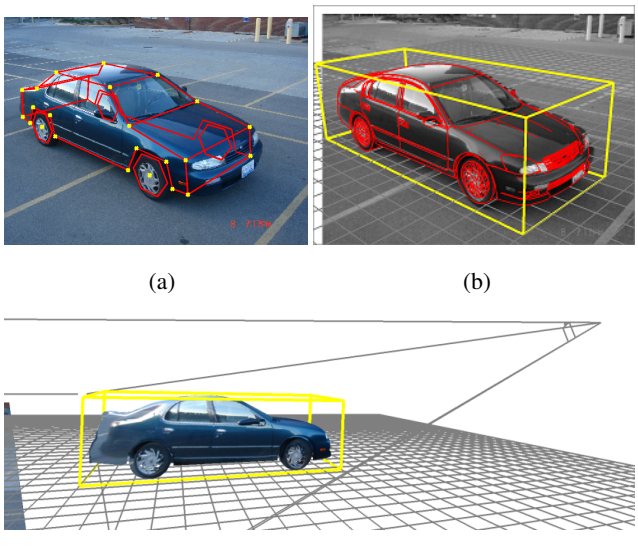
Revisiting 3D Geometric Models for Accurate Object Shape and Pose

M. Zeeshan Zia[1], Michael Stark[2], Bernt Schiele[2], and Konrad Schindler[1]

[1] Photogrammetry and Remote Sensing Laboratory, ETH Zürich, Switzerland
[2] Max-Planck-Institute for Informatics, Saarbrücken, Germany
{mzia, konrads}@ethz.ch, {stark, schiele}@mpi-inf.mpg.de

Abstract

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



(a) (b)

(c)

Figure 1. Fully automatic shape and pose estimation results: (a) estimated 3D wireframe, (b) overlaid closest training 3D CAD model, (c) reconstruction of object shape, pose, and camera pose (CAD model rendered from novel viewpoint using original image as texture; pyramid denotes camera field-of-view).

1. Introduction

In the early days of computer vision, 3-dimensional geometric representations of object shape were considered the holy grail for both the recognition of individual objects and the interpretation of entire visual scenes [28, 5, 31, 26, 22, 36, 17]. These approaches typically provided rich descriptions of constituent scene entities, but proved difficult to match robustly to real-world imagery. As a consequence, subsequent approaches often traded geometric accuracy for robustness, achieved by combining local features with statistical learning techniques. While this has lead to impressive performance for the recognition of a variety of object classes [10] as well as for scene classification and region labeling, the degree to which interactions between scene entities can be modeled and leveraged is typically limited to co-occurrence statistics and 2D spatial relations.

More recently, researchers have revived coarse 3D geometric representations in the context of indoor [40, 19] and outdoor [20, 4, 15] scene understanding, and demonstrated the benefit of 3D geometric reasoning both in terms of increased expressiveness of the models and increased 2D recognition performance. Similarly, scene-level geometric reasoning has been shown to improve performance for recognizing and tracking pedestrians and vehicles from mobile platforms [9, 41]. These approaches, however, either use purely qualitative geometry descriptions [20, 15] or resort to geometric representations of rather coarse granularity [9, 41], where reasoning is performed at most on the level of entire objects: current off-the-shelf object class detectors [8, 11] typically deliver object hypotheses in the

form of 2D bounding boxes, which convey only little geometric information and limit the scope of high-level reasoning.

We believe that more fine-grained geometric detail on the level of (putative) objects constitutes a key aspect for more accurate 3D scene understanding. In the present paper, we therefore go back to the early days of computer vision, and revisit 3D geometric representations for object class modeling, in the spirit of [22, 36, 17]. These representations characterize an object class by a combination of a global 3D wireframe with local edges, and explicitly project the 3D model into the images to gather evidence for recognition. In contrast to those approaches, we base our implementation on modern local shape features, discriminative part detectors, and efficient techniques of approximate probabilistic inference. As we will show, these more recent innovations make the "old-fashioned" 3D shape models applicable to challenging real-world imagery.

