Learning Hierarchical Models for Class-Specific Reconstruction from Natural Data

Arun CS Kumar
University of Georgia
aruncs@uga.edu

Suchendra M. Bhandarkar
University of Georgia
suchi@cs.uga.edu

Mukta Prasad
Daedalean AG
muktaprasad@gmail.com

Abstract

We propose a novel method for class-specific, single-view, object detection, pose estimation and deformable 3D reconstruction, where a two-pronged (sparse semantic and dense shape) representation is learned from natural image data automatically. Then, given a new image, it can estimate camera pose and deformable reconstruction using an effective, incremental optimization. Our method extracts a continuous, scaled-orthographic pose (without resorting to regression and/or discretized 1D azimuth-based representations). The method reconstructs a full free-form shape (rather than retrieving the closest 3D CAD shape proxy, typical in state-of-the-art). We learn our two-pronged model purely from natural image data, as automatically and faithfully as possible, reducing the human effort and bias typical to this problem. The pipeline combines data-driven deep learning based semantic part learning with principled modelling and effective optimization of the problem's physics, shape deformation, pose and occlusion. The underlying sparse (part-based) representation of the object is computationally efficient for purposes like detection and discriminative tasks, whereas the overlaid dense (skin like) representation, models and realistically renders comprehensive 3D structure including natural deformation, occlusion. The results for the car class are visually pleasing, and importantly, outperform the state-of-the-art quantitatively too. Our contribution to visual scene understanding through the two-pronged object representation shows promise for more accurate 3D scene understanding for real world applications on virtual/mixed reality, autonomous navigation, to cite a few.

1. Introduction

The ability to jointly reason about the shapes, viewpoints and locations of objects and their relative interactions from input images, is fundamental to almost all strands of research in computer vision, *e.g*., object recognition, scene reconstruction, content-based retrieval and problem parametrization. Each individual research strand in computer vision, has been pushed to its limits in terms of peak performance in the past few years; hence the final lap of progress lies in being able to perform joint reasoning and scene understanding. This will enable applications at the intersection of vision and robotics, *e.g*. robotic navigation, autonomous driving *etc*., to become more "intelligent", robust and efficient. The need for joint scene understanding has been long recognized but progress has been made in only small and steady steps owing largely to the fact that the underlying computer vision problems are ill-posed (many combinations of 3D object shapes and viewpoints can result in a given image), entail a combination of discrete and continuous variables and the input image data is often contaminated by noise with complex statistics that is difficult to model. However, the information extracted while solving one problem, such as object recognition, can help reduce the ambiguities in another, such as scene reconstruction. Class-specific treatment provides a useful, sensible prior, *e.g*., 3D instances of the *car* class share a similar topology and a learnable family of shapes and appearances. Additionally, recent advances in machine learning and optimization, allow the exploration of more powerful mathematical models and data-driven approaches for joint scene understanding.

As detailed in the abstract, we propose a novel approach to learn a two-pronged class representation for jointly solving class-specific object detection, pose estimation and deformable 3D reconstruction from a single 2D image. At test time, given an unseen image, the sparse model allows us to detect parts efficiently, reason shape deformation and model occlusion (self-occlusion), to estimate pose and recover the underlying 3D shape of the object. The dense model, subsequently, allows us to build on the recovered sparse representation to render a detailed 3D reconstruction of the object.

What's more, this work has implications on many real world applications such as VR/Motion capture games, self-driving cars *etc*. Before delving further, we will now discuss some related work and to better highlight how we adapt and advance it.

