Class-Specific Object Pose Estimation and Reconstruction Using 3D Part Geometry

Arun C.S. Kumar[1](✉), András Bódis-Szomorú[2], Suchendra Bhandarkar[1], and Mukta Prasad[3]

[1] University of Georgia, Athens, USA
aruncs@uga.edu, suchi@cs.uga.edu
[2] ETH Zürich, Zürich, Switzerland
bodis@vision.ee.ethz.ch
[3] Trinity College Dublin, Dublin, Ireland
prasadm@tcd.ie

Abstract. We propose a novel approach for detecting and reconstructing class-specific objects from 2D images. Reconstruction and detection, despite major advances, are still wanting in performance. Hence, approaches that try to solve them jointly, so that one can be used to resolve the ambiguities of the other, especially while employing data-driven class-specific learning, are increasingly popular. In this paper, we learn a deformable, fine-grained, part-based model from real world, class-specific, image sequences, so that given a new image, we can simultaneously estimate the 3D shape, viewpoint and the subsequent 2D detection results. This is a step beyond existing approaches, which are usually limited to 3D CAD shapes, regression based pose estimation, template based deformation modelling etc. We employ Structure from Motion (SfM) and part based models in our learning process, and estimate a 3D deformable object instance and a projection matrix that explains the image information. We demonstrate our approach with high quality qualitative and quantitative results on our real world RealCar dataset, as well as the EPFL car dataset.

1 Introduction

Despite big advances, core computer vision problems in the area of detection and reconstruction are far from perfectly solved. It is increasingly recognized that to combat the problems faced by these areas of vision, effective solutions must tackle them jointly, modelling the physics of image formation, learn from data, expert-knowledge and allow one problem to handle the ambiguities of the other. Although 3D geometric reasoning has become increasingly common in several computer vision applications, it is still some way off from becoming a standard consumer-level technique.

In this paper, we propose a framework that, given a 2D image, simultaneously detects an object category instance, estimates the object pose and shape in 3D, reasons about its part appearance and occlusion, thus performing object

G. Hua and H. Jégou (Eds.): ECCV 2016 Workshops, Part III, LNCS 9915, pp. 280–295, 2016.
DOI: 10.1007/978-3-319-49409-8_23

reconstruction in 3D and detection in 2D, jointly. The problem is ambitious because the proposed framework learns a class-specific, deformable fine-grained, part-based model from image sequences, learning both appearance and geometry. Note that the ill-posed nature of the problem results in a complex solution landscape with several local minima. In order to enable reasonable solutions, we solve the problem by tackling the complexity in a gradual, incremental way. We start from a constrained setup for which the solution can be found reliably and then gradually increase the flexibility in the model to handle more variables in the problem.

The idea of tackling vision problems jointly, has been gaining traction recently [10,12,13,15,23,24]. But the modern approaches, while making strides in tackling this problem, have often resorted to using high quality CAD models (which are expensive, painstaking to design, and/or limited in their capability to capture the object shape, appearance, especially the surface texture). Another tendency, is to model camera viewpoint using regression rather than modelling the physical projection process. Also, shape variation has been often been modelled using a bank of representations/templates. In our proposed approach, we learn SfM based class-specific shape and appearance models from real image sequences as faithfully as possible. Some supervised input is acquired through minimal, intuitive input for fine-grained part understanding. At test time, we formulate the detection and reconstruction problem in terms of the actual reprojection error (this models the scene physics more accurate than regression) and use a variety of RANSAC-based techniques in order to make estimation efficient and effective. We will expand on the related work in the next section.

2 Related Work

As mentioned above, the problem of joint detection, reconstruction and pose estimation of object classes from images has received considerable attention within the computer vision research community in recent years [10,12,24]. Existing approaches to solve this problem can be broadly categorized into two main subclasses, i.e., distinctive view-based techniques and 3D geometry-based techniques. Distinctive view-based techniques exploit robust but less descriptive 2D features for view-specific models for detection and recognition [2,3,5]. The proven usefulness and robustness of statistical 2D feature based methods from the computer vision research literature inspired the development of most distinctive view-based techniques. Existing techniques [4,6,8] treat viewpoint estimation as a classification problem by dividing the viewpoint range into discrete bins. Ghodrati *et al.* [6] train multiple Support Vector Machine (SVM) classifiers, one for each discrete viewpoint, treating each classifier independently of the others. He *et al.* [8] use a two-step process, wherein a viewpoint-parametrized classifier is first used to estimate a coarse viewpoint followed by fine-tuning step. Fenzi *et al.* [4] treat continuous viewpoint estimation as a regression problem which is solved using a Radial Basis Function Neural Network (RBF-NN). The RBF-NN is trained to predict the appearance features as a function of the viewpoint. Tulsiani *et al.* [15] train a Convolutional Neural Network (CNN) that can jointly predict the viewpoints for all classes using a shared feature representation. The CNN is used to estimate the coarse viewpoint which is subsequently leveraged for keypoint prediction. Though these view-based methods have been effective, one would expect that accurately modelling the physical projection process would be beneficial.

