Detailed 3D Representations for Object Recognition and Modeling

M. Zeeshan Zia, Michael Stark, Bernt Schiele, and Konrad Schindler

Abstract—Geometric 3D reasoning at the level of objects has received renewed attention recently, in the context of visual scene understanding. The level of geometric detail, however, is typically limited to qualitative representations or coarse boxes. This is linked to the fact that today's object class detectors are tuned towards robust 2D matching rather than accurate 3D geometry, encouraged by bounding-box based benchmarks such as Pascal VOC. In this paper, we revisit ideas from the early days of computer vision, namely, detailed, 3D geometric object class representations for recognition. These representations can recover geometrically far more accurate object hypotheses than just bounding boxes, including continuous estimates of object pose, and 3D wireframes with relative 3D positions of object parts. In combination with robust techniques for shape description and inference, we outperform state-of-the-art results in monocular 3D pose estimation. In a series of experiments, we analyze our approach in detail, and demonstrate novel applications enabled by such an object class representation, such as fine-grained categorization of cars and bicycles according to their 3D geometry, and ultra-wide baseline matching.

Index Terms—3D Representation, recognition, single image 3D reconstruction, scene understanding, ultra-wide baseline matching

1 Introduction

Over the last decade, automatic visual recognition and detection of semantic object classes have made spectacular progress. It is now possible to detect and recognize members of a semantic object categories with reasonable accuracy. Based on this development, there has been a renewed interest in high-level vision and scene understanding, *e.g.* [26], [13], [59], [24], [21], [5], [60].

The present work starts from the observation that although modern object detectors are very successful at finding things, the object hypotheses they output are in fact extremely crude: typically, they deliver a bounding box around the object in either 2D image space [58], [12], [16] or 3D object space [34], [24], [43]. That is, the detected object is represented by a box, which differs from other objects only by its size and aspect ratio. We believe that such simplistic object representations severely hamper subsequent higher-level reasoning about objects and their relations, since they convey very little information about the objects' geometry.

We thus try to take a further step towards the ultimate goal of scene-level image understanding, by looking back at ideas from the early days of computer vision. Starting from Marr's seminal ideas [39], many 3D models of objects were proposed, which provided rich and detailed descriptions of object shape and pose [8], [44], [38], [27], [53], [22]. Unfortunately, these models proved difficult to match to real world images. As a consequence, later researchers traded off model accuracy for robustness in matching, for example by representing objects by the statistics of local features in an image window. This has led to impressive performance for recognition of a variety of object classes [14] as well as related tasks like scene classification [28], but the extent to which relations between scene entities can be modeled with such representations is rather limited. Also, we note that the recognition performance of 2D appearance representations at present is showing only small improvements and seems to be saturating (*e.g.* at ≈35% average precision for the well-known PASCAL VOC challenge [14]). Although *per se* this does not mean that more complex models are the way to go, it does raise the question whether

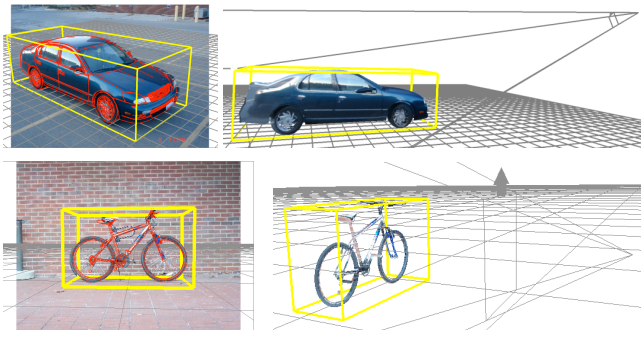


Fig. 1. Fully automatic shape and pose estimation results. (Left) overlaid closest training 3D CAD model. (Right) reconstruction of object shape, pose, and camera pose (CAD model rendered from novel viewpoint using original image as texture).

M. Z. Zia and K. Schindler are with the Photogrammetry and Remote Sensing Laboratory at ETH Zurich. E-mail: {mzia, konrads}@ethz.ch
M. Stark is with the Department of Computer Science at Stanford University. E-mail: mst@stanford.edu
B. Schiele is with the Computer Vision and Multimodal Computing Laboratory at MPII Saarbrücken. E-mail: schiele@mpi-inf.mpg.de

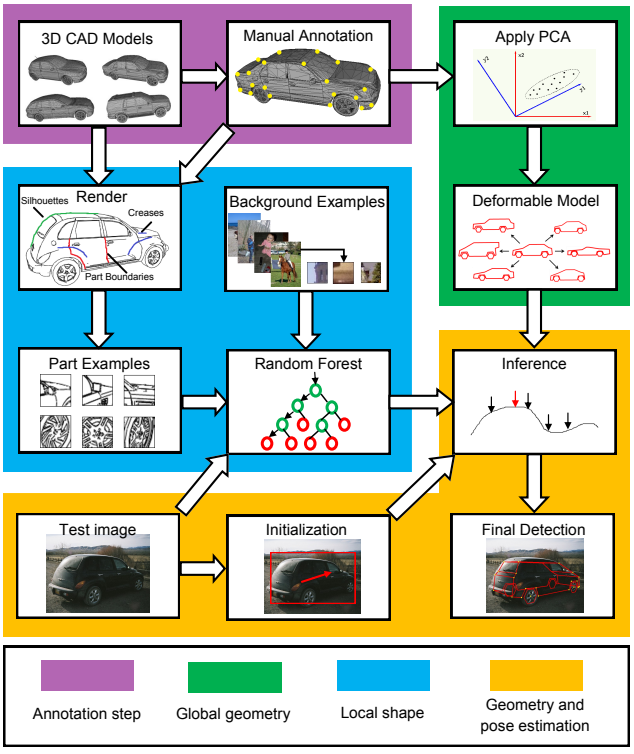


Fig. 2. Full system diagram.

some of the difficulties could be overcome with 3D models, which allow one to segment, reconstruct, and recognize in a more integrated fashion.

Over the last couple of years researchers have explored coarse "box-level" representations of 3D geometry in the context of scene understanding [26], [13], [59], [24], [21], [5], [60], and have shown that 3D geometric reasoning is not only interesting as a goal in itself, but that the additional information it supplies also leads to better recognition performance. In this work, we try to go one step further. Inspired both by early work on 3D recognition and by more recent advances in 2D appearance descriptors, we combine detailed models of 3D geometry with modern discriminative appearance models into a richer and more fine-grained object representation.

Using a 3D model naturally affords invariance to viewpoint. While viewpoint-invariant detection has been a hot topic for some time now [49], [55], [62], [42], [3], [64], [51], [20], [43], [19], [57], [45], most approaches are made up of several flat viewpoint-dependent representations connected together in one way or the other. There are some more recent works which model the 3D geometry more explicitly [6], [34], [54], [9], [45]. While these are an important step towards true 3D recognition, they typically still deliver 2D or 3D bounding boxes as output, and there is still room for improvement in the granularity of the output hypotheses.

System overview. We exploit the fact that for many important classes there are already high-quality 3D models available, and start from a database of 3D computer aided design (CAD) models of the desired object class as training data. After simplifying the raw CAD models we apply principal components analysis to obtain a coarse 3-dimensional wireframe model which captures the geometric intra-class variability. In order to capture appearance, we train detectors for the vertices of the wireframe, which we call "parts". The training is also based on renderings of the (original, unsimplified) CAD models, such that our model does not require any image annotation. We apply the model to two rather different object classes, cars and bicycles.

At test time, we generate evidence for the parts by densely applying the part detectors to the test image. We then explore the space of possible object geometries and poses by guided random sampling from the shape model, in order to identify the ones that best agree with the image evidence. The system is schematically depicted in Fig. 2.

Contributions. The paper makes the following contributions. (i) we show that for certain object types classical 3D geometric object class representations better fulfill the requirements of detailed visual modeling, and deliver object hypotheses with much more geometric detail than current detectors (see Fig. 1). We believe this geometric richness is an important ingredient for scene-level geometric reasoning. (ii) we demonstrate that a 3D model enriched with local appearance descriptors can accurately predict 3D object pose and shape from single still images. In particular, our model improves over state-of-the-art results for pose estimation on a standard multi-view dataset. (iii) we show the benefit of detailed geometric category models for a geometric modeling task, namely ultra-wide baseline matching, where we successfully recover relative camera pose over viewpoint changes up to 180°, again improving over previous work. And (iv) we give experimental results on predicting more fine-grained object categories (different types of cars and bicycles) based solely on the inferred 3D geometry.

Parts of this work have appeared in a preliminary conference paper [66]. The present paper introduces an appearance model based on random forests which is both more accurate and much more efficient, a modified objective function for model-to-image matching, and improved and extended experimental results, including the addition of the challenging bicycle class.

The remainder of this paper is structured as follows. Sect. 2 reviews related work. Sect. 3 introduces our 3D geometric object class model. Sect. 4 gives experimental results, and Sect. 5 concludes the paper with an outlook on future work.

2 Related Work

Our work attempts to recover detailed geometric 3D object representations from single input images. As such, it is related to 3D geometric modeling from the

earlier days of computer vision, more recent advances in scene understanding, and multi-view object class recognition, each of which we review in the following.

Early 3D modeling. Geometric modeling in 3D used to be an important component of visual object recognition, from the inception of computer vision until about the mid 1990ies. Many systems [46], [8], [44] were proposed which built complex shapes from simpler primitives, such as polyhedra [46], generalized cylinders [8], and super-quadrics [44]. With these primitives, single objects as well as entire scenes were represented. Alternatively, salient local parts of the 3D shape, such as triplets of line segments, were matched to their image projections [38]. Hand-crafted, rigid 3D models were proposed to track vehicles in scenes with static background [22], [27], later extended to deformable models [53].

