# 3D Object Representations for Fine-Grained Categorization

Jonathan Krause[1], Michael Stark[1,2], Jia Deng[1], and Li Fei-Fei[1]

[1]Computer Science Department, Stanford University
[2]Max Planck Institute for Informatics

## Abstract

*While 3D object representations are being revived in the context of multi-view object class detection and scene understanding, they have not yet attained wide-spread use in fine-grained categorization. State-of-the-art approaches achieve remarkable performance when training data is plentiful, but they are typically tied to flat, 2D representations that model objects as a collection of unconnected views, limiting their ability to generalize across viewpoints. In this paper, we therefore lift two state-of-the-art 2D object representations to 3D, on the level of both local feature appearance and location. In extensive experiments on existing and newly proposed datasets, we show our 3D object representations outperform their state-of-the-art 2D counterparts for fine-grained categorization and demonstrate their efficacy for estimating 3D geometry from images via ultra-wide baseline matching and 3D reconstruction.*

## 1. Introduction

Three-dimensional representations of objects and scenes have been deemed the holy grail since the early days of computer vision due to their potential to provide more faithful and compact depictions of the visual world than view-based representations. Recently, 3D methods have been revived in the context of multi-view object class detection [26, 18, 24] and scene-understanding [14, 11]. For these applications, 3D representations exhibit favorable performance due to their ability to link object parts across multiple views. Surprisingly, these strengths have hardly been exploited in fine-grained recognition, one of the most active areas in computer vision today. There, most state-of-the-art approaches still rely entirely on "flat" image representations that model both the appearance of individual features and their location in the 2D image plane. The former typically takes the form of densely sampled local appearance features such as SIFT [20], often followed by a non-linear coding step [30]. For the latter, various spatial pooling strategies have proven successful, ranging from global

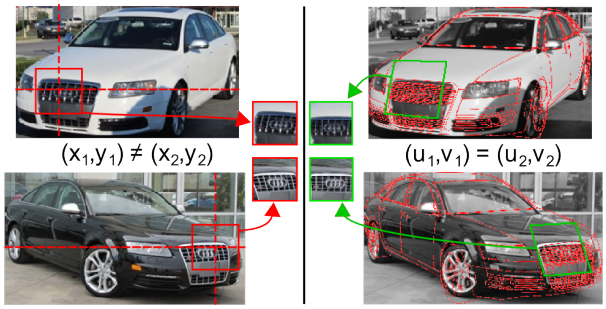


Figure 1: Corresponding patches may have vastly different image coordinates and appearances. Our 3D representations make their positions and appearances directly comparable.

histograms [22] over spatial pyramids [17] to local search regions [8]. While these approaches have delivered remarkable performance in fine-grained categorization tasks that are difficult even for humans [9, 29, 36], they are still limited by the need to observe a relatively dense viewpoint sampling for each category in order to learn reliable models.

In this paper, we therefore take a different route, and follow the intuition that the distinctive features of a fine-grained category, such as the characteristic grille of the car in Fig. 1, are most naturally represented in 3D object space, not in the 2D image plane – this comprises both the appearance of the features (appearance varies with viewpoint, so a viewpoint-independent appearance representation is desired) and their location with respect to an object (the grille appears in a specific region of the 3D car surface, not necessarily in a particular image position). We establish the notion of 3D object space by first obtaining an estimate of the 3D geometry of an object, then representing features relative to this geometry. Specifically, the geometry estimate allows rectification of an image patch with respect to the estimated surface normal at its center point and characterizes its location as coordinates on the 3D object surface.

The basis of our implementation is given by two state-of-the-art 2D object representations. The first, spatial pyramid matching [17], specifically using locality constrained linear coding (LLC [30]), has attained wide-spread use due to

its consistently high performance on various image categorization benchmarks, combining local feature coding with a spatial pyramid representation. The second representation, BubbleBank (BB [8]), has recently been shown to outperform prior work in fine-grained categorization. It relies on extracting discriminative patches from training images, convolving them in local regions within a test image, and using the responses as features.

Our paper makes the following contributions: First, we lift two state-of-the-art 2D object representations to 3D w.r.t. both the appearance and location of local features. We demonstrate the resulting 3D object representations outperform both their respective 2D counterparts and state-of-the-art baselines in fine-grained categorization. Second, we introduce a new dataset of 207 fine-grained categories that will be made publicly available upon publication, separated into two subsets: a small-scale, but ultra-fine-grained set of 10 BMW models, and a large-scale set of 197 car types. Third, we demonstrate the usefulness of our 3D object representation for estimating 3D geometry from test images in the form of ultra-wide baseline matching [37]. Fourth, we provide first experimental results on the challenging task of 3D reconstruction of fine-grained categories, which, to our knowledge, has not been attempted in the literature before.

## 2. Related Work

Fine-grained recognition is a growing subfield of computer vision. Driven by real-life applications, the focus has so far mostly been on distinguishing animate categories, such as flowers [22], leaves [16], dog breeds [19, 23], and birds [9, 29, 36]. For these categories, the challenge consists in capturing subtle appearance differences (such as a differently colored beak of a bird) despite variations in articulated pose, which are most reliably handled by collecting and memorizing discriminative, local appearance features from each available object view [34, 8].