2160-7516/18 $31.00 © 2018 IEEE
DOI 10.1109/CVPRW.2018.00153

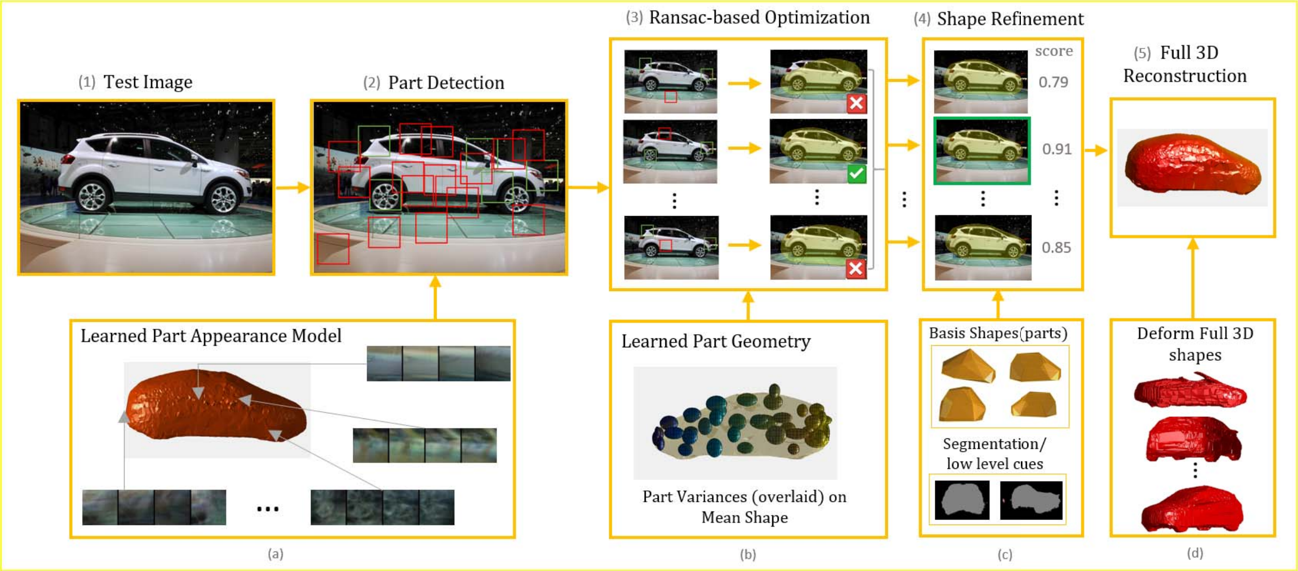
(1) Test Image
(2) Part Detection
(3) Ransac-based Optimization
(4) Shape Refinement
score
0.79
0.91
0.85
(5) Full 3D Reconstruction
Learned Part Appearance Model
...
(a)
Learned Part Geometry
Part Variances (overlaid) on Mean Shape
(b)
Basis Shapes(parts)
Segmentation/ low level cues
(c)
Deform Full 3D shapes
(d)

Figure 1: Pipeline: Given a set of class image sequences, a two-pronged (dense shape based + sparse part based) model is learned. The part based geometry and appearance model is learned as in (a, b). At test time, the part based model is used to bootstrap the view and deformation parameters in an initial optimization (3), followed by a refinement of these parameters and shape according to the dense shape model (4), to reconstruct a full 3D mesh (5).

2. Related Work

Object detection relies on powerful mathematical models that can represent the object shape, camera viewpoint and input noise effectively, and can be learned and deployed effectively. Early approaches to object detection and recognition employed sparse, 2D object representations (with considerable hand-crafted elements and some parameter learning) typically in the form of templates, pictorial structures [9] and constellations [11]). As the field progresses steadily (see PASCAL VOC [7]), abstract, less supervised and more data-driven representations [19] and joint object detection and object reconstruction [2, 15, 26] are being increasingly explored. In this work, we exploit the progress in deep learning [35] to discover 3D parts (spatial distribution and image appearance), characteristic of the class from real data, registered with respect to their dense shape reconstructions, thus providing a mapping between the semantics and shape. This extends the models of [11, 2, 26] to a more comprehensive 3D part representation derived fully and automatically from 2D data.

Interpreting and handling camera viewpoint well is crucial, even if it is only a means to an end (of recognition/reconstruction). Initial attempts at 3D object recognition/reconstruction were limited to frontal views [21, 30]; each new viewpoint used a new model [10]. Multiple viewpoints were often handled using discriminative, inverse modeling-based approaches based on classification [26] or regression [27]), wherein viewpoint was a 1D (sometimes 2D) variable. While these have performed well in limited evaluations, we model the physics of projection more faithfully (also see [15, 36]) and show that a 6-DOF, continuous, scaled orthographic projection, can be solved effectively in a RANSAC-based perspective-n-point-like ([16]) framework, rather than resorting to A* search or regression/classification.