Unfortunately, while these models provided rich descriptions of objects and scenes, robustly matching them to cluttered real-world images proved to be exceedingly difficult at the time. Thus, later research abandoned them in favor of less expressive, but more robust 2D models. These include sparse sampling at locally confined regions of interest [1], [11], [29]; modeling the spatial relationship between these regions at different levels of detail [17], [16] (or not considering such relations at all [11]); and densely sampling (usually gradient-based) features from the object's extent in 2D [12].

Recent 3D modeling. With the advent of powerful computers and advances in machine learning, it has become feasible to revisit some of the classical ideas of 3D object modeling. In the context of indoor scene understanding, [59] proposes a method to infer the 3D layout of the walls and segment out the clutter objects, and [24] shows that such 3D modeling not only provides a better interpretation of the scene, but also improves 2D object detection performance. Along the same lines, [26] models interactions between objects, surface orientations, and 3D camera viewpoint for outdoor scene understanding, and demonstrates improved performance in object detection. [21] takes into account qualitative geometric and mechanical properties of objects and model their relationships, in order to generate a qualitative 3D interpretations of outdoor scenes. Similarly, pedestrian and vehicle tracking from mobile platforms has been demonstrated to benefit from 3D reasoning [13], [60].

Inspired by this comeback of 3D scene understanding, our work aims to furnish the underlying representations with a lot more geometric detail [66]. By combining a deformable 3D shape model with powerful local descriptors, we obtain more detailed and more expressive object class models, that directly lend themselves to detailed 3D reasoning about object and scene geometry. Recent works with similar ambitions as ours are [61], [25]. An object is represented in [61] as a collection of a few planar segments in 3D space called "aspect parts" (*e.g.* one planar "aspect" for the bicycle class, six for the car class). Like us they train on 3D CAD models, manually defining the aspect parts for different object categories. Geometric relations are represented in a similar way as in [48], whereas pose is represented by a discrete set of viewpoints. In [25], a 2D part-based object model predicts the location of land marks, which is lifted to 3D in a second stage by fitting a coarse 3D model to these land marks with non-rigid SfM. In our work, we go even further in terms of 3D detail and predict in a continuous pose space. In another paper [65], we apply our representation to explicitly model occlusions.

Multi-view recognition. A closely related problem to ours is multi-view recognition, which has received a lot of interest in recent years. The most frequently used approach for that task are banks of viewpoint-specific detectors [49], [42], [64], [51], [57], [43], [45]. Other approaches, while still relying on several flat, viewpoint-specific representations, establish connections between viewpoints via homographies [62], probabilistic morphing of object parts [52], discriminative mixtures of global templates [20], or by feature tracking with integrated single-view codebooks [55]. One step further towards true 3D recognition are models with rigid 3D configurations of local 2D features [35], [3], [19].

Similar to the renewed trend of 3D modeling on the scene-level, attempts are recently being made to explicitly represent 3D object class geometry alongside appearance. A coarse, volumetric blob model is learned from 3D CAD data in [34], and combined with 2D appearance models, which have been learned from annotated real-world images. The implicit shape model [29] is augmented in [54] with the relative depth between codebook entries, obtained from a structured light system. [45] extend the deformable part model (DPM) of [16] to include coarse viewpoint estimates in a structured prediction framework, and enforce part correspondences across viewpoints by 3D constraints.

While these approaches internally capture 3D object class geometry to some degree, they typically still provide 2D bounding boxes and coarse viewpoint labels as their output, and do not guarantee that the local parts are localized correctly. In contrast, our method generates complete hypotheses of 3D object geometry, including continuous viewpoint estimates with 5 degrees of freedom.

Efficient part detection. As the number of object classes and viewpoints increases, the computational cost for appearance-based detection grows significantly. Several attempts have been made to solve this problem by sharing information between object classes on different levels, *e.g.* [47]. Random Forests [7] provide a natural way to perform classification with multiple classes, and allow sharing at the level of weak learners inside the algorithm. They have suc-

cessfully been used to train detectors for interest points [32], [30]. In our experience, random forests also handle multi-modal distributions rather well. We use a single multiclass random forest classifier with one class per object part, combining examples from many different viewpoints in each class.

3 3D Geometric Object Class Model

Decomposing object class representations into separate components for global layout and local appearance is a widely accepted paradigm in object class recognition [17], [16]. Its main advantages are the ability to account for variations in object shape better than rigid template models, and robustness to partial occlusion. The paradigm is often implemented by optimizing a smooth, continuous function of the global layout at recognition time, *e.g.* in the form of tree-structured [16] or fully connected [17] Gaussian densities over part positions. While these approaches have efficient implementations and have proven robust in terms of image matching, the resulting object hypotheses are hard to interpret and reason about in terms of geometry: deviations from geometrically plausible layouts are merely penalized, but not rendered impossible, and in fact individual parts are misplaced rather frequently.

Since we aim to not only detect the object, but also recover its geometry, we choose a different route and generate only geometrically valid hypotheses to start with. In a second step, we then verify that the generated hypotheses are supported by sufficient image evidence, a strategy sometimes termed *hypothesize-and-verify*, or *sample-based inference*.

We model an object class as a 3D wireframe representing global layout, with attached local appearance representations of object parts. Like several other recent works in multi-view recognition we leverage synthetic training data besides real-world images [34], [51], [66], [45], and learn both shape and appearance from a collection of 3D computer aided design (CAD) models, thereby ensuring consistency between global layout and local part models by design. At recognition time, we establish the connection between the 3D wireframe and the 2D image by means of a projective transformation, which is part of the object hypothesis. The transformation could potentially be shared among multiple objects in the same scene, however this is not further explored here.

3.1 Global geometry representation and learning

Our global geometry representation is given by a deformable 3D wireframe, which we learn from a collection of exemplars obtained from 3D CAD models. More formally, a wireframe exemplar is defined as an ordered collection of n vertices, residing in 3D space, chosen from the set of vertices that make up a 3D CAD model. In our current implementation the topology of the wireframe is pre-defined (manually defined for each object class, similar to [61]) and its vertices are chosen manually on the 3D CAD models. In the future, they could potentially be obtained using part-aware mesh segmentation techniques from the computer graphics literature [50].

We follow the classical "active shape model" formulation of point-based shape analysis [10], and perform PCA on the resulting (centered and rescaled) vectors of 3D coordinates. The final geometry representation is then based on the mean wireframe μ plus the m principal component directions $\mathbf{p}_j$ and corresponding standard deviations σ_j, where $1 \leq j \leq m$. Any 3D wireframe $\mathbf{X}$ can thus be represented, up to some residual ϵ, as a linear combination of r principal components with geometry parameters $\mathbf{s}$, where s_k is the weight of the k^{th} principal component:

$$\mathbf{X}(\mathbf{s}) = \mu + \sum_{k=1}^{r} s_k \sigma_k \mathbf{p}_k + \epsilon \tag{1}$$

Example 3D wireframe models for cars and bicycles are shown in Fig. 3. Please note how principal directions represent the diversification of cars into sedan, SUV, sports car, and compact car, and of bicycles into mountain bike, racing bike, and children's bike. In our experiments, we show that we can in fact recover these fine-grained vehicle categories by fitting the model to single input images (Sect. 4.6).

3.2 Local shape representation

In order to match the 3D geometry representation to real-world images, we train a distinct part shape detector for each vertex in the wireframe, for a variety of different viewpoints. This is in contrast to early approaches relying on the matching of discrete image edges to model segments [27], [53], [22], which has proven to be of limited robustness in the face of real-world image noise and clutter.

Following [51], [66], we employ sliding-window detectors, searching over image locations and scales, using a dense variant of shape context [2] as features. For each wireframe vertex, a detector is trained from vertex-centered patches of non-photorealistic renderings of our 3D CAD models (Fig. 4). Despite the apparent difference from real-world appearance, this particular combination of edge-based rendering and shape feature has shown to generalize well from rendered to real-world images [51], [45]. Rendering positive training examples further has the advantage of being able to generate massive amounts of artificial training data from arbitrary viewpoints. Following [51], [66], [45], we render three different types of edges: crease edges, which are inherent properties of a 3D mesh, and thus invariant to the viewpoint, part boundaries, which mark the transition between semantically defined object parts and often coincide with creases, and silhouette edges, which describe the

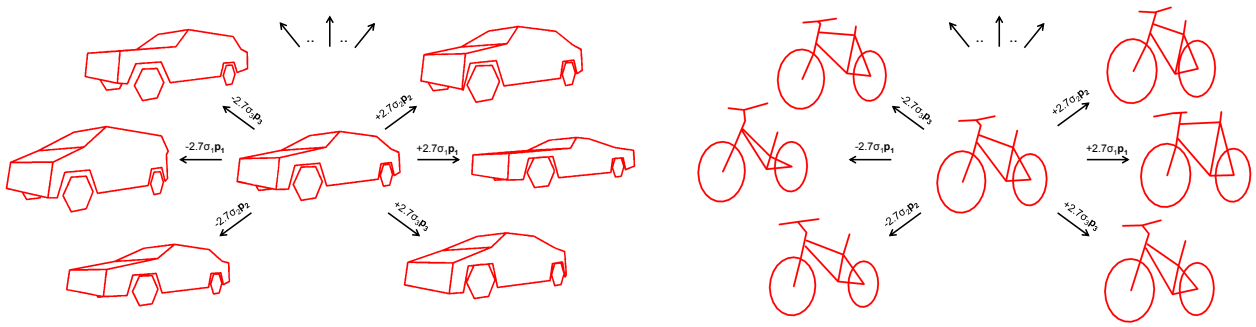


Fig. 3. Coarse 3D wireframe representations of cars (left) and bicycles (right). Modes of variation along the first three principal component directions.