**The role of object parts.** It has been realized that spatial information typically aids categorization, either for providing a frame of reference in which appearance features are computed (spatial pooling [17, 30, 9, 29, 36, 8]), or as a feature in itself [2, 27]. For both, object part detectors have proven to provide reliable spatial information, as constellations [2], deformable parts [29, 27], and poselets [4, 36]. While these models successfully leverage spatial information, they are still "flat", i.e., built on independent views.

A remarkable exception to this trend is the recent work by Farrell *et al*. [9], which implements local appearance features and spatial pooling relative to a volumetric 3D bird model. While our work is similar in spirit, it goes beyond [9] in several important directions. First, in contrast to [9], our work does not rely on extensive annotation of training data. Instead, we leverage existing 3D CAD models for the basic-level object class of interest, without the need for any manual intervention. Second, we explicitly design our methods to be robust against errors in the estimation of rough 3D geometry by pooling over multiple predictions of coarse category and viewpoint instead of relying on a single prediction. Third, instead of focusing on a single representation for spatial pooling (two 3D ellipsoids for a bird's head and body for [9]), we compare and extend into 3D two different, state-of-the-art 2D methods for spatial pooling (SPM and BB), demonstrating improved performance.

**3D representations for recognition.** We draw inspiration from approaches in multi-view object class recognition that leverage 3D representations to establish correspondences across different viewpoints [37, 24, 12, 25]. While these prior approaches establish correspondences on a fixed, small set of object parts, our 3D variant of BB yields hundreds of correspondences on the level of individual local features, making them applicable to challenging tasks such as 3D reconstruction from a fine-grained category (Sect. 5.3). Similar to prior works in multi-view detection, we leverage 3D CAD models to generate synthetic training data [18, 37, 32, 24].

## 3. 3D Object Representations

In the following, we describe the design of our 3D object representations for fine-grained categorization. Both are based on estimating the 3D geometry of an object under consideration (Sect. 3.2), and represent both the appearance of individual local features (Sect. 3.3) and their locations (Sect. 3.4) in 3D object space. While our representations are based on state-of-the-art 2D representations that have proven to be effective for recognition (SPM [30], BB [8]), we effectively lift them to 3D by exchanging the underlying parameterization of feature extraction and spatial pooling, leading to improved performance (Sect. 5).

### 3.1. 2D Base Representations

We start with a brief review of our 2D base representations, spatial pyramids (SPM) and bubble bank (BB). The **spatial pyramid** (SPM) representation [17, 30] is an extension of the bag-of-visual-words paradigm [6], enriching local appearance features with spatial information by pooling them with respect to spatial grids of varying resolution, concatenating the result into a large feature vector. We consider SPM in combination with locality-constrained linear coding (LLC [30]), which is considered state-of-the-art for a wide variety of image classification benchmarks.

The **BubbleBank** (BB) [8] representation is based on the notion of "bubbles": feature templates that are convolved with an image in local search regions, where the regions are determined by the template's image location during training. These responses are max-pooled over the region to

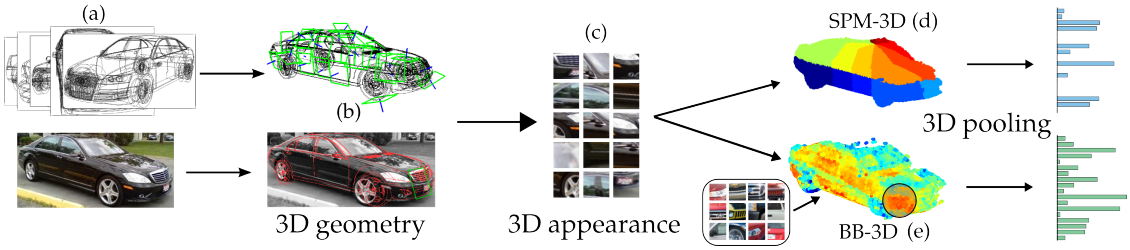


Figure 2: An overview of how we estimate 3D geometry and lift SPM and BB to 3D. See text for details.

produce a single feature. The final representation consists of pooling responses of a large bank of bubbles over their respective search regions. In [8], bubbles were defined manually via crowdsourcing, but also deliver state-of-the-art performance even when randomly sampled from training data, in a manner similar to [33].

### 3.2. 3D Geometry Estimation

The basis of our 3D object representations is given by estimating the 3D geometry of an object, providing a frame of reference for both our 3D appearance representation (Sect. 3.3) and our 3D spatial pooling (Sect. 3.4). Since we are focusing on rigid objects (cars), we can model rough 3D object geometry by a discrete set of exemplar 3D CAD models, which are readily available. The first stage in our fine-grained categorization pipeline (Fig. 2) consists of identifying one (or multiple) CAD model(s) that best fit the image. The matching between a 3D CAD model and a 2D image is implemented by a set of classifiers that have been trained to distinguish among CAD models and viewpoints.

**Synthetic training data.** In line with recent work in multi-view recognition [18, 24], we leverage 3D CAD models as a cheap and reliable source of synthetic training data. Crucially, this gives precise 3D coordinates that we can use to anchor our appearance and location representation and is entirely free of human intervention (in contrast to approaches relying on part annotations [9, 36, 19]). In our experiments (Sect. 5), our 3D geometry classifiers are trained from 41 CAD models of cars (Fig. 2(a)), rendered at 36 azimuths, 4 elevations, and against 10 random background variations, for a total of 59,040 synthetic images.