The paper makes the following contributions. First, we propose to revisit 3D geometric object class representations, providing object hypotheses with much more geometric detail than current object class detectors (see Fig. 1). We consider this geometric richness a vital ingredient for accurate scene-level geometric reasoning. Second, we demonstrate the ability of our model to 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 data set, at the same time improving over previously published results w.r.t. robust 2D localization. Third, we show the benefit of detailed geometric category models for ultra-wide baseline matching, where we successfully recover relative camera pose over viewpoint changes up to 180°. And fourth, we give first experimental results on predicting fine-grained object categories (individual car models) based on inferred 3D geometry.

2. Related work

The 3-dimensional nature of objects has lately received renewed attention in the context of viewpoint-independent object class recognition. The multi-view aspect is usually reflected by a class model made up of several flat, viewpoint-dependent representations [30] (corresponding to the viewpoints used for training). Additionally, the relations among those viewpoints are modeled in order to also cover previously unobserved ones. Prominent approaches range from establishing connections between corresponding codebook entries by feature tracking [38] and homography estimation [42, 2] to probabilistic viewpoint morphing between object parts [35] and training discriminative mixtures of viewpoint templates [14].

More recently, attempts have been made to explicitly represent 3D geometry alongside object appearance. Liebelt and Schmid [25] couple a coarse volumetric blob model, which they learn from 3D data, with 2D parts laid out on a regular grid. Stark et al. [34] train banks of constellation models [12] from rendered 3D CAD data, using semantic part correspondences. Sun et al. [37] enrich the implicit shape model [23] by including the relative depth between codebook entries obtained from a structured light system.

While these richer object class models constitute promising steps towards embedding 3D geometry, and have lead to remarkable recognition performance under large viewpoint variations [34, 37], they still capture – and output – only a limited degree of 3D geometric information: effectively they perform 2D localization and coarse classification into discrete viewpoints. In contrast, we seek accurate continuous estimates of object pose and shape, much like early approaches like [22, 36, 17]. Notably, these estimates comprise the relative 3D positions of potentially occluded object parts, and hence, the 3D extent of objects, lending itself to volumetric [15] or functional [16] reasoning.

To that end, our approach leverages ideas from active shape (ASMs) and active appearance models (AAMs) usually applied in a 2D setting [7, 24]. Our model is based on a true 3D wireframe representation that is projected into image space only at recognition time. In this respect, our approach also provides a more powerful geometric representation than the recently proposed method by Sun et al. [37], which hypothesizes relative depths of individual feature matches, conditioned on the object center. In contrast to recent work on 3D shape recovery involving an interactive segmentation step [6] our approach is directly applicable to challenging real-world images.

3. 3D Geometric object class model

In order to capture geometric detail, we represent an object class as a combination of a coarse 3D wireframe, representing global object geometry, with attached local shape representations of object parts [22, 36, 17] (see Fig. 1). This is in line with factoring object representations into separate components for global layout and local appearance, as done in many modern recognition systems [12, 11]. At the same time, we draw from recent results [34] and base the object class model *entirely on 3D computer aided design (CAD) models* rather than real-world training images. In particular, we use a collection of 3D CAD models of the object class of interest to learn both the coarse global wireframe and local part shape models. We thus leverage the geometric accuracy of 3D CAD models, while ensuring consistency between global wireframe and local part shape models by design.

In contrast to [34], we propose a true 3D geometric object class representation learned from these models rather than a bank of viewpoint-dependent 2D detectors. The resulting recognition hypotheses are thus qualitatively dif-

ferent from those in [34]: we predict relative 3D positions of object parts, not merely 2D bounding boxes and coarse viewpoint labels. Our geometric representation also ensures that the placement of individual parts always results in a plausible global object shape. This is contrary to [34], where false image evidence for an individual part can wrongfully outweigh an implausible constellation.