In recent years, due to the wide availability of affordable depth sensors, 3D shape repositories and 3D CAD models, coupled with the fact that it makes more sense to reason in terms of the underlying 3D structure of the object, the research focus has shifted towards 3D geometry-based techniques for solving the 3D object pose estimation and reconstruction problem. With improved optimization techniques and processing power, we are able to learn these, more powerful models. Pepik *et al.* [12,13] extended the Deformable Parts Model (DPM) [3] to represent the part locations and deformations in 3D. Yu *et al.* [17] on the other hand, propose an approach for learning a shape appearance and pose (SAP) model for both 2D and 3D cases, where the training instances with unknown pose are used to learn a probabilistic object part-based model. The class label and the pose of the object are inferred simultaneously by joint discovery of parts and alignment to a canonical pose. Xiao *et al.* [16] and Kim *et al.* [11], exploit synthetic 3D models to incorporate 3D geometric information into the DPM framework [3] for pose estimation. More recently, Choy *et al.* [1] use Non-Zero Whitened Histogram-of-Gradients (NZ-WHO) features [7] to synthesize, on the fly, discriminative appearance templates from 3D CAD models, for several poses, scales on multiple CAD model instances of the object, to jointly estimate the viewpoint and the instance associated with the object. In particular, Pepik *et al.* [12,13] rephrase the DPM framework [3] to formulate a structured learning output predictor to estimate the 2D bounding box of the object along with its viewpoint by enriching the object appearance model using 3D CAD data. The combination of robust DPM matching with the representational power of 3D CAD models is shown to result in a boost in performance across several datasets. We aim to extend this work by learning from real, SfM shapes and associated image appearance models and also treat viewpoint using a full projection model instead of regression.

There has been progress in this regard. Hejrati and Ramanan [9] learned the 3D geometry and shape of the object from 2D part annotations using a non-rigid SFM technique. In particular, Hejrati and Ramanan [10] represent 2D object part appearances using a Gaussian mixture model (GMM) that captures the appearance variations due to variations in the viewing angle. Zia *et al.* [18] use a 3D shape representation scheme to jointly model multiple objects allowing them to reason about inter-dependencies between the objects, such as occlusion, in a more deterministic and systematic manner.

Our proposed method departs from the beaten path described above, through the following means: (i) employing automatically estimated, real world 3D shapes to learn deformable models (manually generated 3D CAD models are often lacking in appearance details (such as surface texture) and make simplifying approximations about the actual 3D geometry that undermine the challenges underlying the 3D object pose estimation and reconstruction problem), (ii) modelling the projection process for geometric reasoning instead of relying on regression models, (iii) solving the shape recovery and view estimation problems using an effective RANSAC based scheme (as opposed to the computationally intensive generative process of [10]) and (iv) using a fine-grained part representation, learnt from real data, to model the shape to a high resolution and accuracy for more complex analysis in the future. The pipeline of the proposed RANSAC based scheme for shape recovery and viewpoint estimation is shown in Fig. 1

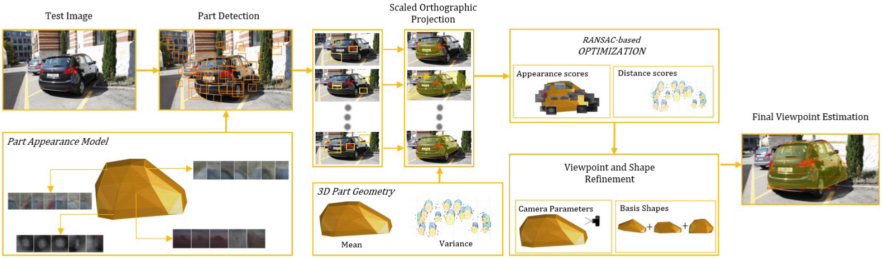

Fig. 1. The proposed object shape and pose (or viewpoint) estimation pipeline. Given a test image, we perform part candidate detection using the learned mixture-model-based part appearance model, followed by viewpoint (scaled orthographic cameras) estimation using a RANSAC-based scheme. The optimization gradually fits more deformation to the shape to recover a realistic reconstruction with a refined camera estimate.

3 The Proposed Method

3.1 Problem Statement

Given a set of image sequences of the same object class, *e.g.* cars, each sequence being taken around a single object instance, our objective is to reconstruct the shape and pose of a new instance observed in a new input image. More precisely, we aim to learn a deformable shape model for the particular object class, which then allows us to estimate both the best deformed shape and the 3D object pose in a new image such that visible semantic, salient parts of the object project to their 2D observations in the image. The latter also involves an occlusion reasoning for the new viewpoint and object instance.

To formulate this, we will use the following notation. Each of the K uncalibrated image sequences given in the training set, indexed by $k \in \{1 \dots K\}$, contains N_k images taken around an object instance. Let us denote the n-th image in the k-th sequence by I_{nk}, and its associated 3×4 camera (projection) matrix by $\mathbf{C}_{nk}$, which encodes the relative object pose and the camera intrinsics. The estimation task is to predict the full projection model $\mathbf{C}$ and the 3D shape S of the object in a new uncalibrated input image I. For simplicity, we define the 2D object detection mask in I as projection of the fitted shape instance through the estimated camera.

3.2 A Class-Specific Deformable Model

There are different ways to represent the shape of an object instance that is of a particular object class. Due to its simplicity and elegance, we have chosen to use a semantic part-based construction in combination with a linear subspace deformation model.

We define the shape S of any object instance via the 3D positions of its P semantic parts in space. The set of parts is predefined per object class. If $\mathbf{s}_p$ is the position of the p-th part of an object instance, then the shape of this instance can be encoded by a $3 \times P$ matrix $\mathbf{S} = \left[\mathbf{s}_1 \ldots \mathbf{s}_P\right]$. The linear subspace model describes any shape as a linear combination of a set of L known basis shapes which capture the modes of variation in the training data. Thus, the shape matrix of a particular object instance is $\mathbf{S} = \sum_{l=1}^{L} \alpha_l \mathbf{B}_l$, where $\mathbf{B}_l$ is the $3 \times P$ matrix of a basis shape and α_l is the corresponding coefficient.