**3D Geometry classifiers.** In order to match 3D CAD models to 2D images, we train a massive bank of classifiers for nearly the entire cross product of CAD models and viewpoints. In practice, we found it sufficient to group CAD models belonging to the same coarse category together (for cars, we define sedan, SUV, coupe, convertible, pickup, hatchback, and station wagon as coarse categories), and train a bank of viewpoint-dependent classifiers for all of them, resulting in 1,008 possible combinations of viewpoints and coarse categories. All classifiers are based on HOG [7] features in connection with a one-vs-all linear SVM. Empirically, we found that exemplar SVMs [21] did not result in a significant improvement but where computationally much more demanding.

**Multiple hypotheses.** An incorrect estimation of 3D geometry leads to errors in later stages of the fine-grained categorization pipeline that are hard to recover from. Thus, rather than commit to a single viewpoint and coarse category, we maintain a list of the top $N$ estimates, and max-pool features across all of them. Fig. 4(a) verifies the positive effect on performance.

### 3.3. 3D Appearance Representation

The goal of our 3D appearance representation is to ensure that a discriminative local feature is represented only once, as opposed to requiring multiple representations in different viewpoints. Making these connections across viewpoints is important in order to generalize to test cases that have been observed from viewpoints not present in the training data. We achieve this through an appearance representation that is (to an extent) viewpoint invariant, by transforming local image patches into a unified frame of reference prior to feature computation.

**Patch sampling.** The basis of our 3D appearance representation is given by a dense sampling of image patches that we extract from a given training or test image. In contrast to existing 2D representations, we sample patches directly from the 3D surface of the object of interest relative to its estimated 3D geometry (Sect. 3.2). In particular, we precompute thousands of uniformly spaced patch locations on the surface of our CAD models by dart throwing [5]. Each patch location comes with an associated surface normal and upward direction, determining its 3D orientation, and a flat, planar rectangle, determining its support region (Fig. 2(b)). For feature extraction, we project all patches visible from the estimated viewpoint into the image, resulting in a set of perspectively distorted quadrilaterals.

**Patch rectification and descriptors.** Prior to feature computation, we rectify the projected quadrilaterals to a common reference rectangle, effectively compensating for perspective projection. While this applies only to locally planar surfaces in theory, it typically results in correctly rectified patches also for curved surfaces, such as the car grille in Fig. 1 and the patches in Fig. 2(c). We densely sample RootSIFT [1, 20] descriptors on each rectified patch.

### 3.4. 3D Spatial Pooling

The goal of our 3D spatial pooling is to characterize the position of local features with respect to the 3D geometry of an object. Like 3D appearance (Sect. 3.3), we utilize our 3D geometry estimate (Sect. 3.2) as the basis.

**3D Spatial Pyramid (SPM-3D).** After patch extraction we have a set of rectified patches with corresponding 3D locations. As in the case of 2D SPM, we can extract descriptors from each of these patches and quantize them using a trained codebook. However, unlike a standard 2D SPM, which only considers the 2D location of each patch, our representation includes the location of each patch in 3D object coordinates, allowing us to pool over a more relevant space. Specifically, we partition the surface of the object based on azimuth and elevation relative to the center of the CAD model, *i.e.* we partition the space $\mathcal{S} = [0, 2\pi] \times [-\frac{\pi}{2}, \frac{\pi}{2}]$, as visualized in Fig. 2(d), and pool quantized descriptors accordingly. As in 2D SPM, we use multiple scales, *i.e.* we pool over $1 \times 1$, $2 \times 2$, and $4 \times 4$ partitions of $\mathcal{S}$.

**3D BubbleBank (BB-3D).** Similar to our lifting of 2D SPM to 3D, we lift 2D BubbleBank (BB) to 3D by converting pooling regions from 2D to 3D. After extracting descriptors for each of the rectified patches, we convolve the descriptors with a set of bubbles obtained randomly over the training set. Crucially, each of the patches and bubbles has an associated 3D location, so we can pool over all patches within a 3D region of the bubble, illustrated in Fig. 2(e). By pooling over regions of sufficient size, additional robustness w.r.t. 3D geometry estimation is obtained. Our approach contrasts with the 2D equivalent [8] in that we do not rely on a feature appearing in the same 2D location within an image during both training and test, but rather at the same location with respect to 3D object geometry.

### 3.5. Classification with 3D Representations

We combine our 3D object representations (SPM-3D, BB-3D) with linear SVM classifiers that we train in a one-vs-all fashion for fine-grained categorization, in analogy to their 2D counterparts (SPM, BB), allowing us to pinpoint performance differences to the respective representations.

## 4. Novel Fine-Grained Dataset

In order to provide a suitable test bed for our 3D representations, we have collected a challenging, large-scale dataset of car models, to be made available upon publication. It consists of BMW-10, a small, ultra-fine-grained set of 10 BMW sedans (512 images) hand-collected by the authors, plus car-197, a large set of 197 car models (16,185 images) covering sedans, SUVs, coupes, convertibles, pickups, hatchbacks, and station wagons. Since dataset collection proved non-trivial, we give the most important challenges and insights.