3.1. Global geometry representation and learning

Our global geometry representation is given by a coarse, 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 m 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 and its vertices are chosen manually on the 3D CAD models, but they could potentially be obtained using part-aware mesh segmentation techniques from the computer graphics literature [33]. We follow the classical formulation of point-based shape analysis [7], 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: $\mathbf{X}(\mathbf{s}) = \mu + \sum_{k=1}^{r} s_k \sigma_k \mathbf{p}_k + \epsilon$.

3.2. Local shape representation and learning

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 [22, 36, 17], which has proven to be of limited robustness in the face of real-world image noise and clutter. Following [34], we employ sliding-window detectors, using a dense variant of shape context as features [1] and AdaBoost as classifiers. For each wireframe vertex, a detector is trained from vertex-centered patches of non-photorealistic renderings of our 3D CAD models. This combination has shown to yield a robust transition from rendered to real world images. Since positives are rendered from wireframes, it allows one to generate massive amounts of artificial training data from arbitrary viewpoints.

3.3. Viewpoint-invariant shape and 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 maximum a posteriori (MAP) hypothesis search over possible projections of the model into the test image. We point out that this means searching over continuous 3D geometry and viewpoint parameters rather than interpolating between flat viewpoint-dependent representations as in previous work [38, 35].

More formally, we denote the image evidence as $\mathcal{E}$, and the recognition hypothesis $\mathbf{h} = (\mathbf{s}, f, \boldsymbol{\theta}, \mathbf{q})$. 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 $\mathsf{R}(\boldsymbol{\theta})$, i.e., $\mathsf{P}(f, \boldsymbol{\theta}, \mathbf{q}) = \mathsf{K}(f) \begin{bmatrix} \mathsf{R}(\boldsymbol{\theta}) & -\mathsf{R}(\boldsymbol{\theta})\mathbf{q} \end{bmatrix}$. It projects wireframe vertices $\mathbf{X}_j(\mathbf{s})$ to image coordinates $\mathbf{x}_j = \mathsf{P}\mathbf{X}_j(\mathbf{s})$. For recognition, we want to find $\mathbf{h}_{map} = argmax_{\mathbf{h}} P(\mathbf{h}|\mathcal{E}) = argmax_{\mathbf{h}} P(\mathcal{E}|\mathbf{h})P(\mathbf{h})$, assuming a uniform prior over $\mathcal{E}$. $P(\mathbf{h})$ is set to a uniform distribution over the PCA space governing $\mathbf{s}$, as well as over f, $\boldsymbol{\theta}$, and $\mathbf{q}$.

Likelihood. We define the likelihood of an instance of our model being present in the image as the combination of the likelihoods of its constituent parts (see Sect. 3.2). For a part j, we obtain its likelihood $S_j(\boldsymbol{\varsigma}, \boldsymbol{\theta}, \mathbf{x}_j)$ by looking up the detection score at image location $\mathbf{x}_j$ and local scale $\boldsymbol{\varsigma}$, observed from viewpoint $\boldsymbol{\theta}$. We turn detection scores into probabilities using Platt scaling [29]. In order to account for object-level self-occlusion, we only consider parts that are visible in the projected model, which we represent by indicator functions $o_j(\mathbf{s}, \boldsymbol{\theta})$. Assuming conditional independence between parts, we obtain the following (pseudo-)likelihood, normalized over visible parts

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

Inference. We approximate $\mathbf{h}_{map}$ by drawing samples from the distribution $P(\mathcal{E}|\mathbf{h})P(\mathbf{h})$, using a particle filter. Similar to the condensation algorithm [21], we maintain a set of weighted samples (particles), each corresponding to a distinct set of values assigned to the constituent parameters of the hypothesis space, $\mathbf{s}$, $\boldsymbol{\theta}$, and $\mathbf{q}$. f is held fixed in our experiments, assuming all images have been taken by the same camera. The particles are updated in an iterative procedure, by re-sampling individual parameters from independent Gaussians centered at their former values. The variances of these Gaussians are successively reduced, according to an annealing schedule, for faster convergence.