Assume that the basis shapes $\{\mathbf{B}_l\}_{l=1}^{L}$ are known from a training phase for an object class for now. Then given a new image I depicting an instance of the same object class, the objective is to compute the shape matrix $\mathbf{S}$ of the depicted instance, as well as the camera (projection) matrix $\mathbf{C}$ that maps 3D parts of the object to its observation in the new image I. The 2D projection $\hat{\mathbf{x}}_p$ of a 3D part location $\mathbf{s}_p$ can be formulated as

$$\hat{\mathbf{x}}_p = \rho(\mathbf{C} \cdot \mathbf{s}_p) = \rho\left(\mathbf{C} \sum_{l=1}^{L} \alpha_l . \mathbf{b}_{lp}\right) \tag{1}$$

where $\mathbf{b}_{lp}$ is the p-th column of basis shape matrix $\mathbf{B}_l$, $\rho(.)$ is a mapping that maps any vector (u, v, w) with $h \neq 0$ to $(u/w, v/w)$. The camera matrix $\mathbf{C}$ can describe a perspective or an orthographic projection. However, not all points on the surface of an object are visible in an image. The binary visibility state of a 3D point $\mathbf{s}$ in an image I of camera matrix $\mathbf{C}$ is modeled by a boolean variable $v(\mathbf{s}, \mathbf{C}) \in \{0, 1\}$, where 0 stands for *occluded* and 1 for *visible*.

Given the matrices of the basis shapes $\{\mathbf{B}_l\}$, the shape of an object instance is fully determined by its deformation parameters $\{\alpha_l\}$. The loss function for computing the shape matrix $\mathbf{S}$ and the camera matrix $\mathbf{C}$ of an object instance depicted in a query image I can be defined as the sum-of-squared Euclidean distances between the projections and the observations $\mathbf{x}_p$ of the visible object parts in image I:

$$L(\{\alpha_l\}, \mathbf{C}) = \sum_{p=1}^{P} v(\mathbf{s}_p, \mathbf{C}) \cdot ||\mathbf{x}_p - \rho(\mathbf{C} \cdot \mathbf{s}_p)||^2, \qquad \mathbf{s}_p = \sum_{l=1}^{L} \alpha_l \mathbf{b}_{lp}, \tag{2}$$

where the vectors $\mathbf{b}_{lp}$ are known from the training phase. The joint shape-pose problem for an input image I can be solved by a minimization of L with respect to the shape coefficients $\{\alpha_l\}$ and projection parameters $\mathbf{C}$.

The loss function for the training phase can be obtained in a similar fashion. There, the squared projection errors of K object instances needs to be measured over all images of the training set. The loss function for training can be written as

$$L_T = \sum_{k=1}^{K} \sum_{n=1}^{N_k} \sum_{p=1}^{P} v(\mathbf{s}_{kp}, \mathbf{C}_{nk}) \cdot ||\mathbf{x}_{klp} - \rho(\mathbf{C}_{nk} \cdot \mathbf{s}_{kp})||^2, \quad \mathbf{s}_{kp} = \sum_{l=1}^{L} \alpha_{kl} \mathbf{b}_{lp}, \tag{3}$$

where $\mathbf{s}_{kp}$ is the 3D location of the p-th part of the k-th object instance, and $\mathbf{C}_{nk}$ is the camera matrix corresponding to the training image I_{nk} as introduced in Sect. 3.1.

In the followings, we present our approach for learning the basis shapes and part appearance from multi-view 3D mesh reconstructions of our input sequences.

3.3 From Dense 3D Reconstructions to Part-Based Shape Models

In order to learn the 3D basis shapes, a 3D surface model of each object instance of the training set is needed. Moreover, we will augment the shape model with an image-based appearance model per object part (Sect. 3.5). This requires the additional knowledge of all camera matrices $\mathbf{C}_{nk}$ for the training images I_{nk}. We now discuss how these prerequisites are obtained and postpone the learning algorithms to Sects. 3.4 and 3.5.

Prior to training, we first apply a state-of-the-art 3D reconstruction pipeline to each sequence, separately. A Structure-from-Motion (SfM) procedure computes the camera matrices $\mathbf{C}_{nk}$, while a dense Multi-View Stereo (MVS) and surface reconstruction algorithm computes a triangle mesh surface of the visible surface areas of the scene, given the camera models. We use 123DCatch that integrates all these steps, but note that other similar tools are also possible here. As a result, each 3D object instance in the training set is reconstructed as a mesh with an arbitrary number of vertices and triangles (see Fig. 2). Intra-class variations and the varying vertex counts make meshes difficult to relate, not to mention that most vertices may not correspond to any salient entity on the object surface or its corresponding images.

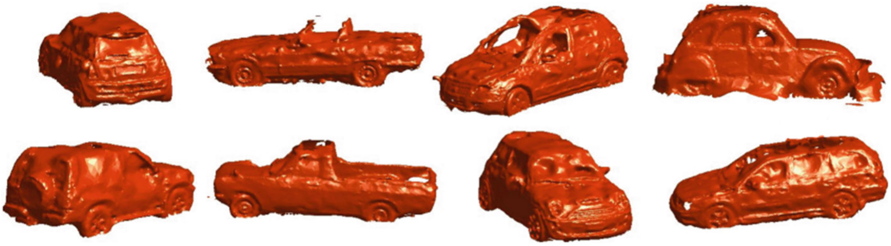

Fig. 2. Training set examples for 'car': 3D meshes obtained from real-world 2D image sequences from 123DCatch. These models are used for data-driven 3D geometric reasoning throughout the paper. Note the intra-class shape variability.