**Identifying visually distinct classes.** Since cars are man-made objects whose class list changes on a yearly basis, and models of cars do not have a different appearance from year to year, no simple list of visually distinct cars exists which we can use as a base. We thus first crawl a popular car website for a list of all types of cars made since 1990. We then apply an aggressive deduplication procedure, based on perceptual hashing [35], to a limited number of provided example images for these classes, determining a subset of visually distinct classes, from which we sample 197 (see supplementary material for a complete list).

**Finding candidate images.** Candidate images for each class were collected from Flickr, Google, and Bing. To reduce annotation cost and ensure diversity in the data, the candidate images for each class were deduplicated using the same perceptual hash algorithm [35], leaving a set of several thousand candidate images for each of the 197 target classes. These images were then put on Amazon Mechanical Turk (AMT) in order to determine whether they belong to their respective target classes.

**Training annotators.** The main challenge in crowdsourcing the collection of a fine-grained dataset is that workers are typically non-experts. To compensate, we implemented a qualification task (a set of particularly hard examples of the actual annotation task) and provide a set of positive and negative example images for the car class a worker is annotating, drawing the negative examples from classes known a priori to be similar to the target class.

**Modeling annotator reliability.** Even after training, workers differ in quality by large margins. To tackle this problem, we use the Get Another Label (GAL) system [15], which simultaneously estimates the probability a candidate image belongs to its target class and determines a quality level for each worker. Candidate images whose probability of belonging to the target class exceeds a specified threshold are then added to the set of images for that category. After obtaining images for each of the 197 target classes, we collect a bounding box for each image via AMT, using a quality-controlled system provided to us by the authors of [28]. Finally, an additional stage of deduplication is performed on the images when cropped to their bounding boxes. Fig. 3 shows example dataset images.

## 5. Experiments

In the following, we carefully analyze the performance of our 3D object representations for a variety of different tasks, highlighting both their discriminative power for categorization and their unique ability to provide precise, pointwise correspondences between largely different views of the same object or even different instances of the same fine-grained category. First, we consider the task of fine-grained

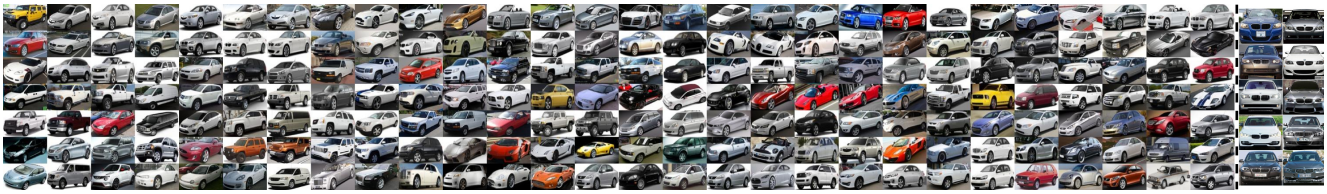

Figure 3: One image each of 196 of the 197 classes in car-197 and each of the 10 classes in BMW-10.

categorization of cars on various datasets (Sect. 5.1), reporting superior performance of our 3D object representations (SPM-3D, BB-3D) in comparison to both their respective 2D counterparts (SPM, BB) and state-of-the-art baselines [7, 3, 27, 30, 8]. Second, we successfully estimate the change in camera pose between multiple views of the same object (ultra-wide baseline matching, Sect. 5.2). And third, we give first promising results on reconstructing partial 3D models from fine-grained category instances (Sect. 5.3).

**Implementation Details.** We present the most important implementation details here, leaving the remainder for supplementary material. Cropped images are used, as is standard in fine-grained classification, and are scaled such that the maximum dimension is 300 pixels. KDES [3] uses only grayscale kernel descriptors SPM uses codebook size 4096 and 3 layers of SPM (1x1, 2x2, and 4x4). BB uses 10k and 20k bubbles for small-scale and large-scale experiments, respectively, and a pooling region of $50\%$ of the image height and the entire image width. For our 3D object representations, we use 3 viewpoint/coarse category hypotheses and extract patches corresponding to 3 CAD models for each such hypothesis. On BMW-10 and BMW-10 (flipped), only sedan CAD models are used. For SPM-3D, we use a codebook of size 4096. For BB-3D, we use 10k bubbles at small-scale and 20k bubbles at large scale. Features for BB and BB-3D use a power-scaling [8] parameter of 16. BB-3D-G refers to pooling bubble responses globally, and BB-3D-L pools across the width and length of the car, but only 25% of its height. SPM-3D-L uses 1x1, 2x2, and 4x4 partitions of azimuth-elevation space, with SPM-3D-G using only a 1x1 partition. In all cases we train one-vs-all $L_2$-regularized $L_2$-loss SVMs, selecting a regularization parameter $C$ from the range $10^{-6}, 10^{-5}, \ldots, 10^{10}$ by 25-fold cross-validation for small-scale experiments and use a constant $C$ value $10^{10}$ for large-scale experiments.

## 5.1. Fine-Grained Categorization

We commence our evaluation by comparing the performance of our 3D object representations, their respective 2D counterparts, and state-of-the-art baselines for the task of fine-grained categorization. To that end, we consider three different datasets of varying granularity. i) We use the existing car-types dataset [27], consisting of 14 car classes from a variety of coarse categories (sedans, hatchbacks, sedans, SUVs, convertibles), to establish that our methods outperform the previous state-of-the-art in car classification. ii) We provide an in-depth analysis of our methods in comparison to their respective 2D counterparts on BMW-10, our ultra-fine-grained set of 10 BMW sedans. iii) We demonstrate the ability of our methods to scale up to hundreds of fine-grained classes on our large-scale dataset (see Sect. 4).