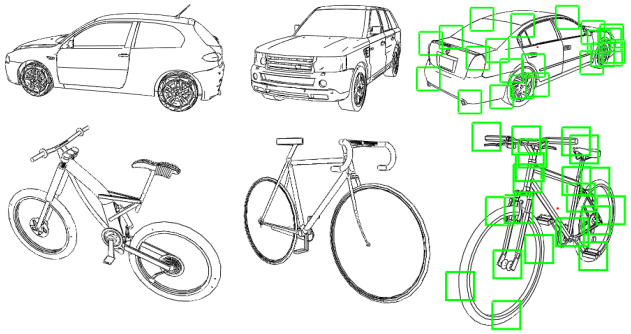
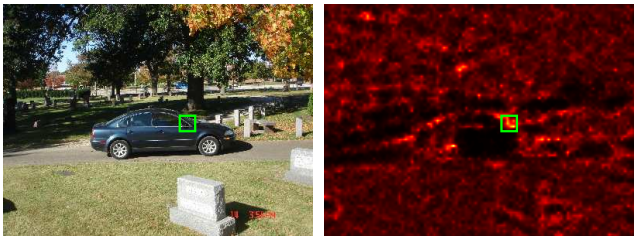


Fig. 4. Non-photorealistic renderings for local part shape detector training, cars (top), bicycles (bottom). Green boxes denote positive training examples.



(a) (b)

Fig. 5. Random forest detection map for one car part. (a) Test image and ground truth part, (b) detection map. Brighter shade corresponds to higher likelihood.

viewpoint-dependent visible outline. Negative training data is obtained by sampling random patches from a set of real-world background images set, as well as random patches from rendered images in the vicinity, but not on the parts of interest. The latter is important in order not to bias the part detectors to label all photorealistic patches as background, and also improves localization accuracy of the detectors.

3.3 Discriminative part detection

As local part detectors, we use discriminative classifiers trained for a discrete set of viewpoints, specified by azimuth and elevation angles. We explore two different variants, namely individual binary AdaBoost [18] classifiers per part and viewpoint, and a monolithic multi-class random forest [7] per object class. As we show in our experiments (Sec. 4.3), random forests prove favorable w.r.t. runtime while maintaining the same part localization performance, which is why all following results in Sect. 4 are based on random forests.

AdaBoost. In this variant, we train for each part and each viewpoint an individual binary AdaBoost classifier, which discriminates that particular part in that particular view from the background. Such a strategy has been employed successfully for people detection in [2], and in our previous work [66].

Random forest. In an attempt to reduce the massive amount of detectors arising from the cross product of parts and viewpoints, we make two modifications to the above scheme. First, we replace the binary classifiers by a single multi-class classifier with one class per part (plus one for the background). We choose random forests [7], since they have been shown to deliver excellent performance for multiclass problems with complex class-conditional distributions. Second, we leverage the ability of random forests to model multi-modal distributions, and combine all training examples into a single class that depict the same part at any viewpoint. That is, we train a single viewpoint-invariant random forest, which distinguishes between parts, irrespective of the viewpoint.

In the individual nodes of the decision trees, we use *oblique* splits that decide based on random hyperplanes of a larger number of randomly chosen dimensions [40], as opposed to the more commonly used *axis-aligned* (or *orthogonal*) splits, where node decisions are based on a single feature dimension. Oblique splits increase the discriminative power in connection with high-dimensional features, such as our dense shape context features. Furthermore we use the ratio between the part-conditional distribution and the background as final part detection score, as in [17], [57]. Fig. 5(b) gives a random forest detection map for the car part of Fig. 5(a).

Our quantitative evaluation indicates that the detection maps from random forests, although more diffuse due to the marginalization over viewpoints, provide

a better tradeoff between discrimination and recall when used in combination with the global geometry model (Sect. 4.3).

3.4 Viewpoint-invariant shape & pose estimation

During recognition, we seek to find an instance of our 3D geometric model that best explains the observed image evidence. This is formulated as an objective function defined over possible configurations of the model as well as its projection to the test image. It is worth noting that this entails a search over continuous 3D geometry and viewpoint parameters rather than switching or interpolating between flat viewpoint-dependent representations as in previous work [55], [52], [51], [45].

More formally, we denote a recognition hypothesis as $\mathbf{h} = (\mathbf{s}, f, \boldsymbol{\theta}, \mathbf{q})$. It comprises object geometry parameters $\mathbf{s}$ (see Sect. 3.1), camera focal length f, spherical viewpoint parameters for azimuth and elevation $\boldsymbol{\theta} = (\theta_{az}, \theta_{el})$, and image space translation and scale parameters $\mathbf{q} = (q_x, q_y, q_s)$. For perspective projection we assume a simplified projection matrix P that depends only on f, $\boldsymbol{\theta}$, and $\mathbf{q}$. It is composed of a camera calibration matrix $\mathsf{K}(f)$ and a rotation matrix $\mathsf{R}(\boldsymbol{\theta})$, and projects wireframe vertices $\mathbf{X}_j(\mathbf{s})$ to image coordinates $\mathbf{x}_j$:

$$\begin{aligned} &\mathsf{P}(f, \boldsymbol{\theta}, \mathbf{q}) = \mathsf{K}(f) \begin{bmatrix} \mathsf{R}(\boldsymbol{\theta}) & -\mathsf{R}(\boldsymbol{\theta})\mathbf{q} \end{bmatrix} \\ &\mathbf{x}_j = \mathsf{P}\mathbf{X}_j(\mathbf{s}) \ . \end{aligned} \tag{2}$$

For recognition, we want to find the maximum a-posteriori estimate

$$\hat{\mathbf{h}} = \arg\max_h \left[\mathcal{L}(\mathbf{h}) + \lambda \mathcal{Q}(\mathbf{h}) \right] \ , \tag{3}$$

where $\mathcal{L}(\mathbf{h})$ is the data likelihood term and $\mathcal{Q}(\mathbf{h})$ is a shape prior (regularizer).

Data likelihood and shape prior. The inference in our framework (see below) is based on sampling part configurations from the explicit 3D model (1) and scoring them. In such a model-driven approach only globally plausible shapes are ever generated, which allows for a relatively simple data likelihood (compared to approaches where the part locations can move independently in a data-driven manner [51]).

We define the (log-)likelihood of an object instance being present as a sum over the likelihoods of its constituent parts, assuming conditional independence between them. The likelihood $S_j(\boldsymbol{\varsigma}, \mathbf{x}_j)$ of part j being present at any given image location $\mathbf{x}_j$ and local scale $\boldsymbol{\varsigma}$ has already been estimated by the part detector (Sect. 3.3). Following [57] we normalize the part likelihood by the background likelihood $S_b(\boldsymbol{\varsigma}, \mathbf{x}_j)$ at the same location. In order to account for object-level self-occlusion, only parts that are visible in the putative projection are considered, leading to binary indicator functions $o_j(\mathbf{s}, \boldsymbol{\theta})$ for the visibility. Finally the likelihood is re-normalized to the number of visible parts. The complete data term then reads

$$\mathcal{L}(\mathbf{h}) = \max_{\boldsymbol{\varsigma}} \left[\frac{1}{\sum_{j=1}^{m} o_j(\mathbf{s}, \boldsymbol{\theta})} \sum_{j=1}^{m} o_j(\mathbf{s}, \boldsymbol{\theta}) \ log \frac{S_j\big(\boldsymbol{\varsigma}, \mathsf{P}\mathbf{X}_j(\mathbf{s})\big)}{S_b\big(\boldsymbol{\varsigma}, \mathsf{P}\mathbf{X}_j(\mathbf{s})\big)} \right] \tag{4}$$

The PCA model (1) implies a zero-mean multivariate Gaussian distribution of the shape parameters around the mean shape. Consequently we introduce a shape prior which penalizes deviations from the mean 3D shape of the object class according to

$$\mathcal{Q}(\mathbf{h}) = \sum_{k=1}^{r} log \ \mathcal{N}(s_k; 0, 1). \tag{5}$$

To avoid overly unlikely shape hypotheses from the extreme tails of the Gaussian we limit the shape parameters to the range $|s_k| < 3$, such that they cover 99.7% of the shape variation observed in the training set.

Inference. The objective (3) cannot be easily maximized, since the data term is highly non-convex and—due to the binary $o_j(\mathbf{s}, \boldsymbol{\theta})$—also not smooth. We thus resort to a stochastic hill-climbing method. To account for the multi-modality of the posterior we generate multiple starting points ("particles") $\{\mathbf{h}_m^n\}$ with corresponding objective values $L(\mathbf{h}_m^n) + \lambda \mathcal{Q}(\mathbf{h}_m^n)$, and iteratively improve them through stochastic search. Each particle $\mathbf{h}_m^n$ corresponds to a distinct set of values in the space of object hypotheses $\{\mathbf{s}, \boldsymbol{\theta}, \mathbf{q}\}$, with m being the particle index and n the iteration.[1] The initial set of particles is drawn from a uniform distribution for the unknown shape parameters, whereas the parameters for location and pose are based on the initialization. In every iteration the particles are then updated to increase their objective value (3). Instead of computing gradients, semi-local update steps are determined by random sampling, which copes better with weak local minima and avoids problems due to visibility changes: for each particle a number of candidates $\{\tilde{\mathbf{h}}_m^{n+1}\}$ are generated by drawing new values for the individual parameters h_m from Gaussians centred at the current values,

$$\tilde{h}_m^{n+1} \sim p(\tilde{h}_m^{n+1} | h_m^n) = \mathcal{N}\big(h_m^n, \sigma_h^2(n)\big) \ . \tag{6}$$

Among the candidates the one with the highest likelihood replaces the original particle, thus yielding a new particle set $\{\mathbf{h}_m^{n+1}\}$. The variances $\sigma_h^2(n)$ of the proposal distributions are successively reduced according to an annealing schedule, for faster convergence. After the last iteration the particle with the highest weight is kept as MAP-solution $\hat{\mathbf{h}}$. Although the underlying posterior distribution may be very complicated, hill-climbing with simple Gaussian perturbations works well in practice. This procedure

[1] f is held fixed in our experiments, assuming that the perspective effects are similar for all images.

is similar to [31] (per particle), except that instead of computing the variances as a function of drawn samples, we choose them according to a pre-defined schedule. While this means that each of our particles might get stuck at local optima, keeping of multiple particles allows choosing the best one among them as well as keeping extra locally optimal hypotheses for a future scene-level reasoning stage.