In a subsequent step, we annotate each 3D mesh (Fig. 2) with a fixed set of parts (up to the closest vertex location), where each part is a repeatable and semantically meaningful region of the object, *e.g.* (center of) *front-left-wheel* or *rear-licence-plate*. The 3D part annotations are obtained via an intuitive user interface by performing a multi-view triangulation of part annotations from two or more images observing the same object instance. As a result, each object instance (indexed by $k \in \{1 \ldots K\}$) yields an ordered set of 3D object part locations $\{\mathbf{s}_{kp}\}_{p=1}^{P}$.

Once the 3D meshes are annotated, the coordinate frames can be aligned using the part annotations. Due to the shape variations, this gives a more accurate alignment than simply applying the Iterated-Closest-Point (ICP) algorithm in our experience. Figure 3 shows the scatter of object part locations (across training instances), as well as the covariance ellipsoids (corresponding to 1σ) to visualize intra-class shape variations in our example training set.

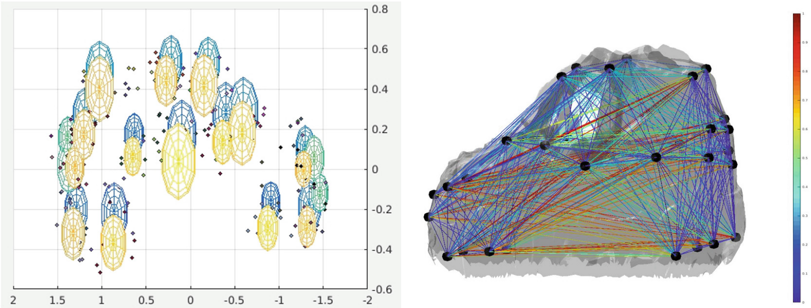

Fig. 3. 3D Part Geometry. *Left:* The standard deviation of part location is plotted in spheres (yellow on car's right, blue on the left). Interestingly, the front door handles vary considerably in location, while the bumpers and lights, not so much. *Right:* Variances in the mutual distance between each pair of parts are overlaid on a sample graph (*red* corresponds to higher variation, *blue* corresponds to lower variation). (Color figure online)

3.4 Learning a Class-Specific Object Shape

Based on the 3D shape model discussed in Sect. 3.2, we perform a Principal Component Analysis (PCA) on the object part positions and retrieve the top M modes of deformation ($M = 4$ in all our experiments), which gives us a set of $L = M + 1$ basis shapes (where $\mathbf{B}_1$ is explicitely defined as the mean shape) for an effective and compact linear subspace model to describe the subspace of possible intra-class shapes.

3.5 Learning the Appearance of Object Parts

The shape bases define a subspace of possible shapes for a particular object class. However, we also need to understand the appearance of the class in order to efficiently relate the shape model to new images. For each object part in the 3D shape representation, we construct an appearance model.

For the training sequences, by estimating visibility and projection, an appearance model for each part is learnt from the ground truth image sequences under real illumination, projection *etc*. For every part, CNN features (*conv5* layer) are extracted from the input images at their projections (when visible), using publicly available network weights [14]. These weights are obtained via training on the ImageNet Challenge 2014 (ILSVRC) dataset based on the part annotations. A mixture model [3,10] over these CNN features is then used to represent the variation in appearance, viewpoint *etc.* We learn a binary SVM classifier for each mixture component of each part of the class, to act as a part detector in images.

3.6 Detecting an Object Shape and Pose in a Query Image

Given a new query image I, and the learnt shape subspace spanned by basis shapes $\{\mathbf{B}_l\}$, and given the appearance-based object part detectors based on deep features and on SVM classifiers, our goal is to jointly fit the deformable shape model and compute the camera matrix $\mathbf{C}$ for this image such that 3D part locations of the fitted 3D shape model project to corresponding part observations in the image. The corresponding loss function is formulated in Eq. 2. The proposed pipeline is outlined in Algorithm 1 (which also invokes Algorithm 2) (Fig. 4).

Algorithm 1. Shape recovery, pose estimation and detection

1: **Part Detection**. Possible candidates for part detections are collected by convolving the trained SVM weight filters on conv5 feature pyramids [25]. Filter responses across multiple scales are combined using Non-Maxima Supression followed by Platt's Scaling [22] to obtain the probabilities of positive responses, such that the responses of different SVM classifiers are comparable. Responses stronger than a certain probability (p=0.35) are considered plausible candidates for the next step.

2: **Viewpoint Estimation**. We find the best camera parameters to project the mean shape to the test image by performing a RANSAC-based view estimation routine explained in Algorithm 2. In this case, the minimal set needs to be size 3 and the unknown parameters correspond to those of scaled orthographic projection.

3: **Viewpoint and Shape Refinement**. We perform a subsequent pass of viewpoint refinement allowing for shape deformation. This is equivalent to optimizing Eq. 2, with respect to the deformation parameters $\{\alpha_l\}_{l=1}^{L}$ in addition to the scaled-orthographic camera parameters. The RANSAC-based procedure can be repeated, but in each pass, one more mode of shape deformation is considered for a stable, incremental optimization. Finally, the a minimal set of 5 2D part candidates is needed for estimation of the extra $L - 1 = 4$ basis shape weights. The optimization of the loss function in Eq. 2, is modified to reflect the new parameters.

4: **Object Mask**. The estimated deformable shape and camera parameters represent the best reconstruction estimate for this image. When projected to the image, this gives us an object detection silhouette for this image.