Reconstruction (when combined with recognition) has spanned many ideas, from warping a shape [2] to approximative, coarse, depth prediction-based models [33] to wireframe reconstructions [22, 36, 15] over the years. In recent work [26, 36], reconstruction is approximated by retrieving a reasonably similar CAD model. Modelling deformation in a linear subspace from annotated data or 3D CAD datasets has also been proposed [15, 36, 26]. The deformation is estimated using a variety of cues, *e.g.* sparse part-based sampling [15, 23, 36], voting-based approaches [14] *etc*. Continuous optimization like [22, 23], which use image edges (or other features) for fitting, are interesting to us, as they allow for more fine-grained and effective optimization of shape and pose.

In recent times, the rapid increase in the availability of training data and the upsurge of sophisticated deep learning methods have led researchers to tackle this problem in a more data-driven manner. While most approaches simply harness the power of regression, there has been a few attempts to train deep neural networks end-to-end, using geometry-aware loss functions [17, 8]. In that regard, Choy *et al.,* [5] proposed a novel Recurrent Neural Network (with

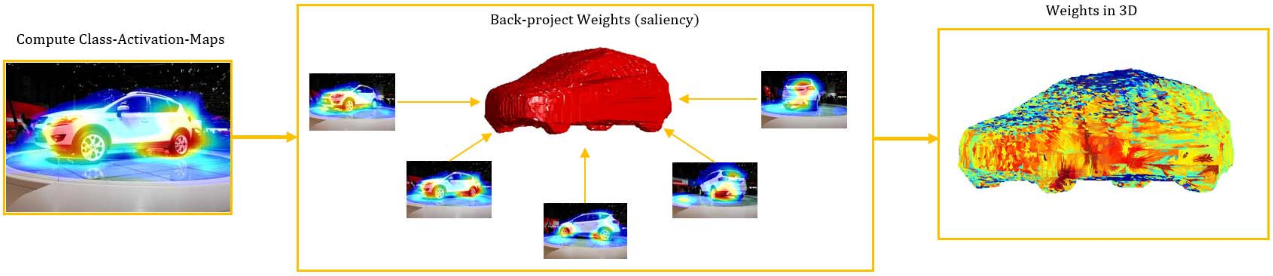
Compute Class-Activation-Maps
Back-project Weights (saliency)
Weights in 3D

Figure 2: A heatmap of the most salient regions in the image are computed (left) using *Class-Activation-Mapping* [35]. Backprojecting and aggregating the saliency maps from different views (center) results in a 3D estimate of saliency (right).

LSTM) based architecture for both single and multi-view reconstruction tasks, that learns mapping from 2D images to the underlying 3D shapes, using a large collection of synthetic data. At test time, given one or more images, the framework outputs the reconstruction in the form of 3D occupancy grid. Similarly, Gwak *et. al.,* [17] proposed a Generative Adversarial Network (GAN) architecture, where a 3D model generator that uses image masks and ray trace pooling to generate 3D shapes, alongside a discriminator which is trained using synthetic 3D shapes, to obtain smoother & realistic 3D reconstructions at test time. Both aforementioned approaches are able reconstruct object from single views, but the quality of reconstruction is comparable only as the number of views increases. On the other hand, Kurenkov *et al.,* [20] proposed DeformNET, an end-to-end single view reconstruction system that extends Spatial Transformer Networks [18] to learn geometric transformations in 3D, coupled with the use of Point Set Generation Network [8], where [8] demonstrated the use of point cloud in contrast to voxel or mesh representation, to be superior in terms of computation with ability to capture natural invariance for a single view object reconstruction problem. DeformNet architecture entails a Free-Form Deformation (FFD) layer that deforms the shape (represented as point clouds) using to achieve smooth geometric deformations.

Finally, the linear subspace that models deformation in most state of the art is still overwhelmingly learned from 3D CAD datasets, which are tedious to build and biased by artists. This approach emphasizes learning from real, arbitrary 2D image sequences, easy to collect, making generalization across classes more plausible. Progress in Structure from Motion and Multi-view Stereo based reconstruction [12] makes this increasingly realizable. Kumar *et al.,* [3] used Structure From Motion reconstructions generated from 2D image sequences (instead of 3D CAD models), for learning deformable shape representation. We extend [3] to model object shape using a more sophisticated two-pronged representation instead of a coarse deformable part model, which allows us to render a dense 3D reconstruction at test time, as opposed to a coarse wireframe reconstruction of [3]. Moreover, we discover parts in 3D automatically, instead of relying on manual annotations like [3]. In our approach, we use a combination of Structure From Motion [1] and space carving methods [12] to reconstruct arbitrary class-specific image sequences and register the dense reconstructions to learn a linear subspace. Additionally, we learn the sparser, semantically meaningful part representation (appearances and their spatial configurations). Our methods discovers the parts are most essential in identifying a class (see [35]); this is easier, scales to more classes and is more principled than manually annotated CAD data. The above two-pronged representation is very useful; the sparser part model allows one to detect and reconstruct the object effectively and bootstrap optimization, while the denser representation provides a comprehensive reconstruction. Since the parts and meshes deform together, one can reason about occlusion accurately, rather than heuristics and statistics.

