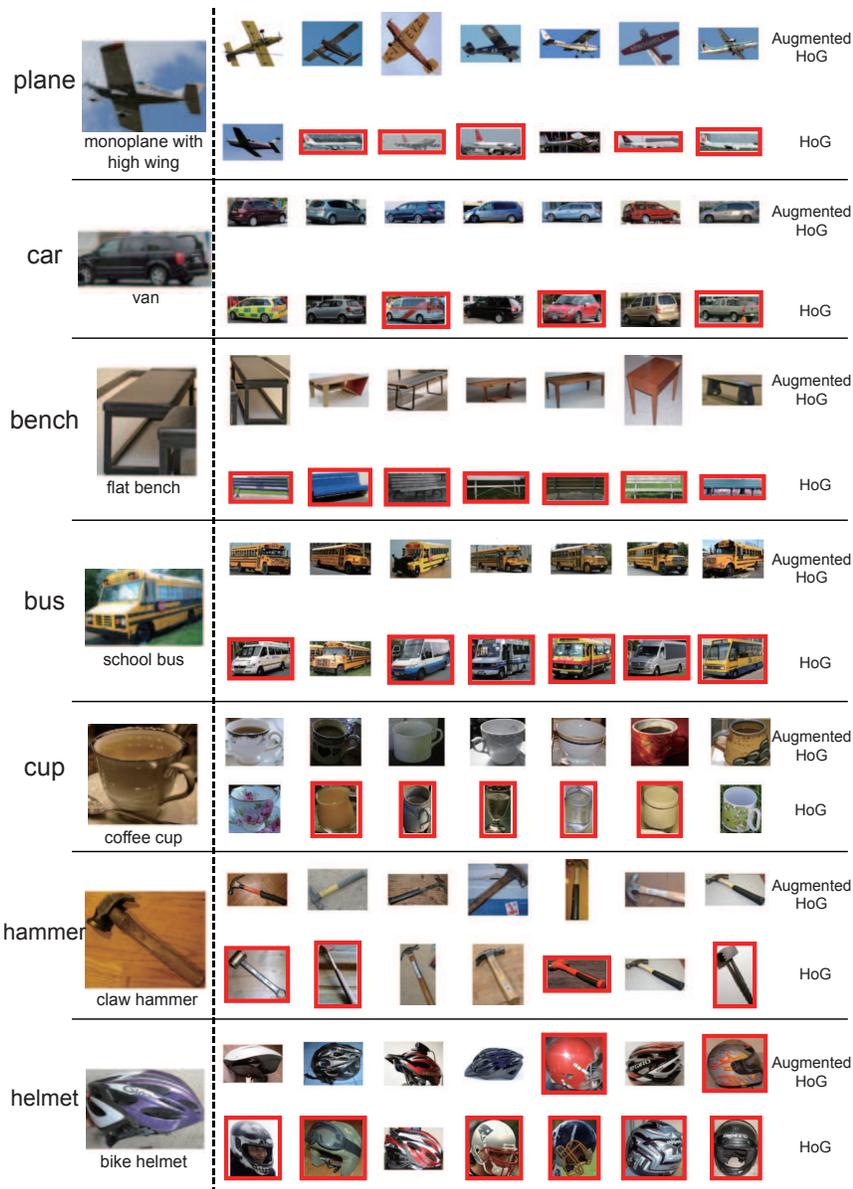


Figure 8: **Fine-grained image retrieval examples.** The first column is the query image and rest columns are retrieval results. Images with red boxes are incorrect retrieval results, which is not from the same fine-grained class according to ImageNet.

To incorporate both the geometric and visual appearance feature, we augment the HoG feature and color histogram feature of each image and concatenate them as one feature vector for retrieval task. The baseline methods include original HoG+Color feature, “Sivic 03” [24], and RANSAC. For the RANSAC verification, the similarity between two images is determined as the number of matched SIFT keypoints points after spatial verification. All methods are evaluated on the bounding boxes of car images, and pr-curve for different methods are shown in Fig. 9.