Algorithm 2. RANSAC-based Viewpoint Estimation Algorithm

```
Perform part detection using the trained part appearance classifiers to obtain *Filter Response* $\mathcal{F}$, on the test image. Threshold these to obtain a set of possible candidates.
for N iterations do
    Assemble a minimal set of randomly-sampled unique parts from the candidates (constraint: they must be simultaneously visible in at least one view).
    Fit the unknown parameters minimizing the projection loss between the mean 3D shape parts corresponding to the 2D minimal set selected above.
    Check for inliers, based on whether candidate detections are within threshold $\tau_1$ for the remaining visible parts projected according to the above derived projection.
    If the number of inliers are greater than $\tau_2$ then store this minimal set and the estimated parameters.
For the set with maximum inliers, re-estimate the parameters minimizing the projection loss, through least-squares fitting on all the inliers, instead of only the minimal set. This is the best parameter estimate.
```

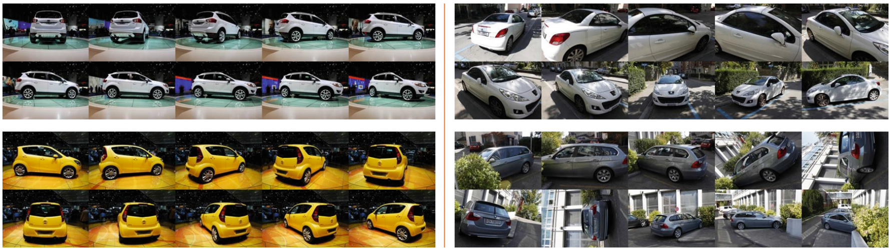

Fig. 4. Examples of 2D image sequences from the EPFL Multi-view Cars dataset (*Left*) and our RealCar dataset (*Right*).

4 Evaluation

4.1 Dataset

Our RealCar dataset consists of 35 image sequences taken around unique and distinct instances of cars, captured in real world conditions with challenging variations in scale, orientation, illumination with instances of occlusion. The total number of images per sequence varies between 30 and 115, across the dataset. When an SfM method like 123DCatch is used to estimate the car shapes and camera matrices, we get full mesh shapes along with full projection matrices (see Fig. 2). We use 29 of these sequences (and associated SfM results) for training and reserve 6 for testing.

The EPFL Multi-view cars dataset [20] contains image sequences of car instances on a turntable. Such sequences do not respond well to SfM preprocessing like RealCar dataset as the scene is not rigid, so this provide images to test on, but no ground truth 3D meshes or part annotations. This dataset is used purely as a second test set of images.

4.2 Experimental Setup

In this section, we evaluate the performance of our approach based on two tasks, (1) Viewpoint Estimation, to measure the accuracy of the estimated camera projection and, (2) Reconstruction, to measure how well the shape of the object in the test image is recovered.

Viewpoint Estimation: In order to evaluate the viewpoint estimation performance of the proposed approach, we run Algorithm 1 and report the Mean Precision of Pose Estimation [19] and Mean Angular Error [21], on individual images from the 6 test sequences of our RealCar dataset as well as from all 20 sequences of the EPFL Multi-view Cars dataset [20], where each car is imaged over a complete 360 degrees, with approximately one image for every 3–4 degrees. To measure viewpoint estimation accuracy we report our results using two standard metrics, Mean Precision of Pose Estimation (MPPE) [19] and Mean Angular Error (MAE) [21]. To report MPPE, we discretize azimuth angles (ϕ) into k number of bins where $k \in \{8, 12, 16, 18, 36\}$ and compute the precision of the viewpoint estimation for different number of bins. Table 1 shows the MPPE obtained using our approach on both images from our RealCar dataset and the EPFL dataset, and compares with Pepik et al. [13] and Ozuysal et al. [20], on EPFL dataset. Similarly, the Mean Angular Error [21], to evaluate the continuous viewpoint estimation performance of the proposed system, on both datasets is shown in Table 2 in comparison with Pepik et al. [13] and Glasner et al. [21] on the EPFL Multi-view cars dataset. In addition to estimating the Mean Angular Error for predicting the azimuth angle, we also estimate MAE for predicting all *3 Euler angles* [15], to provide a more accurate measure of performance of the proposed approach, for continuous viewpoint estimation. Table 3 shows MAE (Mean Angular Error) computed by estimating all *3 Euler angles*.

Table 1. Viewpoint Classification Accuracy using MPPE [19] on our RealCar dataset *(left)*, and on EPFL Multi-view Cars dataset [20] *(right)*. For our dataset, in addition to the test set, pose estimation experiments are also conducted on a subset of the training set to demonstrate the performance of the proposed approach in estimating viewpoint &recovering shape, on images, where the part detection accuracy is quite high.

θ	RealCar dataset		EPFL-Multiview cars dataset [20]		
	(Ours) Training set	(Ours) Test set	(Ours)	3D^{2}PM-D [13]	Ozuysal et al. [20]
$\pi/4$	93.79	86.09	59.86	78.5	-
$\pi/6$	89.44	79.13	50.06	75.5	-
$\pi/8$	83.85	71.30	40.47	69.8	41.6
$\pi/9$	78.26	65.22	36.67	71.8	-
$\pi/18$	46.58	43.48	19.22	45.8	-

Table 2. Continous/Fine-Grained Viewpoint Estimation error using MAE [21] on our dataset *(left)* and on EPFL Multi-view Cars dataset [20] *(right)*.