To summarize, the major contributions of this paper are,

- A two-pronged model for sparse (part-based) and dense (comprehensive) representation of 3D meshes; for jointly solving class-specific object detection, continuous pose estimation and deformable dense 3D reconstruction of the test object instance.
- We use sequences of images taken around objects to learn the 3D shape representation (both dense and sparse), using SfM reconstructions as opposed to using tediously acquired CAD models.
- Instead of relying on data driven regression models for binning poses, we reason the image based evidence in a *PnP*-like scheme that is cognizant of surface occlusion, to estimate pose that is physically faithful.
- We represent the *sparse* part model using a linear shape subspace, where parts (3D) are discovered automatically from images —based on a combination of image saliency and appearances —and the SfM reconstructions (to project them to 3D); replacing the need for human part annotations,
- The final reconstruction is optimized using least squares minimization, in an incremental, stable manner.

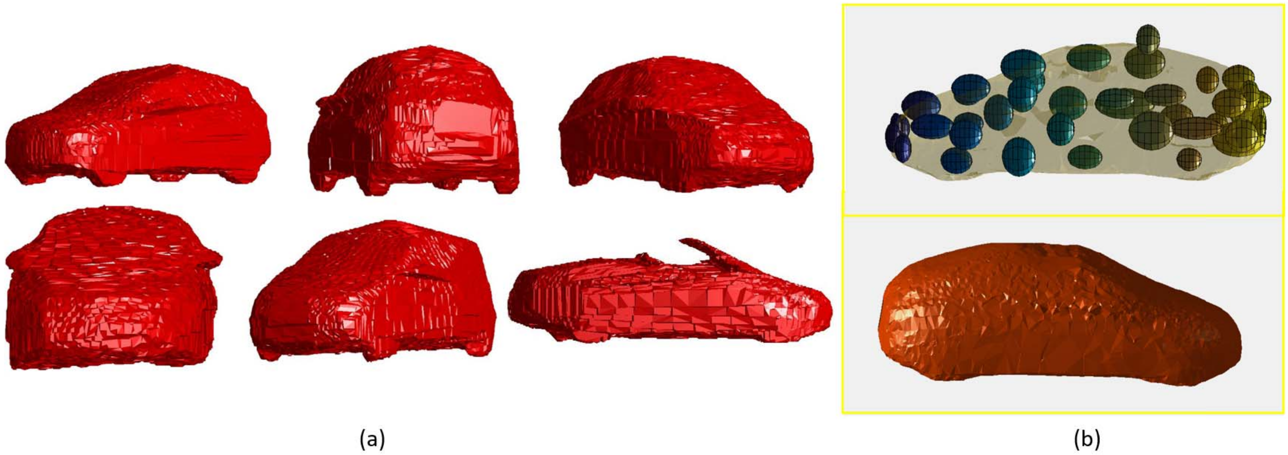

Figure 3: *(a)* Examples of 3D meshes obtained using space carving techniques on 2D image sequences. *(b)* Standard deviations of the part positions in 3D plotted using ellipsoids *(top)*. The larger the ellipsoid, the higher the standard deviation is in the corresponding direction. The mean car shape obtained by averaging full 3D meshes *(bottom)*.

3. Proposed Approach

We will now detail our approach, our model, how it is learnt and how to apply it to unseen test images.