### 5.3. Method Analysis

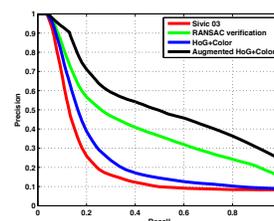
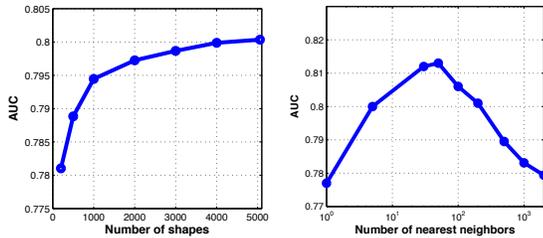


Figure 9: **Instance-level object retrieval results.**

**Applicability for Other Features** Our approach is not restricted to any specific kind of descriptors. Several different kinds of features, including HoG, Bag-of-Visual-



(a) Size of shape collection. (b) Neighborhood for LLE  
Figure 10: **Parameter sensitivity.**

Words [6], Fisher Vector [22], LLC [29], and features extracted by convolutional neural networks (CaffeNet [10]) from different layers are all augmented and tested here. AUC scores for pr-curves of fine-grained image retrieval tasks are reported in Table 2, and the image data sets used here are several example classes from the 55 classes in §5.2. It can be seen that for different choices of underlying features, our method can always boost the performance.

**Parameter Sensitivity** Fig. 10a shows the AUC score changing with different number of 3D models in fine-grained retrieval on “Chair” class. Intuitively, a larger shape collection is preferred since it can provide better coverage of the shape space and further help better reconstruct the descriptor on novel views. However, we also observe that the performance with 200 3D models is only 2% lower than the performance with the full collection of 5,057 3D models. The reason is that our model has the ability to “interpolate” in the shape space, which compensates for the absence of large shape collection at query time.

Fig. 10b shows the AUC changing with the parameter  $k$  for obtaining the local neighborhood in Eq (3). Specifically, for  $k = 1$ , it is equivalent to using the most similar shape to represent the query object, which is an intuitive baseline method. It is beneficial to use an appropriate range of neighborhood to reconstruct the query latent shape, as shown in Fig. 10b.  $k = 200$  is used for other experiments.

**Robustness of Multi-View Feature Augmentation** It is intuitive to synthesize image features at a single predefined view point, and perform image retrieval on this particular view. Fig. 11 shows that, searching on one view point definitely provide reasonable results (the first 3 rows), but the resulted ranking is not stable. Additionally, the feature synthesis works better on some view points because they are more informative than the others. However, if we combine features from all view points, the retrieval results look much

	[17] (SPM)	[17] with b.box	Ours
Accuracy	0.487	0.561	0.603

Table 1: **Accuracy comparison on FGVC-aircraft.** Note that our results is based on [17] with bounding boxes.

Feature	Method	Chair	Car	Bus	Motorbike	Train
HoG	original	0.710	0.278	0.374	0.407	0.521
	augmented	0.801	0.320	0.430	0.480	0.636
BoVW	original	0.678	0.280	0.380	0.402	0.521
	augmented	0.702	0.309	0.417	0.441	0.610
Fisher	original	0.675	0.270	0.353	0.421	0.481
	augmented	0.702	0.307	0.384	0.469	0.602
LLC	original	0.717	0.283	0.354	0.406	0.559
	augmented	0.749	0.348	0.449	0.457	0.602
Caffe Pool5	original	0.690	0.267	0.391	0.421	0.553
	augmented	0.746	0.310	0.420	0.448	0.557
Caffe FC7	original	0.744	0.287	0.386	0.456	0.582
	augmented	0.785	0.348	0.425	0.498	0.613

Table 2: **Performance by different image features.**

better because the augmented multi-view feature contains information from all views and is more robust.

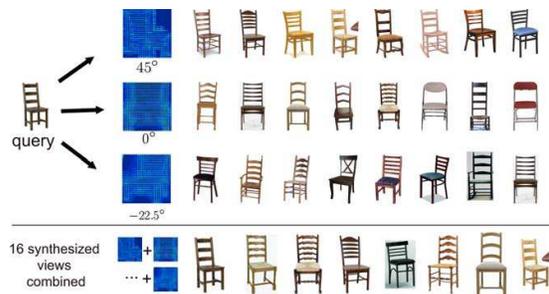


Figure 11: **Image retrieval results by synthesized features from different views**

## 6. Conclusion and Future Work

In this paper, we have proposed a framework for synthesizing features of an object in a single input image from a novel view point, given a collection of 3D models from the same object class. The synthesized features from a predefined list of views serve as an augmentation of the original feature, which is a view-independent description of the object. We then achieve view-invariant image comparison, only focusing on the intrinsic object properties. The proposed feature synthesis framework is analyzed theoretically and empirically, and the augmented features have been evaluated on various computer vision tasks.

**Acknowledgement.** We would like to thank the anonymous reviewers, Chuiwen Ma, Liang Shi, and Qixing Huang for the useful comments. We especially thank Jiantao Jiao for his insightful suggestions. This work was supported in part by NSF grants IIS 1016324, 1528025, and DMS 1521608, AFOSR grant FA9550-12-1-0372, ONR grant N00014-13-1-0341, a Google Focused Research Award, and the Mac Planck Center for Visual Computing and Communication, and Nvidia corporation.