θ	RealCar Dataset		EPFL-Multiview Cars Dataset [20]		
	(Ours) Training set	(Ours) Test set	(Ours)	3D^{2}PM-D [13]	Glasner et al. [21]
$\pi/4$	13.02	14.13	17.35	12.9	24.8
$\pi/6$	11.88	12.35	13.58	9.0	-
$\pi/8$	11.05	10.87	10.68	7.2	-
$\pi/9$	10.32	9.92	9.58	6.2	-
$\pi/18$	5.47	6.2	4.81	5.2	-

Table 3. Continous/Fine-Grained Viewpoint Estimation using our Ransac-based viewpoint estimation technique, MAE [21] on EPFL Cars dataset [20] by computing all *3 Euler angles*.

θ	Our dataset		EPFL-Multiview cars dataset [20]
	Training set	Test set	(Ours)
$\pi/4$	16.08	18.28	31.48
$\pi/6$	14.92	16.31	22.71
$\pi/8$	13.15	14.45	17.27
$\pi/9$	12.32	13.54	15.06
$\pi/18$	6.92	6.59	8.05

The result tables show that our method performs very well on our dataset and competes well with the state of the art on the EPFL dataset, despite training on a smaller dataset appearance-wise. We report the viewpoint estimation accuracy on our dataset as well as on EPFL Multi-view cars dataset, we used our dataset (barely 29 3D object instances) to learn part appearances and 3D part geometry, and test it on EPFL Multi-view cars dataset. The performance of our approach relies heavily on the part detection performance generating inliers for at least a few parts. If part detections are even reasonable, the viewpoint/shape estimation is generally accurate, and so the accuracy on the RealCar dataset tends to be high (running our approach on the data that it has been trained on, shows best case results and an upper bound on how well our algorithm can do, due to the familiarity with appearance, though projection must still be figured out). The experiments show that, most of the bad viewpoint estimations are mainly due to bad part detection performance as shown in shown in Fig. 6 or mistakes due to symmetry of the car class.

Another important factor that affects the viewpoint estimation performance of our approach is the lack of a strong global appearance prior or a root filter. Unlike other regression based methods, we solely rely on detected 2D part locations for reasoning the 3D shape of the object, where slight anomalies with one or more part detections can cause a considerable error in the estimated final

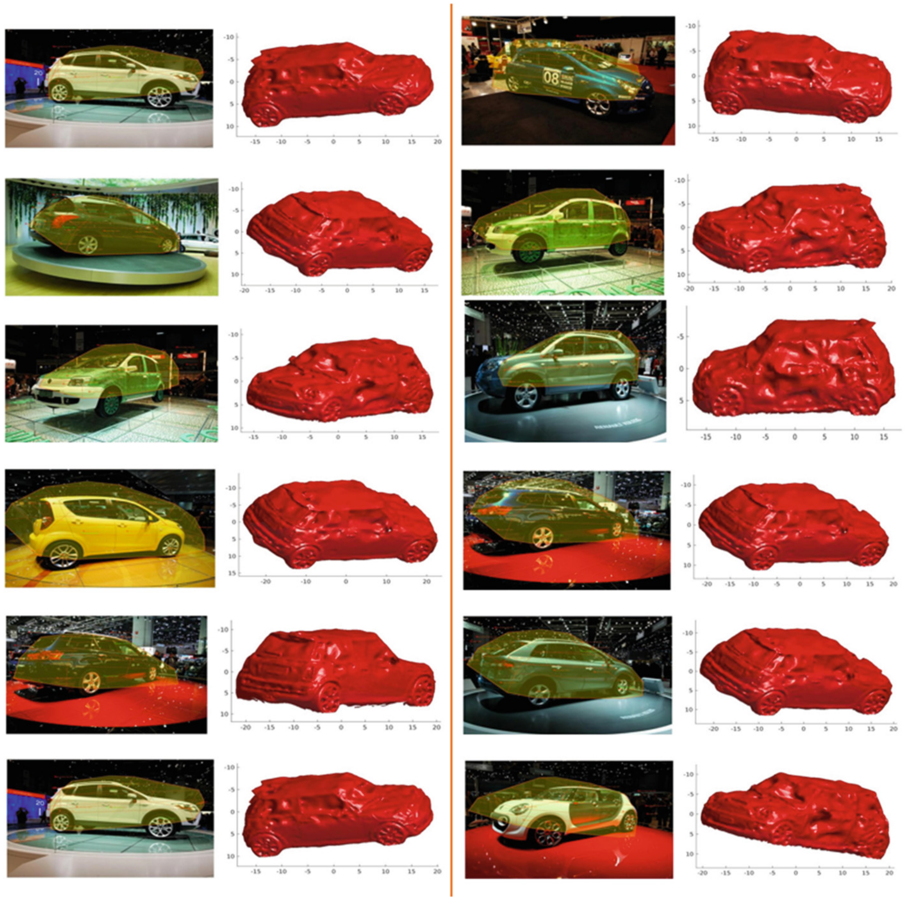

Fig. 5. Qualitative results of the proposed *RANSAC*-based Viewpoint Estimation and Shape Recovery, on EPFL Multi-view Car dataset. Odd columns illustrate the test image with corresponding Viewpoint/Shape estimations overlayed on it. Even columns illustrate the Viewpoint Estimation of their corresponding test image (on its left), using a sample mesh (from our dataset) for better visualization. (*note:* meshes (in even columns) are not generated/reconstructed by our viewpoint estimation approach, and are used only for the purpose of better visualization in all our qualitative results).

viewpoint. In the future, we will train robust part appearance classifiers over more appearance data with hard-mined data negatives, along with strong root filters, to try improving part detection accuracy and performance.