Initialization. Rather than running inference blindly over entire test images, we start from promising image positions, scales, and viewpoints, which we obtain in the form of predicted object bounding boxes from a conventional 2D multi-view detector. In particular, we use the recently proposed multi-view extension of the deformable part model by Pepik et al. [45], which has been shown to yield excellent performance w.r.t. both 2D bounding box localization and coarse viewpoint classification. Specifically, we initialize q_x and q_y inside of a predicted object bounding box, and q_s according to the bounding box size. Similarly, we initialize the viewpoint parameters $\boldsymbol{\theta}$ according to the coarse viewpoint predicted by the detector. Due to the highly non-convex nature of the problem the overall system performance is strongly influenced by the initialization quality (Sect. 4.2).

4 Experimental evaluation

In the following, we carefully analyze the performance of our 3D object class model in a series of experiments, focusing on its ability to provide detailed 3D object geometry. To that end, we evaluate its performance in four different tasks, comparing to results reported by prior work where appropriate.

(i) first we evaluate the ability to accurately predict the locations of individual object parts in the 2D image plane (Sec. 4.3). In the context of 3D scene understanding, this ability is important in order to establish geometric relations between different scene entities, such as an object touching the ground plane at a specific location. (ii) we evaluate the ability to recover the full 3D pose of recognized objects (Sec. 4.4). In contrast to most prior work, we report results for both coarse *viewpoint classification* and continuous 3D *pose estimation* with 5 degrees of freedom (pictures are assumed to be upright, without in-plane rotation). In either case, we achieve results en par with or better than previous work. (iii) we evaluate our object class representation in the context of a 3D scene modeling task, namely to recover relative camera pose from wide-baseline pairs of images depicting the same object (Sec. 4.5). Here, the model is challenged to recover consistent 3D object geometries across different viewpoints, and improves over previously reported results for all baselines, up to 180°. (iv) we leverage the detailed 3D shape hypotheses provided by our approach for fine-grained object categorization based on geometric shape (Sect. 4.6).

4.1 Setup

We commence by describing the experimental setup w.r.t. test and training data, random forest training, inference, and initialization.

Test datasets. The evaluation is based on the *3D Object Classes* [48] and *EPFL Multi-view cars* [42] datasets, which both have been designed specifically for multi-view recognition. These datasets constitute a suitable trade-off between controlled conditions for experimentation and challenging real-world imagery. Our focus is on the object classes *car* and *bicycle*. The *3D Object Classes* test set depicts 5 object instances from 8 different azimuth angles, 3 distances, and 2 (cars) or 3 (bicycles) elevation angles, against varying backgrounds, amounting to a total of 240 cars and 360 bicycle test images. The *EPFL Multi-view cars* test set comprises 10 different car models with largely varying shape, rotating on a platform, with a sample every 3 to 4 degrees, totaling to about 1000 images. Fig. 10 and 11 show qualitative results obtained by our method on images of these data sets.

Synthetic training data. In all experiments, we use 38 commercially available 3D CAD models of cars[2] and 32 freely available CAD models of bicycles[3] for training. We annotate 36 model points for cars and 21 for bicycles (Fig. 4) in order to train both global geometry (Sec. 3.1) and local part shape (Sec. 3.2). Each part is rendered from 72 different azimuth ($5°$ steps) and 2 elevation angles ($7.5°$ and $15°$ above the ground) for cars, respectively 3 elevation angles ($7.5°$, $15°$, and $30°$) for bicycles, densely covering the relevant part of the viewing sphere (the bicycle test set covers a larger range of viewpoints). CAD models are rendered using the non-photorealistic style of [51], [45]. Rendered part patches serve as positive examples, randomly sampled image patches as well as non-part samples from the renderings serve as negative examples. The total number of training patches is 140,000 per class, evenly split into positive and negative ones.

Random forest training. As part detectors, we train a single random forest classifier [7] for each object class (one for bicycles and one for cars), distinguishing between the parts of interest (36 for cars, 21 for bicycles) and background. In both cases the random forests have 30 trees with a maximum depth of 13. Node tests are given by random hyperplanes of dimensionality 200 (chosen from a total of $3,500$ dimensions of the shape context descriptor), which for our high-dimensional input we found empirically to deliver much higher performance than the more commonly used single dimension node tests.

Inference. We sample θ_{az} over a continuous range of $20°$ centered around the initialization and θ_{el} from ground level to $20°$ for cars and $30°$ for bicycles. For part detections, we consider the maximum score in a

[2] www.doschdesign.com

[3] www.sketchup.google.com/3dwarehouse/

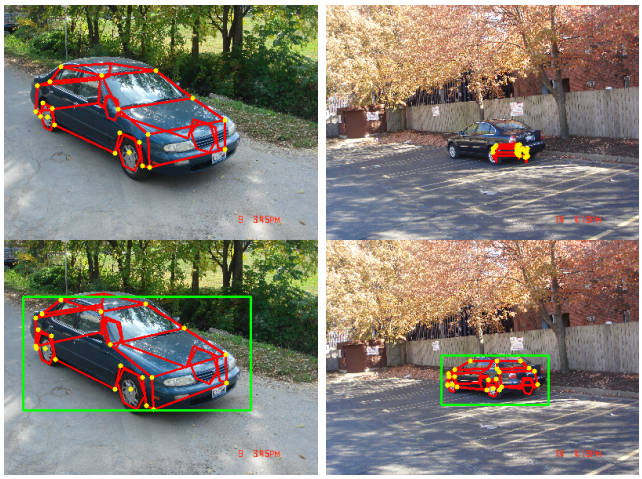


Fig. 6. Example detections without (top row) and with informed initialization [45] (bottom row).

scale range of $\pm 30\%$ around the bounding box scale.

Initialization. We report results for two different, informed initializations of our model, as well as results obtained by running our model from random starting points, not using any prior information about object location and pose (Sect. 4.2).

The first initialization is provided by the state-of-the-art multi-view detector [45], providing almost perfect 2D bounding box localization on the *3D Object Classes* dataset for cars and bicycles (97.5% average precision each). Specifically, we use the multi-view DPM referred to as DPM-VOC-VP in [45], trained from the respective car and bicycle training sets provided by the *3D Object Classes* and *EPFL Multi-view cars* data sets [42]. In the following, we refer to the combination of this initialization and our model as the *full system*, since it constitutes a fully automatic procedure that infers detailed 3D geometric hypotheses from input images, as it would be used in a real-world application.

The second initialization (termed *GT*) aims at providing a best case evaluation of our model isolated from the effects of the multi-view DPM, starting from annotated ground truth bounding boxes and coarse viewpoint estimates.

4.2 Recognition without initialization

We commence by exploring the performance of our approach in isolation, independent from any informed initialization, by running it from a number of randomly selected starting points (250 particles drawn uniformly at random from the location, pose, and shape parameter space). We evaluate over the car class in the *3D Object Classes* dataset. Considering the highest scoring hypothesis in each of the 240 test images, we are able to localize the correct 2D bounding box in 51.7% of the cases (according to Pascal criterion [14]). Further, following the experimental protocol of [52], the viewpoint of these true positive detections is correctly classified into 8 different azimuth angle classes (*left*, *front-left*, *front*, *front-right*, *right*, *back-right*, *back*, *back-left*) in 66.9% of the cases.

Although these numbers are encouraging, running our detailed 3D geometric model blindly over entire test images is obviously inferior to current state-of-the-art object class detectors, both w.r.t. 2D localization performance and computational complexity. In the following, we thus provide our model with *informed initializations* in the form of rough 2D object locations and poses (*full system*), obtained by a 2D multi-view detector [45]. This cascaded approach drastically reduces computation, and results in state-of-the-art performance in pose estimation (Sect. 4.4). Fig. 6 compares example detections obtained with and without informed initialization.

Please note that other recent work [33] on deformable object models also relies on initializing models within a small operating window centered around the object (much like our *GT* initialization), and even assumes fixed object scale.

4.3 Part localization

One way of performing accurate geometric reasoning on the scene-level is to have object class models that provide well-defined anchor points, so as to geometrically relate them to other scene entities. Consider for example the wheels of a vehicle, which can be assumed to rest on a supporting surface, and can hence provide hints on the likely position and orientation of a ground plane. Likewise, localizing extremal points on the vehicle body (such as bumper corners) can help to assess the area of covered ground and hence its 3D extent in the scene.

Since the parts in our model are chosen to correspond to well-defined regions of an object's anatomy (Sect. 3.1), we can evaluate the ability of our model to localize these parts individually. To that end, we annotate the 2D locations of all visible parts in our test images. We have made all annotations publicly available[4].