Initialization. Rather than running inference blindly over entire test images, we start from promising image positions and scales, which we obtain in the form of predicted object bounding boxes from our 2D multi-view detector [34]. 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 estimate predicted by [34]. We also initialize from the corresponding opposing viewpoint (180° apart in azimuth), since opposing views are often confused by the detector.experiments, we set the

4. Experimental evaluation

In the following, we give a series of experiments, in which we evaluate the ability of our 3D object class model to accurately capture object shape and pose. We base our evaluation on the 3D Object Classes data set [32], since it has been designed for multi-view recognition and constitutes a suitable trade-off between controlled conditions for experimentation and challenging real-world imagery. We focus on the car class, shown at 8 different azimuth angles, 2 elevation angles, and 3 distances, against varying backgrounds (see Fig. 2(c)). We report the performance of our *full system*, consisting of running a multi-view detector [34] and then applying our model on top of the resulting hypotheses (Sect. 3). In order not to depend on the particular initializations provided by [34], we also give results for initializing our model with the *ground truth bounding boxes* (*GT*) and corresponding coarse azimuth angle estimates defined by the data set.

Training. In all experiments, we train our object class model from a collection of 38 commercially available 3D CAD models of different car models (www.doschdesign.com). From each of the CAD models, we select 36 vertices to yield wireframe exemplars for training our 3D geometry representation (see Sect. 3.1; due to the symmetry of cars, only 20 vertices per CAD model have to be annotated). We further train separate local part shape detectors (see Sect. 3.2) for each of the vertices, from 72 different azimuth angles (5° steps) and 2 elevation angles (7.5° and 15° above the ground), densely covering the relevant part of the viewing sphere. CAD wireframe models are rendered (without texture) into non-photorealistic images in the same way as already done in [34]. Detectors are trained using part renderings as positive examples, and random crops from a background image set as negatives. Positive examples are randomly jittered in order to improve robustness [1].

Inference. We sample θ_{az} and θ_{el} over continuous ranges of 70° and 20°, respectively, centered around the initialization. For part detections, we consider the maximum score in a scale range of $\pm 30\%$ of the bounding box scale.

4.1. Multi-view recognition

As a sanity check, we evaluate the performance of our model in a classical multi-view recognition setting, measuring its ability to localize objects in 2D. For that purpose, we use our model to rescore detection hypotheses from the detector of [34], over the same test set as used in their experiments (240 test images, 5 cars in total). In particular, for each hypothesis, we form a linear combination of the two detections scores, normalized to comparable value ranges. Fig. 2 (b) gives the corresponding precision-recall plots, using the PASCAL criterion [10] ($\frac{intersection}{union}$ >50%), and previous results from a comparable setting [35, 13, 25, 34]. The combined score (red curve) reaches the – to our knowledge – best published result. While being only marginally better than the original detector of [34] (green curve) in terms of average precision (90.4% AP vs. 89.8% AP), it consistently delivers higher recall. This holds true in particular for the region of high precision (> 90%), in line with the intuition that a detailed geometric model should facilitate precise localization and recognition.

Part localization. Aiming at accurate geometric reasoning on the scene level requires object class models to provide well-defined anchor points by which they can be geometrically related to other scene entities. Consider, e.g., the wheels of a car, which usually touch the ground, and can hence provide constraints for the geometry of the supporting plane. In our model, the parts defined by vertices of the coarse wireframe serve that purpose, in contrast to recent successful detectors deciding on object parts purely according to discriminative power [11] rather than geometric relation. Since the geometric parts have physical manifestations in the real world, we can also evaluate the accuracy of our model to localize these parts individually. For that purpose, we annotate the 2D locations of all visible parts[1].

Fig. 2 (a)(left) depicts the localization performance for all 36 parts, averaged over all test images, distinguishing between the *full system* (blue bars), and *GT* (red bars), considering true positive detections. 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 a car side view at scale 1.0, covering 460×155 pixels, that number is 20, which amounts to localizing a part accurately to $\approx 4\%$ of the car length. Note that this strict criterion is applied in all cases, even for hypothesws with grossly wrong viewpoint estimates.