**i) Car-types.** Tab. 1(a) gives classification accuracy for our methods SPM-3D, BB-3D-G, and prior work. Curiously, the performance on this dataset seems to have almost saturated, with the weakest prior method (a simple HOG template) achieving $77.5\%$ and the strongest (structDPM, a multi-class DPM [10]) achieving $93.5\%$ accuracy. We believe this high level of performance to indicate a rather coarse granularity of this dataset, which is reinforced by the two strongest prior methods (PB(mvDPM), $85.3\%$, structDPM, $93.5\%$) being based on part-layout information rather than discriminative local features (such as SPM, $84.5\%$). In comparison, our method BB-3D-G ($94.5\%$) outperforms the best reported prior result of structDPM ($93.5\%$) by $1\%$. In addition, unlike structDPM, it scales effortlessly to large-scale datasets such as car-197, since it does not rely on joint regularization across all classes. Comparing 3D to 2D, BB-3D-G outperforms BB ($92.6\%$) by $1.9\%$, and SPM-3D ($85.7\%$) beats SPM by $1.2\%$.

**ii) BMW-10.** Tab. 1(b) gives the results for our ultra-fine-grained dataset of 10 BMW sedans, focusing on different variants of our methods SPM-3D and BB-3D and their respective 2D counterparts, and adding KDES [3] to the state-of-the-art baselines. We make the following observations: first, the general level of performance is significantly lower than for car-types (HOG achieves an accuracy of $28.3\%$, PB(mvDPM) $29.1\%$), which indicates that our dataset is significantly more fine-grained. Second, as a result, methods relying on discriminative local features rather than a global feature template (HOG) or part layout (PB(mvDPM)) perform much better (KDES $46.5\%$, SPM $52.8\%$, BB $58.7\%$). Third, our 3D object representations improve significantly over their respective 2D counterparts: SPM-3D-L ($58.7\%$) improves over SPM by $5.9\%$ and BB-3D-G ($66.1\%$) improves over BB by $7.4\%$.

In Tab. 1(b), we also investigate the impact of enriching the original set of training images by flipped versions of each image, effectively doubling the amount of training data and increasing the density with which different object viewpoints are represented. Performance improves significantly

(a)

| | HOG [7] | PB(mvDPM) [27] | structDPM [27] | SPM [30] | SPM-3D-L **(ours)** | BB [8] | BB-3D-G **(ours)** |
|---|---|---|---|---|---|---|---|
| car-types | 77.5 | 85.3 | 93.5 | 84.5 | 85.7 | 92.6 | **94.5** |

(b)

| | HOG [7] | PB(mvDPM) [27] | KDES [3] | SPM [30] | SPM-3D-G **(ours)** | SPM-3D-L **(ours)** | BB [8] | BB-3D-G **(ours)** | BB-3D-L **(ours)** |
|---|---|---|---|---|---|---|---|---|---|
| BMW-10 | 28.3 | 29.1 | 46.5 | 52.8 | 58.3 | 58.7 | 58.7 | **66.1** | 64.7 |
| BMW-10 (flipped) | - | - | - | 66.1 | - | 67.3 | 69.3 | **76.0** | - |

Table 1: (a) Comparison to state-of-the-art on *car-types* [27]. (b) In-depth analysis on our *BMW-10* dataset.

for all methods, by 8.6% (SPM-3D-L), 9.9% (BB-3D-G), 10.6% (BB), and 13.3% (SPM). Notice that while the 2D methods benefit more from adding training data, the relative ordering of results between different methods is consistent with BMW-10 without flipping: SPM-3D-L (67.3%) outperforms SPM by 1.2%, and BB-3D-G (76.0%) outperforms BB by 6.7%. Fig. 5(a) visualizes the discriminative power of each of the 10k bubbles of BB-3D-G on BMW-10 (flipped). Each point is a 3D bubble location, with its size and hue proportional to $\sum_{j=1}^{10} |w_{i,j}|^5$, where $w_{i,j}$ is the SVM weight on feature $i$ for class $j$. This indicates that discriminative features are primarily at the front and back of cars, which is both correct and human-interpretable.

**Global vs. local pooling.** In Tab. 1(b), we further examine the impact of the size of the 3D pooling region on performance, distinguishing global pooling (SPM-3D-G, BB-3D-G) and local pooling (SPM-3D-L, BB-3D-L). For SPM-3D, local pooling improves performance slightly by 0.4%. For BB-3D, global pooling is 1.4% better, beating the next best result by 6%. We believe this counterintuitive superiority of BB-3D-G over BB-3D-L to be due to 1) mispredictions in viewpoint, and 2) the strong left-right symmetry of cars: it can help to look on the right side of a car for a bubble originally found on the left side. On the basis of these results all other experiments have used SPM-3D-L and BB-3D-G.