3.1. A two-pronged shape model

The Dense Shape Model: A 3D shape instance of a class is given as $\underset{3\times P}{\mathtt{S}}$, which can be modelled by a linear combination of basis shapes so that each vertex is a weighted sum of the corresponding basis vertices: $\mathbf{s}_p = \sum_l \alpha_l \cdot \mathbf{b}_{lp}$, ($\boldsymbol{\alpha}$ are the shape coefficients). Note, the same representation applies to the dense shape mesh and the sparse set of parts. When viewed through a camera $\mathtt{C} = \mathtt{K}[\mathtt{R}\,\mathbf{t}]$ at a particular viewpoint, the projected shape vertices are given by:

$$\hat{\mathbf{u}}_p = \pi\left(\mathbf{C} \cdot \mathbf{s}_p\right), \qquad (1)$$

where $\pi(.)$ represents the perspective projection in a pinhole camera. Here we approximate this with a scaled orthographic camera. $\mathtt{K}$ represents the intrinsic camera matrix, and $\mathtt{R}$ and $\mathbf{t}$ stands for rotation matrix and translation vector.

Figure 3 shows examples of dense shape models learned from real images. Using real image sequences for the car class, each from around a unique 3D object class instance, and space carving based SfM [12], we reconstruct rigid shapes for each sequence, minimizing the image reprojection error. These meshes are normalized to a uniform mesh resolution, scale and registered in orientation using ICP. A linear subspace is then learned using Principal Component Analysis from this data. Also, learning 3D shapes directly from 2D images allows us to go back and forth between 2D and 3D representations easily (projection and back-projection); thus the appearance features (of parts) can be learned from 2D images, whereas their position and deformation can be reasoned about in 3D.

The Sparse Part Model: The sparse part model follows the conventional deformable representation of an object using a set of part positions in 3D. A part can be described as a salient regions of an object, that are repeatably identifiable, and invariant to deformation, viewpoint and illumination *etc*. Thus the proposed automatic 3D part discovery technique is based on finding the most distinctive and informative 2D features needed for identifying the category correctly against background. We use a Class Activation Mapping (CAM) based approach (based on [35], detailed in § 4) leveraging the *Global activation pooling* layer. A deep network is trained to perform a binary classification task (in our case *cars vs. not-cars*). Then, for every image fed into the network, the CAM approach outputs a heat map which can be understood as the relative importance of image features for the classification task.

The heat maps for training sequence images are then back-projected to its corresponding 3D instance reconstruction, which leaves us with a large number of 3D points (that correspond to salient image regions in 2D), across multiple 3D shape instances. The goal of our part discovery is to identify regions in 3D that are salient as well as repeatable (occurs in multiple shapes). For that purpose, these 3D part estimates are then clustered using K-means, where the distance between two parts is a sum of their 3D distance (in the normalized frame of reference) and distance in the appearance space (between feature descriptors extracted from their corresponding 2D image regions). This extracts a sparse set of parts (with consistent appearance and location), in the registered and normalized frame of reference. For each part a Gaussian 3D location distribution and a mixture model for part image appearance is learned (see § 4).

3.2. Pose Estimation

At test time, given a query image, we first convolve the learned part appearance classifiers on the test image, to obtain a set of candidate part detections. In the correct camera pose, the correctly deformed shape instance should project to the image, so that the visible parts match their projections in appearance. This reasoning is performed using a RANSAC-based perspective-n-point [16] like approach described in Algorithm 1. We search randomly sampled part combinations to jointly estimate the best view projection parameters assuming a mean shape, that agrees best with the visual evidence. To reason pose, the RANSAC scheme only uses part subsets that are jointly visible, for accurate 3D-2D fitting (joint visibility statistics can also be learned during training for efficiency). Each fit is evaluated by maximizing the part appearance score, and minimizing the distance between estimated and actual part positions along with a *root filter*. The root filter as used by [10], is a categorization the pose of the object (as a whole) into fixed-sized viewpoint bins, that acts much like a regularizer (more details in Section 4).

Algorithm 1 RANSAC-based Pose Estimation Algorithm

```
1: A set of candidate parts are obtained by performing part detection on the test image.
2: for N iterations do
3:     A minimal set of parts candidates are chosen randomly. In this case the minimal set needs to be of size 3 (to compute Scaled Orthographic projection). The chosen part candidates must be collectively visible in at least one of the views.
4:     Estimate the pose and deformation parameters by estimating reprojection loss between the mean shape and the 2D part positions chosen in step 1 (above).
5:     Check for inliers (support), the remaining parts (or corresponding part detections in 2D), that satisfies visibility criteria, projects within a threshold $\tau_1$.
6:     Store the set and the estimated parameters if the number of inliers are greater than threshold $\tau_2$.
7: Re-estimate the parameters minimizing the projection loss for all inliers (candidate parts), through least-squares fitting, instead of just the minimal set, to obtain the best parameter estimate.
```