In Fig. 2 (a)(left), we observe that there are in fact differences in the localization accuracy of different parts. Notably, parts located in the wheel regions (9-18, 27-36) are localized almost perfectly when starting from *GT*, and still with at least 89.3% by the *full system*. This is not surprising, since wheels provide plenty of local structure, and hence can be robustly found by local part detectors. Parts on the front portion of the car (1-4, 19-22; between 60.3% and 76.6% for the *full system*) tend to perform better than the corresponding ones on the back (5-8, 23-26; between 50.6% and 62.3%). We attribute this difference to the greater flexibility that our learned global geometry model allows

[1] Annotations: http://www.igp.ethz.ch/photogrammetry/research/index

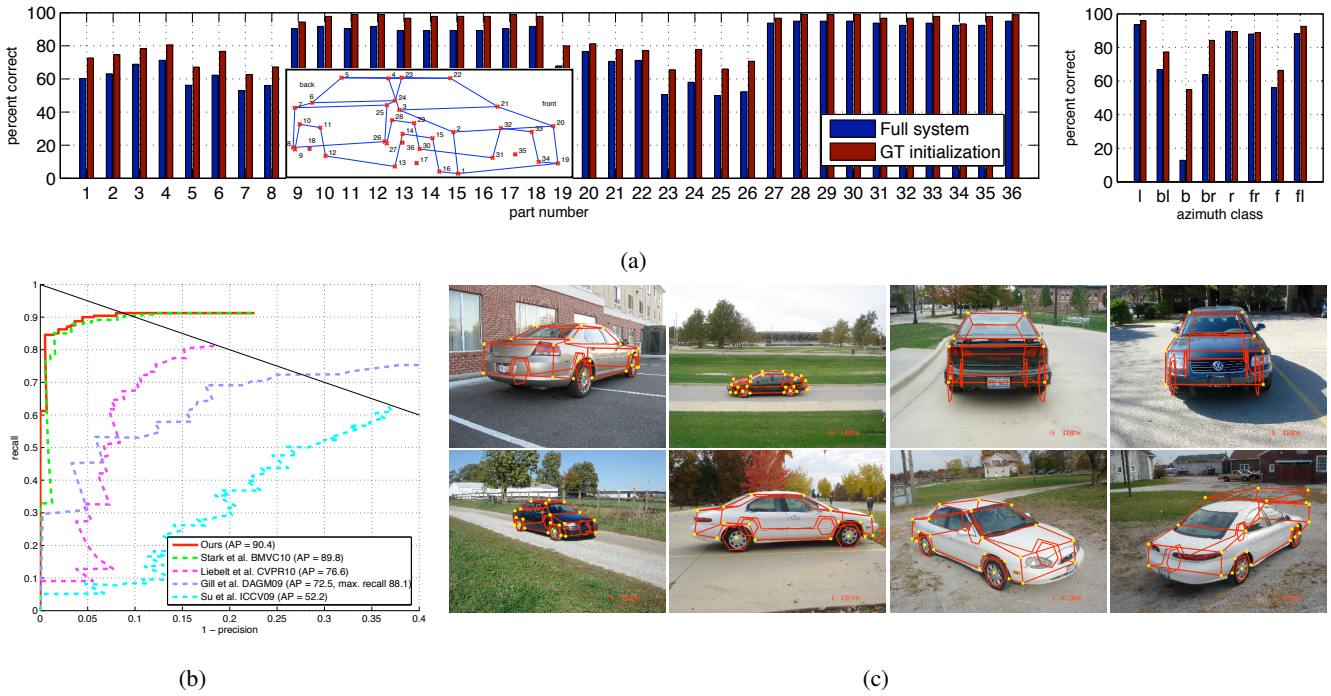


Figure 2. Results: (a) part localization for individual parts (left) and viewpoints (right). (b) Multiview recognition in comparison to related work [35, 13, 25, 34], (c) example detections in the form of estimated wireframes (yellow crosses mark visible parts).

in the back: the collection of 3D CAD models comprises limousines and sports cars as well as SUVs and station wagons (Fig. 2(c)(bottom, right) shows a limousine wrongly considered a station wagon).