Protocol. We measure part localization accuracy as the fraction of correctly localized parts of a specific type across test images, restricted to those test images where the method under consideration delivers a true positive detection in terms of the Pascal criterion [14] on the 2D bounding box. A part is considered correctly localized if its estimated 2D position deviates less than a fixed number of pixels from annotated ground truth, relative to its estimated scale. For instance, for a car side view at scale 1.0, covering 460×155 pixels, that number is 20, which amounts to localizing a part to within $\approx 4\%$ of the car length. The same criteria is used for bicycles. Note that this strict

[4] http://www.igp.ethz.ch/photogrammetry/downloads

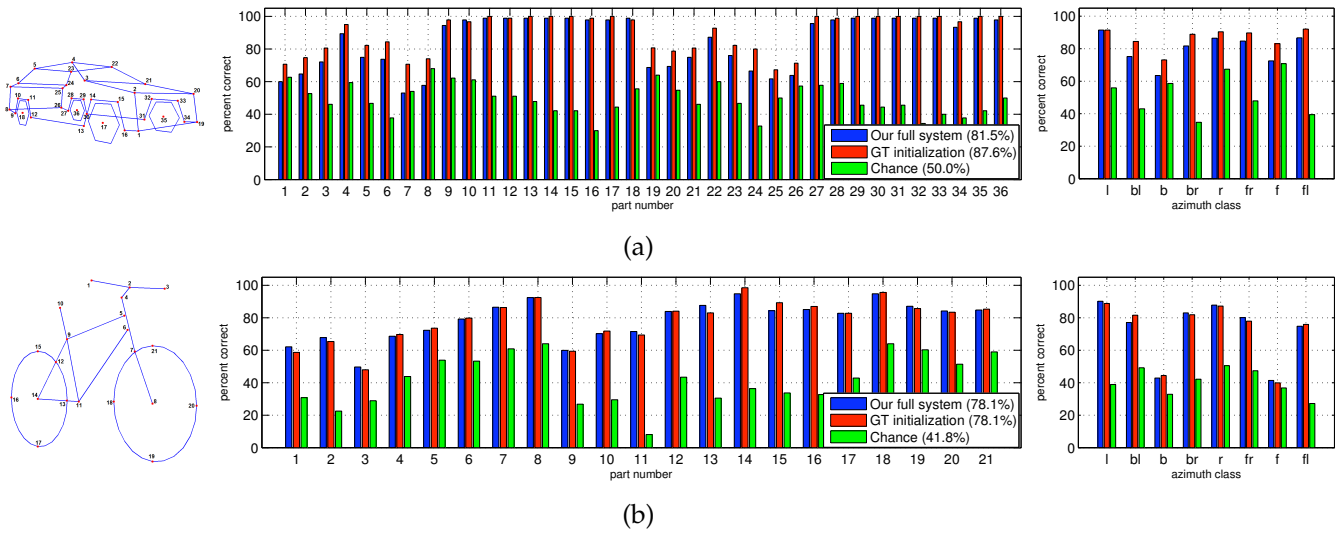


Fig. 7. Part localization results on *3D Object Classes*. Part numbering schemes (left), localization performance for individual parts (center) and viewpoints (right), for (a) bicycles, and (b) cars.

criterion is applied in all cases, even for hypotheses with grossly wrong viewpoint estimates.

Results. Fig. 7 gives the results for part localization for cars (a) and bicycles (b), averaged over all test images, grouped by individual parts (center bar plots) and viewpoints (right bar plots). We distinguish among the performance of the *full system* (blue bars), our system initialized from ground truth (*GT*,red bars), and a baseline also initialized from *GT*, but using uniform part score maps (*chance*, green bars).

Per-part evaluation. In Fig. 7 (a) and (b) (center), we observe that there are in fact differences in the localization accuracy of different parts. Notably for cars (Fig. 7(a)(center)), parts located in the wheel regions (9-18, 27-36) are localized almost perfectly by both the *full system* and when starting from *GT*. This is not surprising, since wheels provide plenty of local structure that can robustly matched by local part detectors, providing strong guidance for the geometric model. Parts on the front roof (4, 22) can also be localized with great accuracy (89.4% and 87.2% by the *full system*), followed by back roof parts (5 with 74.9% and 23 with 76.0%) and hood parts (3 with 72.1% and 21 with 74.9%). Parts in the trunk region tend to perform worse (7 with 53.0% and 25 with 61.7%). We attribute this difference to the greater flexibility that our learned global geometry model allows in the back: the collection of training CAD models comprises limousines and sports cars as well as SUVs and station wagons.

Bicycles (Fig. 7(b)(center)) appear to be more challenging than cars in general (*GT* performance drops by 9.5% from 87.6% to 78.1%), possibly due to their wire-like nature, which amplifies the influence of background clutter. Concerning the ranking of the parts, we observe a similar trend as for cars: parts located on the wheels (7-8,12-21) have localization accuracy of at least 82.8% for the *full system*, whereas the wheel centers (8,14) even reach 92.4% and 94.8%, respectively. Again, parts that exhibit more variability in the training CAD models perform worse, such as the handle region (1-3, between 49.7% and 67.8%) and the joint below the seat (9 with 59.5%).

On average, we achieve correct part localization in an encouraging 81.5% of all cases for cars using the *full system* (87.6% using *GT*), and in 78.1% for bicycles (for both *full system* and *GT*).

Per-viewpoint evaluation. Fig. 7 (a) and (b) (right) groups the part localization results according to the different azimuth angles of test images, averaged over all parts. For cars (Fig. 7(a)(right)), we observe that part localization performs best for plain side views (left 91.5%, right 86.5%, *full system*), followed by diagonal front (front-left 86.7%, front-right 84.7%) and back views (back-right 81.7%, back-left 75.2%). Plain front (72.5%) and back (63.5%) views perform moderately, apparently due to the absence of the strong evidence provided by the wheels in the other views.

The same tendency can be observed for bicycles (Fig. 7(b)(right)). Plain side views perform best (left 90.2%, right 87.8%, *full system*), followed by the diagonal views (back-right 83.0%, front-right 80.1%, back-left 77.1%, front-left 74.8%) and the plain back and front views (42.9% and 41.4%).

Comparison to AdaBoost [66].

Tab. 1 compares the part localization performance of the *full system* using *random forest* classifiers as part detectors with two variations of AdaBoost, as we previously proposed in [66]. The first variant trains a single binary AdaBoost classifier for each part (36 for cars), azimuth (72), and elevation angle (2),

Classifier type	classifiers trained	class. tested per detection	1 mode	3 modes	post inference
AdaBoost [66]	5,184	36	56.3%	75.5%	79.9%
AdaBoost [66]	5,184	432	**61.4%**	**79.0%**	81.1%
Random forest	36	36	35.0%	57.8%	**81.5%**

TABLE 1
Comparison of part detector performance using random forests and AdaBoost [66] (on cars).

resulting in $5,184$ trained classifiers. At test time, only those classifiers belonging to the coarse viewpoint predicted by the initialization are considered (36 in total). The second variant uses the same set of trained classifiers, but considers neighboring viewpoints at test time as well (432 in total) to account for viewpoint uncertainty.

Tab. 1 gives results for post-inference part localization (*i.e.*, applying the *full system* end to end) as well as pre-inference localization, considering 1 and 3 highest modes in the part detection maps as hypotheses, respectively. When using 3 highest modes we consider a part detection as correct if any one of the modes falls on the ground truth part location. Not surprisingly, we observe that both AdaBoost versions perform much better in pre-inference localization than *random forests* (up to 26.4% for 1 and 21.2% for 3 modes), since the restriction to a narrow range of viewpoints increases the discriminative power of the resulting classifiers. While the inclusion of neighboring viewpoints aids robustness, including all viewpoints (as we do for *random forests*) degrades performance. Post-inference, however, *random forests* have a slight edge (81.5% vs. 79.9% and 81.1%), achieved with two orders of magnitude fewer classifiers (36 vs. $5,184$). This seemingly counter-intuitive behavior stems from the fact that in difficult cases the binary AdaBoost classifiers are sometimes "too convinced" that a part is *not* present, and these false negatives (low part likelihoods at the correct position) drive the inference away from the correct shape.

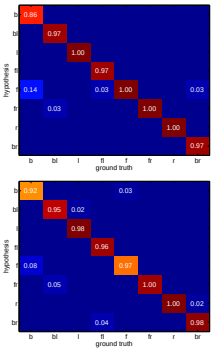
Summary. We conclude that our model yields accurate estimates of the 2D locations of individual parts in the majority of cases, providing a solid basis for 3D geometric reasoning. Since we also observe a non-negligible difference between the results obtained by different initializations (*full system* vs. *GT*), we expect further improvements in response to improved initial detections to start from.

4.4 Pose estimation

In this section, we evaluate the ability of our model to accurately estimate the 3D pose of recognized objects. Even without considering individual parts (as in Sect. 4.3), pose estimation facilitates monocular 3D perception and can provide valuable geometric information for scene-level reasoning. As an example, consider the effect of observing an object, say, a bicycle from different azimuth angles: knowledge

3D Object Classes	cars	bicycles
Liebelt et al. [34]	70.0%	75.5%
Stark et al. [51]	81.0%	-
Zia et al. [66]	84.0%	-
Glasner et al. [19]	85.3%	-
Payet et al. [43]	86.1%	80.8%
Initialization [45]	**97.5%**	**97.5%**
Full system	97.1%	97.1%
GT	98.7%	99.4%

(a)



(b)

Fig. 8. Coarse viewpoint classification on *3D Object Classes*. (a) Accuracies, (b) confusion matrices for cars (top), bikes (bottom), using our *full system*.

EPFL Multi-view	cars
Ozuysal et al. [42]	41.6%
Xiang et al. [61]	64.8%
Lopez et al. [36]	66.0%
Initialization [45]	**76.5%**
Full system	**76.5%**

TABLE 2
Coarse viewpoint classification on *EPFL Multi-view cars*, using our *full system*.

about its 3D shape enables the viewer to estimate the perspective distortion not only of the object itself, but of the entire scene, and thus reason about distances and relations in 3D Euclidean space.

While the focus of our approach lies on providing detailed, continuous 3D pose estimates with 5 degrees of freedom (or even 6, if initialized with an object detector that is invariant to in-plane rotation), we start by reporting results for the popular task of viewpoint classification with 8 and 16 equally spaced viewpoint bins on *3D Object Classes* and *EPFL Multi-view cars*, respectively. In that setting, pose estimation is discretized into a multi-class labeling problem. Since our method relies on coarse viewpoint estimates provided by [45] as an initialization, this evaluation also serves as a sanity check, to ensure that the added expressiveness of our model does not significantly degrade viewpoint classification performance.

Coarse viewpoint classification. Following the experimental protocol of [52], we report results on *3D Object Classes* dataset for the classification of true positive object detections according to 8 different azimuth angle classes (*left, front-left, front, front-right, right, back-right, back, back-left*). Fig. 8(a) gives the corresponding results for cars and bicycles, comparing our *full system* to our system initialized from *GT* bounding boxes, the estimate provided by the *initialization* [45], and results reported in prior work.