3.3. Pose and Shape Refinement

The estimation is further refined by estimating the shape deformation parameters in addition to refining the camera parameters. We perform a least-squares optimization but instead of using just the mean shape, we estimate the shape deformation parameters $\boldsymbol{\alpha}$ in addition to the camera parameters $\mathtt{C}$ (initialized from the previous estimation) allowing the optimization to jointly reason about the test object's shape deformation and pose. For a query image I, the loss function can be defined as the difference between the projected and observed object parts:

$$L_1(\boldsymbol{\alpha}, \mathtt{C}) = \frac{1}{P}\sum_{p=1}^{P} v(\mathbf{s}_p, \mathtt{C}) \cdot ||\mathbf{u}_p - \underbrace{\pi(\mathbf{C} \cdot \mathbf{s}_p)}_{\hat{\mathbf{u}}_p}||^2 \quad (2)$$

where $v(\mathbf{s}_p, \mathtt{C})$ is boolean with a value of 1 if $\mathbf{s}_p$ is visible to $\mathtt{C}$ and 0 for occluded parts. $\hat{\mathbf{u}}_p \in \hat{\mathbf{U}}$ is the 2D projection of 3D part $\mathbf{s}_p$ where $\mathbf{u}_p \in \mathbf{U}$ is corresponding part detection. The shape and pose parameters are estimated by minimizing L with respect to $\boldsymbol{\alpha}$ and $\mathtt{C}$. The loss function in equation 2 can be augmented with terms that score a part in terms of how well the image appearance matches a learned model $p(\hat{\mathbf{u}}_p|\gamma_p)$ and how likely its relative 3D position is with respect to the learned related 3D part distribution $p(\hat{\mathbf{s}}_p|\delta_p)$ similar to [28]. Thus the loss for a part p projected using camera $\mathbf{C}$ and shape (α) parameters is given as:

$$L_2(\boldsymbol{\alpha}, \mathtt{C}) = -\frac{1}{P}\sum_{p} \left(\ln(p(\hat{\mathbf{u}}_p|\gamma_p)) + \ln(p(\hat{\mathbf{s}}_p|\delta_p))\right) \quad (3)$$

We additionally employ a regularizer in our optimization based on off the shelf object detectors like [10]. Here, we convolve the learned root filters with the image to get possible image bounding boxes that the parts should lie within. This is converted to a map with low values inside the bounding box detections (and high outside). The loss term minimizing the cost of projected part positions $\hat{\mathbf{u}}_p$ on this map can be evaluated as :

$$L_3(\boldsymbol{\alpha}, \mathtt{C}) = -\frac{\lambda}{P}\sum_{p} \mathrm{map}^2(\hat{\mathbf{u}}_p) \quad (4)$$

Thus the total loss between estimated projection and actual image evidence can be formulated as:

$$L(\boldsymbol{\alpha}, \mathtt{C}) = L_1(\boldsymbol{\alpha}, \mathtt{C}) + L_2(\boldsymbol{\alpha}, \mathtt{C}) + L_3(\boldsymbol{\alpha}, \mathtt{C}) \quad (5)$$

In Algorithm 1, steps 4 to 5 minimizes L_1 loss (equation 2), where the shape refinement (step 7) minimizes the total loss $L(\boldsymbol{\alpha}, \mathtt{C})$ (equation 5). Please refer to Section 4 for a detailed explanation of the optimization.

Full Reconstruction: Estimating the deformation and camera parameters allows us to recover the underlying the skeleton of the shape instance in the test image along with its viewpoint. To perform a complete 3D reconstruction, we deform the full 3D meshes corresponding to the basis shapes used for estimating the shape of the object instance, using the deformation weights estimated above.

4. Implementation Details

The performance of our approach depends on the implementation of several units, described at a high level above.

CAM based learning: We use the publicly available pre-trained weights of VGGnet 16-layer architecture [32], trained on ImageNet challenge [31]. We modify the network architecture by replacing the fully connected layers (layers after *conv5-3* in VGGnet) with a *global average pooling* layer, which simply computes the spatial average of the feature map (from *conv5-3*) at each unit κ, whose weighted sum outputs the final saliency map. The global average pooling layer is then followed by a softmax layer that outputs the likelihood of the image belonging to the category or not (for two-way/binary classification). We learn weights corresponding to each class for each unit κ. The weighted sum of the feature maps of the *conv5-3* layer is used to compute class activation maps.