**Amount of training data.** Fig. 4(b) plots classification accuracy (y-axis) of the two best performing 2D and 3D methods of Tab. 1(b) versus the number of training images, which we vary from 1 to 16 in powers of two (we also add the results on all of BMW-10 and BMW-10 (flipped) as the two right-most points for reference). For each experiment, we randomly sample training images and fix them for both methods. We observe the 3D representation to outperform its 2D counterpart consistently for all numbers of training images. The difference is most pronounced for 16 training images (9.8%) and decreases for larger numbers (to 7.4% and 6.7%, respectively), indicating that the 3D representation can utilize training data more effectively.

**iii) Car-197.** Tab. 2(top) gives the results for our large-scale dataset of 197 fine-grained car categories, again comparing SPM-3D-L and BB-3D-G to SPM and BB. Surprisingly, SPM performs remarkably well with the increased data (69.5%), beating our best 3D representation BB-3D-G (67.6%) by 1.9%. Similarly to on the car-types, BMW-10, and BMW-10 (flipped) datasets, BB-3D-G outperforms the

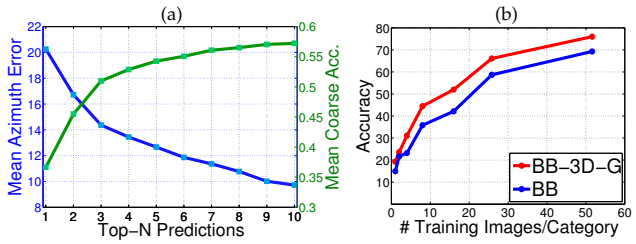


Figure 4: (a) Viewpoint/coarse category acc. over top-$N$ predictions. (b) Accuracy vs. number of training images.

2D version BB (63.6%) by 4%, and SPM-3D-L (65.6%) by 2%. Analyzing these results, we believe our 3D representations to suffer from certain coarse categories (e.g., large pickup trucks) being underrepresented in our 3D CAD models, which is a shortcoming of this particular experimental setup, not a limitation of our methods. Stacking the four methods together naturally results in the best performance, 75.5%, indicating that our 3D representations encode information that is complementary to their 2D counterparts, despite using the same base descriptors.

**Summary.** We conclude that lifting 2D object representations to 3D is in fact beneficial for all except one case for both SPM and BB, leading to significant improvements in classification performance over state-of-the-art on existing (car-types) and ultra-fine-grained (BMW-10) datasets.

## 5.2. Ultra-Wide Baseline Matching

Characterizing the relation between different views of the same visual scene is one of the most important tasks in computer vision, providing the basis for 3D reconstruction using structure-from-motion techniques [13]. For known intrinsic camera parameters, it can be phrased as estimating the fundamental matrix, based on putative point-to-point correspondences between the views. Ultra-wide baseline matching has been suggested [37] as a way to quantify the ability of a method to localize corresponding 3D points across viewpoints, specifically for wide baselines between 45° and 180° for which pure local feature-based methods such as SIFT [20] typically fail.

**Methodology.** We follow the protocol of [37] and perform ultra-wide baseline matching on 134 image pairs of the *3D Object Classes* dataset [26]. We modify BB-3D-L to find putative correspondences for pairs of images as follows: for a given pair, each patch in the first image defines a bubble in the second image, and is convolved with

all patches of the second image that fall into the bubble's 3D pooling region (0.1 of the object surface). For each bubble, the maximally responding patch is memorized, and possibly kept as a putative correspondence after thresholding. A fundamental matrix relating both images is then computed using standard multi-view geometry [13] and RANSAC. Fig. 5(b) visualizes the results for 3 image pairs using BB-3D-L by depicting a random selection of epipolar lines and corresponding feature matches. Please note, although this method uses rough 3D geometry information to bootstrap patch rectification and local pooling (Sect. 3.2), it still mainly relies on the local evidence of the test image pair to establish feature-level correspondences (like SIFT). It can not be expected to deliver results for baselines larger than $135^\circ$ (denoted "-" in Tab. 2), since no local evidence is shared between the views.

**Results.** Tab. 2 gives ultra-wide baseline matching results, where performance is measured as the fraction of fundamental matrices for which the Sampson error [13] w.r.t. ground truth correspondences falls under a threshold (20 pixels as in [37]). Rows of Tab. 2 correspond to different baselines (difference in azimuth angle between two views of the same object). The last two rows give averages over all baselines (Av.1, imputing zero accuracy for "-"), and over only the first two baselines that can be delivered by all methods (Av.2). Columns of Tab. 2 compare two variants of our BB-3D-L, BB-3D-S (using only a single coarse category and viewpoint prediction) and BB-3D-M (using multiple), with comparable local feature-based methods, SIFT [20] and the multi-view local part detectors of [37]. Although we can not expect to compete with the full-blown 3D shape model of [37] and the multi-view incarnation of DPM [10], 3D$^2$PM [24], that leverage global shape and have been explicitly designed and trained for viewpoint prediction on that dataset, we include their results as reference.