Figure 5 shows qualitative results on the EPFL dataset. Figure 6 demonstrates the challenges of part detection and appearance symmetry in viewpoint estimation success. Figure 7 shows the viewpoint/shape recovery results on our dataset. Also Fig. 8 compares the shape recovery results before and after the viewpoint and shape refinement step.

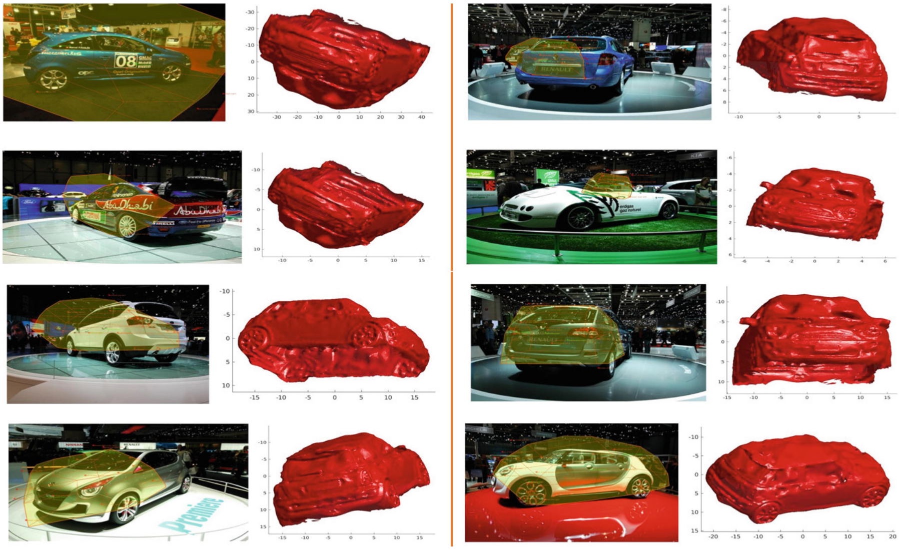

Fig. 6. A qualitative illustration on how the failure of part detection and the effect of symmetry in part appearances affect the the viewpoint estimation performance.

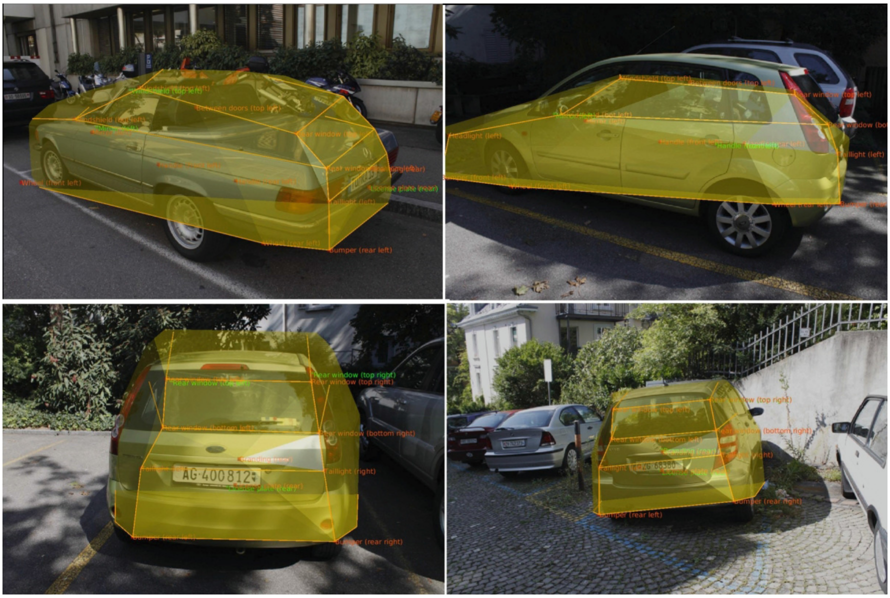

Fig. 7. Qualitative results of the proposed *RANSAC*-based viewpoint estimation on *our* dataset, with Viewpoint/Shape estimations overlayed on the object.

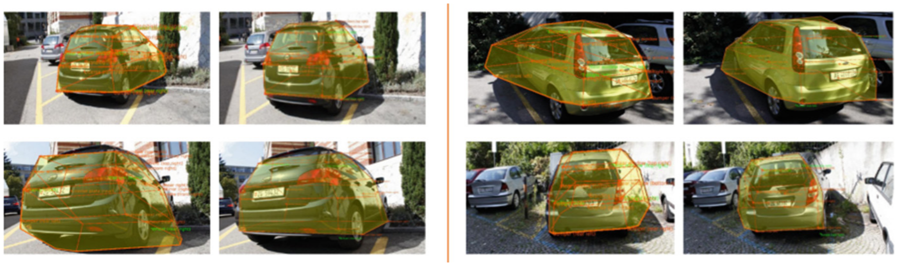

Fig. 8. An illustration on the improvement in Viewpoint and Shape Estimation due to the *Viewpoint and Shape refinement* step. Each pair of the image represents the Viewpoint and Shape estimations before *(left)* and after *(right)* the *Viewpoint and Shape refinement* step. The shapes on the right (of each pair) tend to be more compact and has a better viewpoint estimate, than the ones on the left.

Reconstruction. Unlike EPFL dataset, the RealCar dataset has the ground truth 3D parts annotated, so we can qualitatively compare the estimated 3D part based model with its actual ground truth, to evaluate shape accuracy. To report the shape recovery performance of our approach, we computed the average sum of squared distances between the estimated and ground truth 3D part locations of the object in the test image, for the 3D parts normalized to unit scale. The reconstruction/shape recovery error is 0.07 on the training set and 0.082 on test set of the RealCar dataset.