In Fig. 8(a), we observe that, for both cars and bicycles, the *initialization* [45] alone already provides almost perfect viewpoint classification (97.5% and 97.5%, respectively), outperforming the next best prior results [43] by margins of 11% and 17%, respec-

3D Object Classes cars	Total Images	True Positives	% Correct Azimuth	Avg. Error Azimuth	Avg. Error Elevation
Stark et al. [51]	48	46	67.4%	4.2°	4.0°
Zia et al. [66]	48	45	73.3%	3.8°	**3.6°**
Initialization [45]	48	**48**	70.8%	**3.4°**	-
Full system	48	47	**95.7%**	3.8°	3.7°
Without init.	48	21	**61.9%**	3.9°	4.8°
GT	48	47	93.6%	3.6°	3.2°

(a)

3D Object Classes bicycles	Total Images	True Positives	% Correct Azimuth	Avg. Error Azimuth	Avg. Error Elevation
Initialization [45]	72	**69**	76.8%	**2.3°**	-
Full system	72	67	**89.6%**	3.4°	**10.4°**
GT	72	69	98.6%	3.2°	8.7°

(b)

TABLE 3
Continuous viewpoint estimation: (a) cars, (b) bicycles.

EPFL Multi-view cars	Total Images	True Positives	% Correct Azimuth	Avg. Error Azimuth
Initialization [45]	994	**981**	73.3%	3.4°
Full system	994	972	**80.3%**	**3.3°**

TABLE 4
Continuous viewpoint estimation (*EPFL cars*).

tively. While our *full system* maintains that high level of performance for both classes (97.1% and 97.1%; compared to [45] we mis-classify the viewpoint of a single car/bicycle), our model initialized from *GT* can further improve to 98.7% for cars and 99.4% for bicycles.

For the *EPFL Multi-view cars* dataset, we perform viewpoint classification into 16 azimuth angle classes as in [42]. The test set contains 10 different car models imaged under fairly poor lighting conditions, thus the performance of most state-of-the-art methods is worse than the results over *3D Object Classes*, as indicated in Tab. 2. Again, the *initialization* [45] already obtains the best viewpoint classification accuracy reported to date. The *full system* again maintains the high level of classification accuracy (76.5 % for both *initialization* and *full system*), though it loses the detections on 9 test images out of 994.

Continuous viewpoint estimation. Since the ground truth of the *3D Object Classes* dataset does not provide accurate viewpoints beyond the eight rough directions, we annotate all images depicting one particular car (48 images) and one particular bicycle (72 images) with continuous azimuth and elevation angles, by manually fitting 3D CAD models to the images. In particular, we start from a CAD model of maximally similar shape, placed on a ground plane, and iteratively adjust the 3D position of the car, the position and orientation of the camera, and its focal length. This procedure is quite time-consuming, but results in precise geometric fits for all images [4].

Tab. 3(a) and (b) give the results for continuous viewpoint estimation, comparing the *full system*, *GT*, and the *initialization* [45], again considering only final true positive detections. For cars (Tab. 3(a)), we also include previously reported results of [51], [66]. A viewpoint estimate is considered correct if it lies within 10° of the annotated ground truth azimuth angle (in contrast to the 45° bins of coarse viewpoint classification). Among those correct estimates, we further measure and report the average angular error in both azimuth and elevation.

Azimuth Diff.	Image Pairs	SIFT [37]	Parts only	Zia [66]	DPM-3D-Const. [45]	*Full system*	*GT*
45°	53	2.0%	30.2%	54.7%	54.7%	**86.8%**	86.8%
90°	35	0.0%	22.8%	60.0%	51.4%	**88.6%**	94.3%
135°	29	0.0%	20.7%	51.7%	51.7%	**65.5%**	89.7%
180°	17	0.0%	0.0%	41.2%	70.6%	**76.5%**	76.5%
Avg.	134	0.5%	18.4%	51.9%	57.1%	**79.4%**	86.8%

TABLE 5
Ultra-wide baseline matching results (cars).

In Tab. 3(a), we observe that our *full system* improves by a remarkable 22.4% over our previous result of 73.3% [66] for cars, amounting to 95.7% viewpoint estimates that are within 10° of the ground truth. At the same time, we improve 24.9% over the *initialization* [45], confirming the ability of our method to provide viewpoint estimates of much finer detail than captured by coarse viewpoint classification. For bicycles (Tab. 3(b)), the improvement over the *initialization* [45] is less pronounced, but still significant (by 12.8% from 76.8% to 89.6%).

Among the correct viewpoint estimates, the actual viewpoint errors for cars are all in the same range. Our *full system* achieves angular errors of 3.8° in azimuth and 3.7° in elevation, which is practically the same as our model starting from *GT* (3.6° and 3.2°). Similar or slightly larger errors are also obtained with competing methods, which however have significantly lower recall, meaning that the "more difficult" cases solved only by our model are nevertheless accurately estimated. Similarly, we achieve 3.4° in azimuth and 10.4° in elevation for bicycles. We attribute the significantly larger elevation errors to the fact that bicycles are largely planar, and thus their elevation angle is rather correlated with the shape (in particular the height-to-length ratio).

Tab. 4 gives the corresponding results for *EPFL Multi-view cars*. Here, cars are depicted from a wide variety of viewpoints sampled densely from the entire 360° viewing circle. In unison with the results on *3D Object Classes*, we improve over the *initialization* [45] by 7%, obtaining precise azimuth angle estimation in 80.3 % of the cases, whereas the average error in azimuth estimation decreases to 3.3°.

4.5 Ultra-wide baseline matching

While the experiments of Sect. 4.3 (part localization) and 4.4 (pose estimation) evaluate our approach from an object-centric perspective, the following experiment quantifies its ability to recover 3D camera and scene geometry. In particular, we consider the task of

estimating relative camera pose from a pair of images depicting the same scene, *i.e.* epipolar geometry fitting. This task quickly gets very challenging as the baseline increases; the best invariant interest point descriptors like SIFT [37] allow matching up to baselines of ≈ 30 degrees in orientation and a factor of ≈ 2 in scale. Only recently, Bao and Savarese [4] have noted that semantic knowledge ("the scene contains a car somewhere") can provide additional constraints for solving the matching problem, increasing the range of feasible baselines. Their approach enforces consistency between 2D object detection bounding boxes and coarse pose estimates across views in a structure-from-motion framework.

In contrast, we leverage the ability of our approach to predict accurate object *part positions*, and use those directly as putative matches. The 3D model is fitted independently to two input images, and the model vertices form the set of correspondences. Matching is thus no longer based on the local appearance around an isolated point, but on the overall fit of the object model. Note, this makes it possible to match even points which are fully occluded. In principle, relative camera pose could be obtained directly from the two object pose estimates. In practice this is not robust, since independent fitting will usually not find the exact same shape, and even in a generally correct fit some parts may be poorly localized, especially if the guessed focal length is inaccurate. Hence, we use corresponding model vertices as putative matches, and robustly fit fundamental matrices with standard RANSAC.

Protocol. As test data we have extracted 134 pairs of images from the car data set, for which the car was not moved w.r.t. the background. The restriction to stable background ensures the comparison is not unfairly biased against SIFT: straight-forward descriptor matching does not need model knowledge and can therefore also use matches on the background, whereas interest points on the cars themselves are rather hard to match because of specularities.

To assess the correctness of the fundamental matrices thus obtained, we manually label ground truth correspondences in all 134 images pairs, on the car as well as the background. A fit is deemed correct if the Sampson error [23] for these points is < 20 pixels.

Results. In Tab. 5, we compare, for varying angular baselines (45°, 90°, 135°, 180°), the results obtained by our *full system* and *GT* to previously reported results (our previous method [66] and the multi-view deformable part model with 3D constraints, DPM-3D-Const. [45]), and two baseline methods: (i) we find putative matches with SIFT (using the default options in [56]); and (ii) in order to assess whether the geometric model brings any benefit over the raw part detections it is based on, we perform non-maximum suppression on the scoremaps and obtain three modes per part in each of the two images. The permutations of these locations form the set of putative correspondences.

(a)

Car cat.	1	2	3	4	5	Total
Full System	**65.9%**	**81.3%**	60.4%	70.8%	**60.4%**	**67.8%**
GT	55.3%	70.8%	**64.6%**	**75.0%**	56.2%	64.4%
Chance	38.9%	30.5%	30.5%	38.9%	38.9%	35.5%

(b)

Bicycle cat.	1	2	3	4	5	Total
Full System	**57.3%**	**62.5%**	**71.2%**	**75.0%**	65.1%	**66.1%**
GT	55.1%	60.9%	68.6%	71.0%	**68.6%**	64.8%
Chance	25.0%	28.1%	40.6%	25.0%	25.0%	28.7%

TABLE 6
Fine-grained categorization of (a) cars , (b) bicycles.

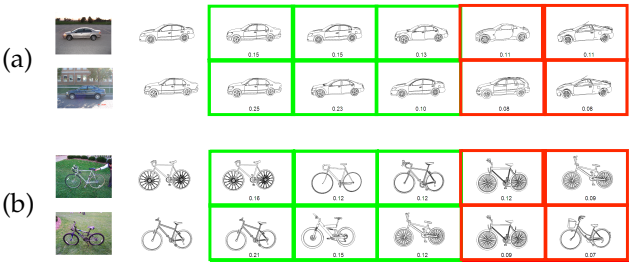


Fig. 9. Fine-grained categorization examples for (a) cars, (b) bicycles. Example input image of true class with corresponding CAD model prototype (left), five most frequently matched CAD model hypotheses (right; green denotes correct, red incorrect matches).