From the four left-most columns of Tab. 2, we observe that our 3D object representations outperform all local feature-based methods by a significant margin even for Av.1: SIFT fails catastrophically ($0.5\%$), and local part detectors [37] ($22\%$) are outperformed by both BB-3D-S ($24.6\%$) and BB-3D-M ($25.8\%$). This difference is even more pronounced when considering individual baselines (for $45^\circ$, BB-3D-M ($71.1\%$) outperforms parts by $44.7\%$, for $90^\circ$, BB-3D-S ($40\%$) outperforms parts by $13\%$) or Av.2 (BB-3D-M ($51.6\%$) outperforms parts by $24.6\%$). While being not quite on par with the state-of-the-art results obtained by 3D$^2$PM ($67.8\%$) overall, for $45^\circ$ baseline, BB-3D-M in fact beats 3D$^2$PM by $13.2\%$, and typically provides hundreds of densely spaced inlier features as opposed to a sparse set of 20 object parts, which we will exploit for 3D reconstruction in Sect. 5.3. Using multiple coarse category models (BB-3D-M) improves over the single case (BB-3D-S) by $1.2\%$ (Av.1) and $2.3\%$ (Av.2), respectively, in step with the increased robustness shown in Fig. 4(a).

| | SPM [30] | SPM-3D-L **(ours)** | BB [8] | BB-3D-G **(ours)** | Stacked |
|---|---|---|---|---|---|
| car-197 | **69.5** | 65.6 | 63.6 | 67.6 | 75.5 |

| Bl. | SIFT [20] | Parts [37] | **BB-3D-S** | **BB-3D-M** | 3D Shape [37] | 3D$^2$PM [24] |
|---|---|---|---|---|---|---|
| 45° | 2% | 27% | 58.5% | **71.7%** | 55% | **58.5%** |
| 90° | 0% | 27% | **40%** | 31.4% | 60% | **77.1%** |
| 135° | - | 10% | - | - | 52% | **58.6%** |
| 180° | - | 24% | - | - | 41% | **70.6%** |
| Av.1 | 0.5% | 22% | 24.6% | **25.8%** | 52% | **66.4%** |
| Av.2 | 1% | 27% | 49.3% | **51.6%** | 57.5% | **67.8%** |

Table 2: Top: Results on car-197. Bottom: Ultra-wide baseline matching results on *3D Object Classes* [26].

### 5.3. 3D Fine-Grained Category Reconstruction

Having verified the ability of our BB-3D representation to establish accurate feature correspondences across different views of the same object (Sect. 5.2), we now move on to the even more challenging task of finding correspondences between examples of the same fine-grained category (but not the same instance). Note that this is feasible for cars: their 3D geometry is almost uniquely determined by fine-grained category affiliation, and our features are invariant to the remaining variation (such as color). At the same time, this task is very challenging, since the background varies drastically between different views, and can hence not provide any evidence for correspondence. To our knowledge, reconstruction of a fine-grained category has not been reported in the literature before.

**Methodology.** We run BB-3D-L for all pairs of images of a category to obtain putative correspondences, and feed feature locations and correspondences into VisualSFM, a front-end for multi-core bundle adjustment [31]. We run their SFM pipeline as a black box, using standard parameter settings except for increasing the number of iterations.

**Results.** Fig. 5(c) depicts qualitative results for the reconstruction of a fine-grained class of our BMW-10 dataset (2012 BMW ActiveHybrid 7 Sedan), rendered from different viewpoints. Please note that we have not applied any dense, pixelwise refinement of the sparse reconstruction. The point density is due to the high number of inlier correspondences generated by BB-3D-L. Clearly, the reconstruction is incomplete and contains spurious points, in particular for textureless regions (e.g., the hood) – however, the local 3D pooling of BB-3D-L successfully bounds the degree to which correspondences can "drift away" on the 3D geometry. The resulting reconstruction has a sedan shape, and shows the characteristic grille, headlight, and rear light features of a BMW. We believe this result to be highly promising, opening a vast array of future research directions, such as high-fidelity reconstruction from fine-grained categories, or using the reconstructed model for recognition.

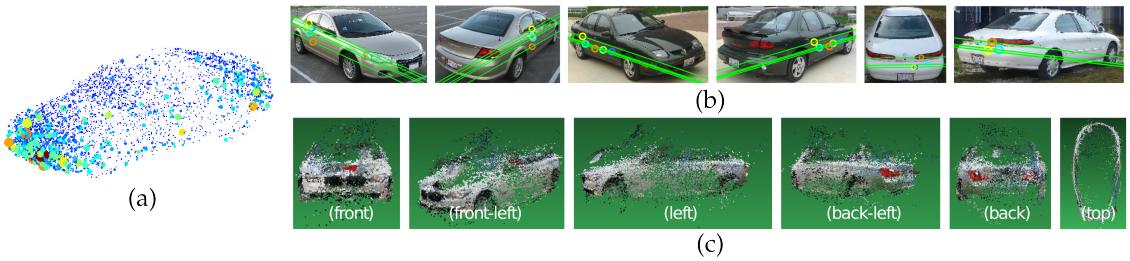


Figure 5: (a) Discriminative bubbles for BB-3D-G on BMW-10 (flipped). (b) Ultra-wide baseline matching on *3D Object Classes* [26] (green: epipolar lines, colored circles: corresponding inlier features). (c) Fully automatic 3D reconstruction from 46 images of a fine-grained category from BMW-10.

## 6. Conclusion

We have demonstrated that 3D object representations can be beneficial for fine-grained categorization of rigid classes, specifically cars. By lifting two state-of-the-art 2D object representations to 3D (SPM and BB), we obtained state-of-the-art performance on both existing (car-types) and our new datasets (BMW-10, car-197). In addition, we leveraged our BB-3D-L representation for ultra-wide baseline matching, and showed first promising results for the automatic reconstruction of 3D models from fine-grained category instances.