Fig. 2 (a)(right) groups the part localization results according to the different azimuth angles of test images, averaged over all parts. We observe that part localization performs best for plain side views (left 93.5%, right 89.7%, *full system*), followed by diagonal front (front-left 88.2%, front-right 88.0%) and back views (back-left 66.9%, back-right 63.8%). Plain front views perform moderately (56.1%), while back views essentially fail (12.8%). We attribute the generally better performance of diagonal views to the added localization ability of the wheel parts, while the roundish shapes of roofs, windshields, hoods, and bumpers provide much less evidence about exact part positions in plain front views, in particular for close-ups. Plain back views are often mistakenly estimated as plain front views (already by the initial detections from [34]), leading to large pixel distances between estimated and annotated parts, which explains the poor performance. On average we achieve correct localization in 74.2% of all cases (*full system*).

Summary. We conclude that our model in many cases yields accurate 2D localization on both the level of entire objects and individual parts, providing a solid basis for accurate 3D geometric reasoning on the scene-level. We also observe a non-negligible difference between the results obtained by different initializations (*full system* vs. *GT*), and thus expect further improvements in response to improved detections to start from.

4.2. Pose estimation

In this section, we evaluate the ability of our model to accurately estimate the pose of recognized objects, in the form of azimuth and elevation angles. We include the corresponding results of the detector of [34] as a reference.

Coarse viewpoint classification. Following the experimental protocol of [35], we report results on the classification of true positive detections according to 8 azimuth angle classes. In comparison to the previously reported best result of (average accuracy 81%, [34]), our model achieves a noticeable improvement (average accuracy 84%, initialized from [34] and using the combined score).

Continuous viewpoint estimation. Since the ground truth poses provided by the *3D Object Classes* data set are limited to a coarse separation into discrete viewpoint classes, we annotate all 48 images depicting one particular car 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 plausible scene geometries for all images [1]. Tab. 1 (a) gives the results for continuous viewpoint estimation, comparing the *full system*, *GT*, and [34],

	Total Images	True Positives	% Correct Azimuth	Avg. Error Azimuth	Avg. Error Elevation
[34]	48	46	67.4%	4.2°	4.0°
Full System	48	45	73.3%	3.8°	3.6°
GT	48	48	89.6%	4.2°	3.6°

(a)

Azimuth Diff.	No. of Image Pairs	*GT*	*Full System*	SIFT	Parts only
45°	53	91%	55%	2%	27%
90°	35	91%	60%	0%	27%
135°	29	69%	52%	0%	10%
180°	17	59%	41%	0%	24%

(b)

Table 1. Results for (a) continuous viewpoint estimation, (b) ultra-wide baseline matching.

over all true positive detections. An azimuth angle is considered correct if it lies within 10° of the annotated ground truth. In Tab. 1 (a), we observe that our full system improves by 6% over [34] in the amount of correct azimuth estimates (73.3% vs. 67.4%). Among those, the average error decreases further from 4.15° to 3.82°. Similarly, the error in estimated elevation angle decreases from 4.04° to 3.56° (technically, [34] does not estimate elevation; we thus use the fixed value of 7.5° used during training as an estimate).

4.3. Fine-grained categorization from 3D geometry

Besides providing an accurate estimate of an object's pose, our model also outputs an explicit representation of its 3D geometry, independent of the image projection. In the case of the car object class, 3D geometry is naturally connected to a more fine-grained categorization into limousines, sports cars, SUVs, etc., which we hope to recover by using our model. In a first experiment, we thus use the wireframes estimated from test images to retrieve the closest wireframe exemplars from the CAD model database, using Euclidean distance between translation- and scale-invariant wireframe representations. Examples are depicted in Fig. 3 (rows 1, 2, 3). Shown are silhouette 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.