5 Conclusion and Future Work

We have shown qualitatively that our method for class-specific shape detection, recovery and pose estimation can yield good results on unseen state-of-the-art data as well as the original training data. We expect our RANSAC based process to be faster, while still efficient, than brute force exhaustive search and also models the projection process more accurately than regression. The fine-grained part representation and linear subspace representation allows us to model deformation effectively, but work with far fewer vertices than an SfM mesh with thousands of vertices. Importantly, we aim to learn such a part representation automatically, and automatically warp and improve the full mesh reconstructions also. As mentioned, better training and engineering should help perform even better. Going forward, using more image evidence (edges, contours, textures *etc.*) to fit the camera projection and reconstruction parameters, should allow for more accurate estimation. We could also perform GraphCut based segmentations for improved detection outlines.

References

1. Choy, C.B., Stark, M., Corbett-Davies, S., Savarese, S.: Enriching object detection with 2D–3D registration and continuous viewpoint estimation. In: Proceedings of IEEE CVPR (2015)
2. Felzenszwalb, P.F., Huttenlocher, D.P.: Pictorial structures for object recognition. IJCV **61**(1), 55–79 (2005)

3. Felzenszwalb, P.F., Girshick, R., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part based models. IEEE T-PAMI (2009)
4. Fenzi, M., Leal-Taix, L., Ostermann, J., Tuytelaars, T.: Continuous pose estimation with a spatial ensemble of Fisher regressors. In: Proceedings of ICCV (2015)
5. Fergus, R., Perona, P., Zisserman, A.: Object class recognition by unsupervised scale-invariant learning. In: Proceedings of IEEE CVPR (2003)
6. Ghodrati, A., Pedersoli, M., Tuytelaars, T.: Is 2D information enough for viewpoint estimation? In: Proceedings of BMVC (2014)
7. Hariharan, B., Malik, J., Ramanan, D.: Discriminative decorrelation for clustering and classification. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012. LNCS, vol. 7575, pp. 459–472. Springer, Heidelberg (2012). doi:10.1007/978-3-642-33765-9_33
8. He, K., Sigal, L., Sclaroff, S.: Parameterizing object detectors in the continuous pose space. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8692, pp. 450–465. Springer, Heidelberg (2014). doi:10.1007/978-3-319-10593-2_30
9. Hejrati, M., Ramanan, D.: Analyzing 3D objects in cluttered images. In: Proceedings of NIPS (2012)
10. Hejrati, M., Ramanan, D.: Analysis by synthesis: 3D object recognition by object reconstruction. In: Proceedings of IEEE CVPR (2014)
11. Lim, J.J., Khosla, A., Torralba, A.: FPM: fine pose parts-based model with 3D CAD models. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8694, pp. 478–493. Springer, Heidelberg (2014). doi:10.1007/978-3-319-10599-4_31
12. Pepik, B., Gehler, P., Stark, M., Schiele, B.: $3D^2$PM - 3D deformable part models. In: Proceedings of ECCV (2012)
13. Pepik, B., Stark, M., Gehler, P., Schiele, B.: Teaching 3D geometry to deformable part models. In: Proceedings of IEEE CVPR (2012)
14. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition arXiv preprint 2014. arXiv:1409.1556
15. Tulsiani, S., Malik, J.: Viewpoints and keypoints. In: Proceedings of IEEE CVPR (2015)
16. Xiao, J., Russell, B., Torralba, A.: Localizing 3D cuboids in single-view images. In: Proceedings of NIPS (2012)
17. Yu, T.-H.: Classification and pose estimation of 3D shapes and human actions. Ph.D. thesis, University of Cambridge (2013)
18. Zia, M.Z., Stark, M., Schindler, K.: Are cars just 3D boxes? Jointly estimating the 3D shape of multiple objects. In: Proceedings of IEEE CVPR (2014)
19. LÃşpez-Sastre, R.J., Tuytelaars, T., Savarese, S.: Deformable part models revisited: a performance evaluation for object category pose estimation. In: Proceedings of IEEE ICCVW (2011)
20. Ozuysal, M., Lepetit, V., Fua, P.: Pose estimation for category specific multiview object localization. In: Proceedings of IEEE CVPR (2009)
21. Glasner, D., Galun, M., Alpert, S., Basri, R., Shakhnarovich, G.: Viewpoint-aware object detection and pose estimation. In: Proceedings of IEEE ICCV (2011)
22. Platt, J.: Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In: Advances in large margin classifiers (1999)
23. Zia, M.Z., Stark, M., Schiele, B., Schindler, K.: Detailed 3d representations for object recognition and modeling. In: IEEE Transactions on PAMI (2013)
24. Kar, A., Tulsiani, S., Carreira, J., Malik, J.: Category-specific object reconstruction from a single image. In: IEEE CVPR (2015)

25. Girshick, R., Iandola, F., Darrell, T., Malik, J.: Deformable part models are convolutional neural networks. IEEE CVPR (2015)
26. Yingze Bao, S., Chandraker, M., Lin, Y., Savarese, S.: Dense object reconstruction with semantic priors. In: IEEE CVPR (2013)
27. Khosla, A., Zhou, T., Malisiewicz, T., Efros, A.A., Torralba, A.: Undoing the damage of dataset bias. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012. LNCS, vol. 7572, pp. 158–171. Springer, Heidelberg (2012). doi:10.1007/978-3-642-33718-5_12