As expected, SIFT catastrophically fails (0.5% correctly estimated relative poses on average). Matching raw part detections works slightly better (18.4%), since the dedicated detectors search for a pre-trained part irrespective of the viewpoint, rather than comparing low-level appearance patterns. The DPM-3D-Const. [45] already outperforms our previous result of 51.9% [66], but is in turn superseded by a significant margin of 22.3% by our *full system* (79.4%). Note that even for 180° viewpoint spacing, 76.5% of the estimated epipolar geometries are correct, see examples in Fig. 11(g).

4.6 Fine-grained categorization by 3D geometry

In addition to the popular task of distinguishing between basic-level categories (such as cat and dog), fine-grained categorization into sub-ordinate categories (such as sheep dog and Labrador) has received increasing attention in the vision literature lately [41], [63], [15]. It is deemed challenging due to the need to capture subtle appearance differences between classes (*e.g.*, fur texture) while at the same time maintaining robustness to intra-class variations induced by viewpoint changes and lighting conditions. As a consequence, the focus has mostly been on classes and categorization methods that favor discrimination by strong local cues (such as random image patches [63], [15]) or global image statistics (such as color and gradient histograms for flowers [41]).

In the following experiment, we choose a different

route, and base the fine-grained categorization entirely on 3D geometry. In particular, we consider the natural distinction between fine-grained sub-ordinate categories of cars and bicycles, such as sedans, sports cars, SUVs, etc. as well as mountain bikes, street bikes, etc.

We perform fine-grained categorization following a nearest neighbor scheme. Starting from a 3D wireframe estimate obtained by our model for a test image, we retrieve the closest wireframe exemplar from the database of CAD models of the basic-level object class of interest (car or bicycle), using Euclidean distance between translation- and scale-invariant wireframe representations. Examples of nearest neighbor matches are visualized in Fig. 11(a) - (f), which show edge renderings of retrieved CAD models, projected into the respective test image at the estimated location, scale, and viewpoint. Please note the remarkable accuracy of the fully automatic 3D geometry estimates.

Protocol. We suggest the following procedure to quantify the performance of fine-grained categorization based on the *3D Object Classes* data set: For each of the 5 car and 5 bicycle instances in the test set, we manually determine the single best matching CAD model in terms of 3D geometry, using the methodology described in Sect. 4.4. We then consider each of these CAD models a prototype of a fine-grained category, and measure how often the retrieved CAD models are sufficiently similar to these prototypes, by thresholding the mean Euclidean distance between corresponding vertices of the 3D fit and the annotated CAD model.

Results. Tab. 6 gives fine-grained categorization results for cars (a) and bicycles (b), comparing our *full system*, *GT*, and a *chance* baseline returning random CAD models from the database. The first five columns give the fraction of retrieved CAD models deemed sufficiently similar to the respective fine-grained category prototype. The last columns give the corresponding total fractions: for both cars (Tab. 6(a)) and bicycles (Tab. 6(b)), our *full system* successfully recovers the fine-grained category in two thirds of the cases (67.8% for cars, 66.1% for bicycles). Fig. 9 shows corresponding examples. The examples show how sedans are most frequently matched to sedans (Fig. 9(a)), racing bikes to racing bikes (Fig. 9(b), top), and mountain bikes to mountain bikes (Fig. 9(b), bottom).

5 Conclusions

We have designed a detailed 3D geometric object class model for 3D object recognition and modeling, complementing ideas from the early days of computer vision with modern techniques for robust model-to-image matching. Combining 3D wireframes with discriminative local shape detectors, we have demonstrated the successful recovery of detailed 3D object shape and pose from single input images. We believe that this high level of geometric detail in an important ingredient to advance scene-level reasoning beyond what can be achieved with box-level object class representations.

In an extensive experimental study on the object classes *car* and *bicycle*, we have quantified the ability of our proposed system to recover detailed geometric object hypotheses from single images. The model has been tested in four different settings, ranging from accurate 2D localization of object parts, through continuous pose estimation, to ultra-wide baseline matching and fine-grained categorization of car and bicycle types. Throughout, the performance is en par with or higher than previously reported results.

In the future, we plan to extend the present work in two directions, namely to explicitly handle occluded object parts, and to reason jointly over multiple instances of several object classes in the same scene, in order to exploit the additional constraints due to the common viewpoint as well as interactions between objects.

Acknowledgements We thank Bojan Pepik for providing his detections [45] for use as initialization. This work has been supported by the Max Planck Center for Visual Computing and Communication.

References

[1] S. Agarwal and D. Roth. Learning a sparse representation for object detection. In *ECCV*, 2002.
[2] M. Andriluka, S. Roth, and B. Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *CVPR*, 2009.
[3] M. Arie-Nachimson and R. Basri. Constructing implicit 3D shape models for pose estimation. In *ICCV*, 2009.
[4] S. Y. Bao and S. Savarese. Semantic structure from motion. In *CVPR*, 2011.
[5] O. Barinova, V. Lempitsky, E. Tretyak, and P. Kohli. Geometric image parsing in man-made environments. In *ECCV*, 2010.
[6] L. Bourdev and J. Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *ICCV*, 2009.
[7] L. Breiman. Random forests. *Machine Learning*, 2001.
[8] R. A. Brooks. Symbolic reasoning among 3-d models and 2-d images. *Artificial Intelligence*, 17(1-3):285–348, 1981.
[9] Y. Chen, T.-K. Kim, and R. Cipolla. Inferring 3d shapes and deformations from single views. In *ECCV*, 2010.
[10] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models, their training and application. *CVIU*, 1995.
[11] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *In Workshop on Statistical Learning in Computer Vision*, 2004.
[12] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.
[13] A. Ess, B. Leibe, K. Schindler, and L. V. Gool. Robust multi-person tracking from a mobile platform. *PAMI*, 2009.
[14] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2):303–338, 2010.
[15] 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.
[16] P. F. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *PAMI*, 32(9):1627–1645, 2010.
[17] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR*, 2003.

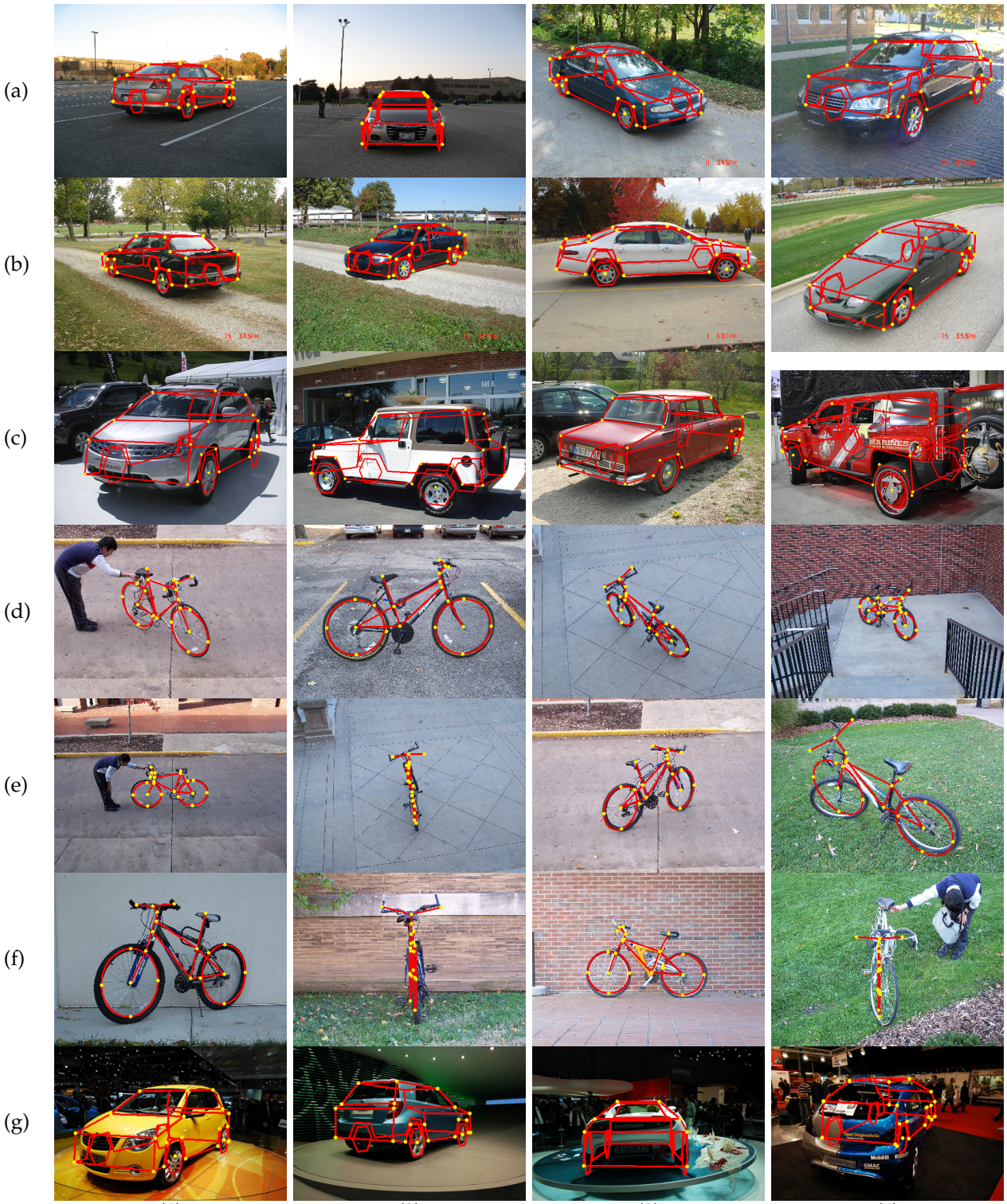


Fig. 10. Example detections using the *full system*: estimated wireframes (yellow dots mark visible parts). *3D Object Classes* cars (rows (a), (b)), *Pascal VOC 2006* cars (row (c)), *3D Object Classes* bicycles (rows (d) - (f)), *EPFL Multi-view* cars (rows (g)). Successful detections (columns (1) - (3)), typical failure cases (column (4)).