Since the *3D Object Classes* car data set exclusively depicts limousines, we suggest the following procedure to quantify the performance of fine-grained categorization. For each of the cars in the test set, we manually determine the single best matching CAD model w.r.t. 3D geometry, using the methodology described in Sect. 4.2. We then consider each of these CAD models a prototype of a fine-grained category. We then 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. In an encouraging 65.8% of the cases, our *full system* successfully recovers the fine-grained category of the car in the test image (68.3% for *GT*).

4.4. Ultra-wide baseline matching

Matching between different views of the same scene is a fundamental problem of computer vision, which quickly gets very challenging as the baseline increases; the best invariant interest point descriptors like SIFT [27] allow matching up to baselines of ≈ 30 degrees in orientation and a factor of ≈ 2 in scale. Only recently, Bao and Savarese [3] 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. We compare to two baseline methods: (1) we find putative matches with SIFT (using the default options in [39]); and (2) 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.

As test data we have extracted 134 pairs of images from the dataset, 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: straightforward 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

Figure 3. Results for fully automatic 3D geometry estimation (rows 1, 2, 3; ground-plane automatically inferred from wheel positions) and ultra-wide baseline matching (rows 4, 5; equal colors denote corresponding epipolar lines per image pair).

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, covering the car as well as the background. A fit is deemed correct if the Sampson error [18] for these points is < 20 pixels. The results are tabulated in Tab. 1 (b) according to (angular) baseline. As expected, SIFT catastrophically fails. Matching raw part detections works slightly better at 22% correct relative poses, since dedicated detectors are trained for each viewpoint. In contrast, our model reconstructs 52% of the pairs correctly when initialized with automatically detected bounding boxes. Starting from the correct bounding box (i.e., if a better initialization were available) the matching is $> 75\%$ correct overall, and $> 90\%$ correct up to baselines of 90°. Note that even for 180° viewpoint spacing more than half of the estimated epipolar geometries are correct (rows 4 and 5 of Fig. 3 give examples).

5. Conclusions

We have revisited 3D geometric object class models as a potential basis for scene understanding. To that end, we have demonstrated their ability to accurately localize objects and their geometric parts in 2D, improving over state-of-the-art results in 3D pose estimation on a standard benchmark data set for multi-view recognition, as well as to robustly localize objects in 2D, again improving over previous results. Having further shown that 3D geometric object class models enable novel applications, such as fine-grained

geometry-based object categorization and ultra-wide baseline matching from object parts, we believe they bear great potential for geometric reasoning at scene level.

References

[1] M. Andriluka, S. Roth, and B. Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *CVPR*, 2009. 3, 4

[2] M. Arie-Nachimson and R. Basri. Constructing implicit 3D shape models for pose estimation. In *ICCV*, 2009. 2

[3] S. Y. Bao and S. Savarese. Semantic structure from motion. In *CVPR*, 2011. 6

[4] O. Barinova, V. Lempitsky, E. Tretyak, and P. Kohli. Geometric image parsing in man-made environments. In *ECCV'10*. 1

[5] R. Brooks. Symbolic reasoning among 3-d models and 2-d images. *Artificial Intelligence*, 17:285–348, 1981. 1

[6] Y. Chen, T.-K. Kim, and R. Cipolla. Inferring 3d shapes and deformations from single views. In *ECCV*, 2010. 2

[7] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models, their training and application. *CVIU*, 61:38–59, 1995. 2, 3

[8] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005. 1

[9] A. Ess, B. Leibe, K. Schindler, and L. V. Gool. Robust multi-person tracking from a mobile platform. *PAMI*, 2009. 1

[10] M. Everingham, L. Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2):303–338, 2010. 1, 4

[11] P. F. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *PAMI*, 2009. 1, 2, 4

[12] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR'03*. 2

[13] G. Gill and M. Levine. Multi-view object detection based on spatial consistency in a low dimensional space. In *DAGM*, 2009. 4, 5