Learning the sparse part model: After discovering parts from the clustering process of 3.1, the location distribution is learned by fitting a Gaussian distribution to the 3D clusters. Part appearance models are learned as a mixture model similar to [10, 15]; for each part we train individual SVM classifiers for each discrete viewpoint bin in which the part is visible. We discretize the viewing sphere into 12 bins and on an average each part has 3-6 mixture components (classifiers). Part appearance classifiers operate on CNN-based *conv5* layer features (see [13]) extracted from the images. To train part appearance classifiers, we fine-tune the top layers of the 16-*layer* VGGnet [32] (pretrained for ImageNet Classification task [31]), to adapt the network to perform a 12-*way* viewpoint classification task (suggested by [6] for domain-specificity). We categorized the Epfl Multi-view Cars dataset [25], into 12 bins (of 30° each). Since our goal is to fine-tune the fully connected layer weights and the last convolutional layer weights, we freeze the weights of the first three sets of convolutional layers *(conv1 - conv3)* for faster learning. We replace the last (1000-way softmax classification) layer with a 12-way softmax layer. We then train the modified network for viewpoint classification task, so that the weights (*conv* layer) are fine-tuned to adapt to our dataset, and also the network learns to discriminate appearances cues (of parts/regions) with respect to viewpoint.

Supplementary model: In addition to the sparse part and dense shape model, we learn a supplementary view-specific root filter model that learns the holistic object appearance conditioned on the viewpoint [10, 26]. The root filter based view proposals are an additional cue in RANSAC based estimation process and help filter out the incorrect poses estimated due to object symmetry. The root filter is learned by binning each image into a certain viewpoint interval and then we extract *fc7 (fully connected layer)* features from the images (using the fine-tuned CNN model to train sets of SVM classifiers, one per viewpoint bin.

5. Evaluation

Most available 3D object datasets such as Pascal3D [34] or ShapeNet [4], use synthetic CAD models for 3D representation of objects along with manually annotated pose, thus not feasible for demonstrating our work. In order for us to learn 3D shape representation from images, we need datasets of image sequences taken around the object. We demonstrate the performance of our proposed framework on EPFL-Multiview [25] Cars dataset, one of the most commonly used datasets for viewpoint estimation problems. EPFL-Multiview Cars dataset [25] contains 20 sequences of images taken around car instances, We randomly split the dataset into two and use 10 sequences for training and 10 for testing. Below are a few evaluation metrics on which we evaluate the performance of our proposed framework.

5.1. Part Detection

Part detections are obtained by convolving learned part appearance filters (SVM classifiers), on *(conv5)* features extracted from the test image pyramid. We combine part detections across all scales, and use non-maxima suppression to remove redundant bounding boxes. Then each part detection score is approximated to a probability using Platt scaling [29]. To evaluate our part detection performance, we use the standard evaluation metric of [7]. A part detection is considered valid if there is a 50% overlap between the groundtruth and the detected bounding box. Also we use Mean Average Precision (mAP) to evaluate the performance of our part detection. The mAP of our part detection system on Epfl-cars dataset [25] is 45.97%.

5.2. Estimation of Camera Parameters

In order to evaluate the viewpoint estimation performance of our system, we compute Mean Precision in Pose Estimation (MPPE) [24] and Median Angular Error (MAE) [14]. MPPE is a measure of viewpoint classification accuracy where we discretize azimuths into k number of bins and compute the classification accuracy using precision for the different number of bins. On the other hand, MAE is for fine/continuous viewpoint estimation, where we compute the median of the angular error between the estimated and groundtruth viewpoints (camera parameters). Table 1 and Table 2 shows the MPPE and MAE respectively, obtained using our approach on Epfl-cars dataset [25] and compares our approach with Pepik et. al [26], Ozuysal et. al. [25] and Lopez-Sastre et. al. [24]. Table 1 shows that our approach consistently outperforms the current state-of-the-art [26] in this dataset, especially the Discrete version of 3D^{2}PM, despite [26] uses synthetic images in addition to real images, to train appearance models.