[18] Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[19] D. Glasner, M. Galun, S. Alpert, R. Basri, and G. Shakhnarovich. Viewpoint-aware object detection and pose estimation. In *ICCV*, 2011.

[20] C. Gu and X. Ren. Discriminative mixture-of-templates for viewpoint classification. In *ECCV*, 2010.

[21] A. Gupta, A. A. Efros, and M. Hebert. Blocks world revisited: Image understanding using qualitative geometry and mechanics. In *ECCV*, 2010.

[22] M. Haag and H.-H. Nagel. Combination of edge element and

Fig. 11. Fully automatic 3D geometry estimation from single still images of *3D Object classes* cars (rows (a) - (c)) and bicycles ((d) - (f)) using the *full system* (edges of nearest database CAD models rendered in red; ground plane inferred from wheel positions). Ultra-wide baseline matching from car image pairs (row (g)).

optical flow estimates for 3d-model-based vehicle tracking in traffic image sequences. *IJCV*, 35(3):295–319, 1999.

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

[24] V. Hedau, D. Hoiem, and D. Forsyth. Thinking inside the box: Using appearance models and context based on room geometry. In *ECCV*, 2010.

[25] M. Hejrati and D. Ramanan. Analyzing 3d objects in cluttered images. In *NIPS*, 2012.

[26] D. Hoiem, A. Efros, and M. Hebert. Putting objects in perspective. *IJCV*, 80(1):3–15, 2008.

[27] D. Koller, K. Daniilidis, and H. H. Nagel. Model-based object tracking in monocular image sequences of road traffic scenes. *IJCV*, 10(3):257–281, 1993.

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

[29] B. Leibe, A. Leonardis, and B. Schiele. An implicit shape model for combined object categorization and segmentation. In *Toward Category-Level Object Recognition*, 2006.

[30] C. Leistner. *Semi-Supervised Ensemble Methods for Computer Vision*. PhD thesis, TU Graz, 2010.

[31] M. Leordeanu and M. Hebert. Smoothing-based optimization. In *CVPR*, 2008.

[32] V. Lepetit and P. Fua. Keypoint recognition using randomized

trees. *PAMI*, 28(9):1465–1479, 2006.
[33] Y. Li, L. Gu, and T. Kanade. Robustly aligning a shape model and its application to car alignment of unknown pose. *PAMI*, 33(9):1860–1876, 2011.
[34] J. Liebelt and C. Schmid. Multi-view object class detection with a 3D geometric model. In *CVPR*, 2010.
[35] J. Liebelt, C. Schmid, and K. Schertler. Viewpoint-independent object class detection using 3D feature maps. In *CVPR*, 2008.
[36] R. J. Lopez-Sastre, T. Tuytelaars, and S. Savarese. Deformable part models revisited: A performance evaluation for object category pose estimation. In *ICCV-WS CORP*, 2011.
[37] D. Lowe. Distinctive image features from scale invariant keypoints. *IJCV*, 2(60):91–110, 2004.
[38] D. G. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 1987.
[39] D. Marr and H. K. Nishihara. Representation and recognition of the spatial organization of three-dimensional shapes. *Proc. Roy. Soc. London B*, 200(1140):269–194, 1978.
[40] B. H. Menze, B. M. Kelm, D. N. Splitthoff, U. Koethe, and F. A. Hamprecht. On oblique random forests. In *ECML/PKDD*, 2011.
[41] M. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *ICVGIP*, 2008.
[42] M. Ozuysal, V. Lepetit, and P. Fua. Pose estimation for category specific multiview object localization. In *CVPR*, 2009.
[43] N. Payet and S. Todorovic. From contours to 3d object detection and pose estimation. In *ICCV*, 2011.
[44] A. P. Pentland. Perceptual organization and the representation of natural form. *Artificial Intelligence*, 28(3):293–331, 1986.
[45] B. Pepik, M. Stark, P. Gehler, and B. Schiele. Teaching 3d geometry to deformable part models. In *CVPR*, 2012.
[46] L. G. Roberts. *'Machine Perception of Three-Dimensional Solides*. PhD thesis, MIT, 1963.
[47] R. Salakhutdinov, A. Torralba, and J. B. Tennenbaum. Learning to share visual appearance for multiclass object detection. In *CVPR*, 2011.
[48] S. Savarese and L. Fei-Fei. 3D generic object categorization, localization and pose estimation. In *ICCV*, 2007.
[49] H. Schneiderman and T. Kanade. A statistical method for 3d object detection applied to faces and cars. In *CVPR*, 2000.
[50] S. Shalom, L. Shapira, A. Shamir, and D. Cohen-Or. Part analogies in sets of objects. In *Eurographics Symposium on 3D Object Retrieval*, 2008.
[51] M. Stark, M. Gösele, and B. Schiele. Back to the future: Learning shape models from 3d cad data. In *BMVC*, 2010.
[52] H. Su, M. Sun, L. Fei-Fei, and S. Savarese. Learning a dense multi-view representation for detection, viewpoint classification and synthesis of object categories. In *ICCV*, 2009.
[53] G. D. Sullivan, A. D. Worrall, and J. Ferryman. Visual object recognition using deformable models of vehicles. In *IEEE Workshop on Context-Based Vision*, 1995.
[54] M. Sun, B. Xu, G. Bradski, and S. Savarese. Depth-encoded hough voting for joint object detection and shape recovery. In *ECCV*, 2010.
[55] A. Thomas, V. Ferrari, B. Leibe, T. Tuytelaars, B. Schiele, and L. Van Gool. Towards multi-view object class detection. In *CVPR*, 2006.
[56] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms, 2008.
[57] M. Villamizar, H. Grabner, J. Andrade-Cetto, A. Sanfeliu, L. V. Gool, and F. Moreno-Noguer. Efficient 3d object detection using multiple pose-specific classifiers. In *BMVC*, 2011.
[58] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR*, 2001.
[59] H. Wang, S. Gould, and D. Koller. Discriminative learning with latent variables for cluttered indoor scene understanding. In *ECCV*, 2010.
[60] C. Wojek, S. Roth, K. Schindler, and B. Schiele. Monocular 3d scene modeling and inference: Understanding multi-object traffic scenes. In *ECCV*, 2010.
[61] Y. Xiang and S. Savarese. Estimating the aspect layout of object categories. In *CVPR*, 2012.
[62] P. Yan, S. Khan, and M. Shah. 3D model based object class detection in an arbitrary view. In *ICCV*, 2007.
[63] B. Yao, A. Khosla, and L. Fei-Fei. Combining randomization and discrimination for fine-grained image categorization. In *ICCV*, 2011.
[64] L. L. Zhu, Y. Chen, A. Torralba, W. Freeman, and A. Yuille. Part and appearance sharing: Recursive compositional models for multi-view multi-object detection. In *CVPR*, 2010.
[65] M. Z. Zia, M. Stark, and K. Schindler. Explicit occlusion modeling for 3d object class representations. In *CVPR*, 2013.
[66] M. Z. Zia, M. Stark, K. Schindler, and B. Schiele. Revisiting 3d geometric models for accurate object shape and pose. In *ICCV-WS 3dRR*, 2011.

M. Zeeshan Zia received a Master's degree in Electrical Communication Engineering from TU Munich in 2009, where he worked on 3D object recognition, and first-person action recognition. He is currently a PhD candidate at ETH Zurich working on 3D object recognition applied to scene interpretation. He won a best paper award at 3dRR-11 (together with co-authors), a Qualcomm Innovation Fellowship 2012, and a best PhD student award at ICVSS 2012. He currently serves as associate editor (book reviews) for the IAPR Newsletter. His research interests are in object detection, scene-level understanding, and activity recognition.

Michael Stark received a Diploma degree in Computer Science from TU Darmstadt, Germany, in 2005, and a PhD from TU Darmstadt, Germany, in 2010. Since then, he has worked as a postdoc with the Max Planck Institute for Informatics in Saarbrücken, Germany, and Stanford University, Stanford, USA. He has been appointed as a Visiting Assistant Professor at Stanford University in 2012, and is currently heading the research group for Visual Object Recognition and Scene Interpretation of the Max Planck Center for Visual Computing and Communication. His research interests encompass computer vision and machine learning, focusing on rich and scalable image cues for scene understanding.

Bernt Schiele received his masters in computer science from Univ. of Karlsruhe and INP Grenoble in 1994. In 1997 he obtained his PhD from INP Grenoble in computer vision. He was a postdoctoral associate and Visiting Assistant Professor at MIT between 1997 and 2000. From 1999 until 2004 he was an Assistant Professor at ETH Zurich and from 2004 to 2010 he was a full professor of computer science at TU Darmstadt. In 2010, he was appointed scientific member of the Max Planck Society and a director at the Max Planck Institute for Informatics. Since 2010 he has also been a Professor at Saarland University. His main interests are computer vision, perceptual computing, statistical learning methods, wearable computers, and integration of multi-modal sensor data. He is particularly interested in developing methods which work under real-world conditions.

Konrad Schindler received a Diplomingenieur (M.tech) degree in photogrammetry from Vienna University of Technology, Austria in 1999, and a PhD from Graz University of Technology, Austria, in 2003. He has worked as a photogrammetric engineer in the private industry, and held researcher positions in the Computer Graphics and Vision Department of Graz University of Technology, the Digital Perception Lab of Monash University, and the Computer Vision Lab of ETH Zurich. He became assistant professor of Image Understanding at TU Darmstadt in 2009, and since 2010 has been a tenured professor of Photogrammetry and Remote Sensing at ETH Zurich. His research interests lie in the field of computer vision, photogrammetry, and remote sensing, with a focus on image understanding and 3d reconstruction. He currently serves as head of the Institute of Geodesy and Photogrammetry, and as associate editor for the ISPRS Journal of Photogrammetry and Remote Sensing, and for the Image and Vision Computing Journal.