## References


[1] R. Arandjelovic and A. Zisserman. Three things everyone should know to improve object retrieval. In *CVPR*, 2012. 3

[2] A. Bar-Hillel and D. Weinshall. Subordinate class recognition using relational object models. In *NIPS*, 2006. 2

[3] L. Bo, X. Ren, and D. Fox. Kernel descriptors for visual recognition. *NIPS*, 2010. 5, 6

[4] L. Bourdev and J. Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *ICCV*, 2009. 2

[5] D. Cline, S. Jeschke, K. White, A. Razdan, and P. Wonka. Dart throwing on surfaces. In *Eurographics*, 2009. 3

[6] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *ECCV*, 2004. 2

[7] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005. 3, 5, 6

[8] J. Deng, J. Krause, and L. Fei-Fei. Fine-grained crowdsourcing for fine-grained recognition. In *CVPR*, 2013. 1, 2, 3, 4, 5, 6, 7

[9] R. Farrell, O. Oza, N. Zhang, V. I. Morariu, T. Darrell, and L. S. Davis. Birdlets: Subordinate categorization using volumetric primitives and pose-normalized appearance. In *ICCV*, 2011. 1, 2, 3

[10] P. F. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *PAMI*, 2010. 5, 7

[11] A. Gupta, A. A. Efros, and M. Hebert. Blocks world revisited: Image understanding using qualitative geometry and mechanics. In *ECCV'10*. 1

[12] P. Gupta, S. S. Arrabolu, M. Brown, and S. Savarese. Video scene categorization by 3d hierarchical histogram matching. In *ICCV*, 2009. 2

[13] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004. 6, 7

[14] D. Hoiem, A. Efros, and M. Hebert. Putting objects in perspective. *IJCV*, 2008. 1

[15] P. G. Ipeirotis, F. Provost, and J. Wang. Quality management on amazon mechanical turk. In *WS ACM SIGKDD*, 2010. 4

[16] N. Kumar, P. N. Belhumeur, A. Biswas, D. W. Jacobs, W. J. Kress, I. C. Lopez, and J. V. Soares. Leafsnap: A computer vision system for automatic plant species identification. In *ECCV*. 2012. 2

[17] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006. 1, 2

[18] J. Liebelt and C. Schmid. Multi-view object class detection with a 3d geometric model. In *CVPR*, 2010. 1, 2, 3

[19] J. Liu, A. Kanazawa, D. Jacobs, and P. Belhumeur. Dog breed classification using part localization. In *ECCV*. 2012. 2, 3

[20] D. Lowe. Distinctive image features from scale invariant keypoints. *IJCV*, 2004. 1, 3, 6, 7

[21] T. Malisiewicz, A. Gupta, and A. A. Efros. Ensemble of exemplar-svms for object detection and beyond. In *ICCV*, 2011. 3

[22] M.-E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *ICVGIP*, 2008. 1, 2

[23] O. M. Parkhi, A. Vedaldi, C. Jawahar, and A. Zisserman. The truth about cats and dogs. In *ICCV*, 2011. 2

[24] B. Pepik, P. Gehler, M. Stark, and B. Schiele. $3d^2pm$ – 3d deformable part models. In *ECCV*, 2012. 1, 2, 3, 7

[25] C. Redondo-Cabrera, R. J. Lopez-Sastre, J. Acevedo-Rodríguez, and S. Maldonado-Bascón. Surfing the point clouds: Selective 3d spatial pyramids for category-level object recognition. In *CVPR*, 2012. 2

[26] S. Savarese and L. Fei-Fei. 3D generic object categorization, localization and pose estimation. In *ICCV*, 2007. 1, 6, 7, 8

[27] M. Stark, J. Krause, B. Pepik, D. Meger, J. J. Little, B. Schiele, and D. Koller. Fine-grained categorization for 3d scene understanding. In *BMVC*, 2012. 2, 5, 6

[28] H. Su, J. Deng, and L. Fei-Fei. Crowdsourcing annotations for visual object detection. In *AAAI-WS*, 2012. 4

[29] C. Wah, S. Branson, P. Perona, and S. Belongie. Multiclass recognition and part localization with humans in the loop. In *ICCV'11*. 1, 2

[30] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *CVPR'10*. 1, 2, 5, 6, 7

[31] C. Wu, S. Agarwal, B. Curless, and S. M. Seitz. Multicore bundle adjustment. In *CVPR*, 2011. 7

[32] Y. Xiang and S. Savarese. Estimating the aspect layout of object categories. In *CVPR*, 2012. 2

[33] B. Yao, G. Bradski, and L. Fei-Fei. A codebook- and annotation-free approach for fine-grained image categorization. In *CVPR'12*. 3

[34] B. Yao, A. Khosla, and L. Fei-Fei. Combining randomization and discrimination for fine-grained image categorization. In *CVPR'11*. 2

[35] C. Zauner. Implementation and benchmarking of perceptual image hash functions. *Master's thesis, Austria*. 4

[36] N. Zhang, R. Farrell, and T. Darrell. Pose pooling kernels for sub-category recognition. In *CVPR*, 2012. 1, 2, 3

[37] M. Z. Zia, M. Stark, B. Schiele, and K. Schindler. Revisiting 3d geometric models for accurate object shape and pose. In *3dRR11*. 2, 6, 7