[14] C. Gu and X. Ren. Discriminative mixture-of-templates for viewpoint classification. In *ECCV*, 2010. 2

[15] A. Gupta, A. A. Efros, and M. Hebert. Blocks world revisited: Image understanding using qualitative geometry and mechanics. In *ECCV*, 2010. 1, 2

[16] A. Gupta, S. Satkin, A. A. Efros, and M. Hebert. From 3d scene geometry to human workspace. In *CVPR*, 2011. 2

[17] M. Haag and H.-H. Nagel. Combination of edge element and optical flow estimates for 3d-model-based vehicle tracking in traffic image sequences. *IJCV*, 1999. 1, 2, 3

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

[19] V. Hedau, D. Hoiem, and D. Forsyth. Thinking inside the box: Using appearance models and context based on room geometry. In *ECCV*, 2010. 1

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

[21] M. Isard and A. Blake. CONDENSATION – conditional density propagation for visual tracking. *IJCV'98*. 3

[22] D. Koller, K. Daniilidis, and H. H. Nagel. Model-based object tracking in monocular image sequences of road traffic scenes. *IJCV*, 1993. 1, 2, 3

[23] B. Leibe, A. Leonardis, and B. Schiele. An implicit shape model for combined object categorization and segmentation. In *Toward Category-Level Object Recognition*, 2006. 2

[24] Y. Li, L. Gu, and T. Kanade. A robust shape model for multi-view car alignment. In *CVPR*, 2009. 2

[25] J. Liebelt and C. Schmid. Multi-view object class detection with a 3D geometric model. In *CVPR*, 2010. 2, 4, 5

[26] D. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 1987. 1

[27] D. Lowe. Distinctive image features from scale invariant keypoints. *IJCV*, 2(60):91–110, 2004. 6

[28] D. Marr and H. Nishihara. Representation and recognition of the spatial organization of three-dimensional shapes. *Proc. Roy. Soc. London B*, 200(1140):269–194, 1978. 1

[29] A. Niculescu-Mizil and R. Caruana. Obtaining calibrated probabilities from boosting. In *UAI*, 2005. 3

[30] M. Ozuysal, V. Lepetit, and P. Fua. Pose estimation for category specific multiview object localization. In *CVPR'09*. 2

[31] A. Pentland. Perceptual organization and the representation of natural form. *Artificial Intelligence*, 28:293–331, 1986. 1

[32] S. Savarese and L. Fei-Fei. 3D generic object categorization, localization and pose estimation. In *ICCV*, 2007. 4

[33] S. Shalom, L. Shapira, A. Shamir, and D. Cohen-Or. Part analogies in sets of objects. In *Eurographics Symposium on 3D Object Retrieval*, 2008. 3

[34] M. Stark, M. Goesele, and B. Schiele. Back to the future: Learning shape models from 3d cad data. In *BMVC'10*. 2, 3, 4, 5, 6

[35] 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'09*. 2, 3, 4, 5

[36] 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. 1, 2, 3

[37] M. Sun, B. Xu, G. Bradski, and S. Savarese. Depth-encoded hough voting for joint object detection and shape recovery. In *ECCV*, 2010. 2

[38] A. Thomas, V. Ferrari, B. Leibe, T. Tuytelaars, B. Schiele, and L. Van Gool. Towards multi-view object class detection. In *CVPR*, 2006. 2, 3

[39] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms, 2008. 6

[40] H. Wang, S. Gould, and D. Koller. Discriminative learning with latent variables for cluttered indoor scene understanding. In *ECCV*, 2010. 1

[41] C. Wojek, S. Roth, K. Schindler, and B. Schiele. Monocular 3d scene modeling and inference: Understanding multi-object traffic scenes. In *ECCV*, 2010. 1

[42] P. Yan, S. Khan, and M. Shah. 3D model based object class detection in an arbitrary view. In *ICCV*, 2007. 2