## References

- [1] A. Dosovitskiy, J. T. Springenberg, and T. Brox. Learning to generate chairs with convolutional neural networks. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2
- [2] M. Aubry, D. Maturana, A. Efros, B. Russell, and J. Sivic. Seeing 3d chairs: exemplar part-based 2d-3d alignment using a large dataset of cad models. In *CVPR*, 2014. 1
- [3] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *ECCV 2006*, pages 404–417. Springer, 2006. 2
- [4] S. Bell and K. Bala. Learning visual similarity for product design with convolutional neural networks. In *ACM Transactions on Graphics (SIGGRAPH 2015)*. ACM, 2015. 2
- [5] D.-Y. Chen, X.-P. Tian, Y.-T. Shen, and M. Ouhyoung. On visual similarity based 3d model retrieval. In *Computer graphics forum*, volume 22, pages 223–232. Wiley Online Library, 2003. 1
- [6] G. Csurka, C. Bray, C. Dance, and L. Fan. Visual categorization with bags of keypoints. *Workshop on Statistical Learning in Computer Vision, ECCV*, pages 1–22, 2004. 2, 8
- [7] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR 2005*, volume 1, pages 886–893. IEEE, 2005. 1, 2
- [8] P. Felzenszwalb and D. Huttenlocher. Pictorial structures for object recognition. *IJCV*, 1:55–79, 2005. 2
- [9] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 6
- [10] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014. 8
- [11] J. D. L. F.-F. Jonathan Krause, Michael Stark. 3D object representations for fine-grained categorization. *4th IEEE Workshop on 3D Representation and Recognition, at ICCV 2013 (3dRR-13)*. Sydney, Australia., 2013. 6
- [12] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012. 2
- [13] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR 2006*. 2
- [14] J. L. Long, N. Zhang, and T. Darrell. Do convnets learn correspondence? In *Advances in Neural Information Processing Systems*, pages 1601–1609, 2014. 1, 2
- [15] Y. Low, D. Agarwal, and A. Smola. Multiple domain user personalization. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 123–131. ACM, 2011. 1
- [16] D. Lowe. Object recognition from local scale-invariant features. In *ICCV*, 1999. 2
- [17] S. Maji, E. Rahtu, J. Kannala, M. B. Blaschko, and A. Vedaldi. Fine-grained visual classification of aircraft. *CoRR*, abs/1306.5151, 2013. 6, 8
- [18] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on*, pages 512–519. IEEE, 2014. 2
- [19] K. Rematas, T. Ritschel, M. Fritz, and T. Tuytelaars. Image-based synthesis and re-synthesis of viewpoints guided by 3d models. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 3898–3905. IEEE, 2014. 2
- [20] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000. 4
- [21] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. *CoRR*, abs/1409.0575, 2014. 2
- [22] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek. Image classification with the fisher vector: Theory and practice. *International journal of computer vision*, 105(3):222–245, 2013. 2, 8
- [23] S. Savarese and L. Fei-Fei. 3d generic object categorization, localization and pose estimation. In *ICCV*, 2007. 1
- [24] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. *Proceedings of IEEE International Conference on Computer Vision*, 2003. 7
- [25] H. Su, Q. Huang, N. J. Mitra, Y. Li, and L. Guibas. Estimating image depth using shape collections. *SIGGRAPH 2014*. 2
- [26] H. Su, M. Savva, L. Yi, A. X. Chang, S. Song, F. Yu, Z. Li, J. Xiao, Q. Huang, S. Savarese, T. Funkhouser, P. Hanrahan, and L. J. Guibas. ShapeNet: An information-rich 3d model repository. <http://www.shapenet.org/>. 2015. 5
- [27] J. B. Tenenbaum, V. De Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 2000. 4
- [28] R. Vidal. A tutorial on subspace clustering. *IEEE Signal Processing Magazine*, 28(2):52–68, 2010. 4
- [29] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3360–3367. IEEE, 2010. 8
- [30] Z. Wu, S. Song, A. Khosla, X. Tang, and J. Xiao. 3d shapenets for 2.5d object recognition and next-best-view prediction. *CoRR*, abs/1406.5670, 2014. 2
- [31] Z. Zhu, P. Luo, X. Wang, and X. Tang. Multi-view perceptron: a deep model for learning face identity and view representations. In *Advances in Neural Information Processing Systems*, pages 217–225, 2014. 2