In addition, we also report our *Top-N* accuracy using MPPE for the Epfl-cars dataset [25] in Table 3. A classification is considered correct if one of the *top-N* estimates

θ	EPFL-Multiview Cars [25]				
	(Ours)	3D^{2}PM-D [26]*	3D^{2}PM-C Lin [26]*	[24]	[25]
$\pi/4$	99.4 / **88.74**	99.4 / 78.5	97.8 / 78.3	91.0 / 73.7	-
$\pi/6$	99.4 / **83.07**	97.9 / 75.5	98.3 / 76.2	-	-
$\pi/8$	99.4 / **76.85**	99.0 / 69.8	97.5 / 69.0	97.0 / 66.0	85.0 / 41.6
$\pi/9$	99.4 / **72.94**	99.2 / 71.8	99.3 / 71.2	-	-
$\pi/18$	99.4 / 46.21	99.3 / 45.8	99.2 / **52.1**	-	-

Table 1: Viewpoint Classification Accuracy using MPPE [24] on EPFL Multi-view Cars dataset [25]. (* - 3D^{2}PM [26] uses synthetic images generated from 3D CAD models for better appearance training in addition to real images. It is unfair to directly compare the performance between the two, as we rely only on available real image data for training).

θ	EPFL-Multiview Cars [25]				
	(Ours)	3D^{2}PM-D [26]	3D^{2}PM-C [26]	3D^{2}PM-D [26]*	[14]
$\pi/4$	12.84	13.1	13.7	12.9	24.8
$\pi/6$	9.04	-	-	9.0	-
$\pi/8$	7.14	-	-	7.2	-
$\pi/9$	6.70	7.4	7.0	6.2	-
$\pi/18$	4.68	6.4	5.6	5.2	-

Table 2: Continous/Fine-Grained Viewpoint Estimation error using MAE [14] on EPFL Multi-view Cars dataset [25].(* - 3D^{2}PM-D [26] uses synthetic images generated from 3D CAD models for better training).

θ	*top-2*	*top-3*	*top-5*
$\pi/4$	91.14	92.07	92.71
$\pi/6$	87.22	88.32	89.46
$\pi/8$	78.35	79.61	81.58
$\pi/9$	74.42	75.84	77.98
$\pi/18$	48.81	51.83	55.61

Table 3: *Top-N* accuracy for Viewpoint Estimation (MPPE [24]) using our Ransac-based viewpoint estimation technique, on EPFL Cars dataset [25].

θ	(ours)	[3]
$\pi/4$	0.4713	0.5492
$\pi/6$	0.3680	0.3931
$\pi/8$	0.2791	0.3011
$\pi/9$	0.2280	0.2628
$\pi/18$	0.1272	0.1404

Table 4: Continous/Fine-Grained Viewpoint Estimation using MAE (Error/distance between quaternions, where a distance of 3.14 = 180°) on EPFL Cars dataset [25] using all *3 Euler angles*.

are correctly classified. This metric accounts for how consistent the system is in predicting, and when it fails how badly it fails. The system is consistent if most of *top-N* detections are accurate. In addition, since the solution space is highly noisy, even if the most confident detection is inaccurate, computing top-N accuracy provides insight into how close our other estimates are. Analyzing the results further, we identify that one of the most important factors that affects the viewpoint estimation accuracy is the misclassification due to the appearance symmetry or the ill-posed nature of the problem. Though most deformable part based models are engineered to address the ill-posed nature of the viewpoint estimation problem, estimating viewpoint with high precision is quite challenging. We conducted an experiment to see what percentage of mis-classifications fall in the viewpoint range (of 30°) in the opposite side of the actual viewpoint. We found that $\{41.61, 25.91, 16.92, 13.87, 8.71\}$ percent of the total misclassification (MPPE) for bin sizes $\{\pi/4, \pi/6, \pi/8, \pi/9, and\, \pi/18\}$ respectively, are due to the appearance similarity. This experiment also shows that a considerable amount of our error is due to the ill-posed nature of the problem and not due to other noise.

5.3. Shape Estimation

We evaluate the full 3D reconstruction performance of our framework quantitatively by computing pixel-level accuracy of the projected 2D silhouette of the reconstruction (using the estimated shape and camera parameters) with respect to the groundtruth segmentation. The full 3D reconstruction using the estimated camera and shape parameters projects into a 2D silhouette. Ideally, if we are able to accurately estimate shape and camera parameters, the reconstructed shape/mesh

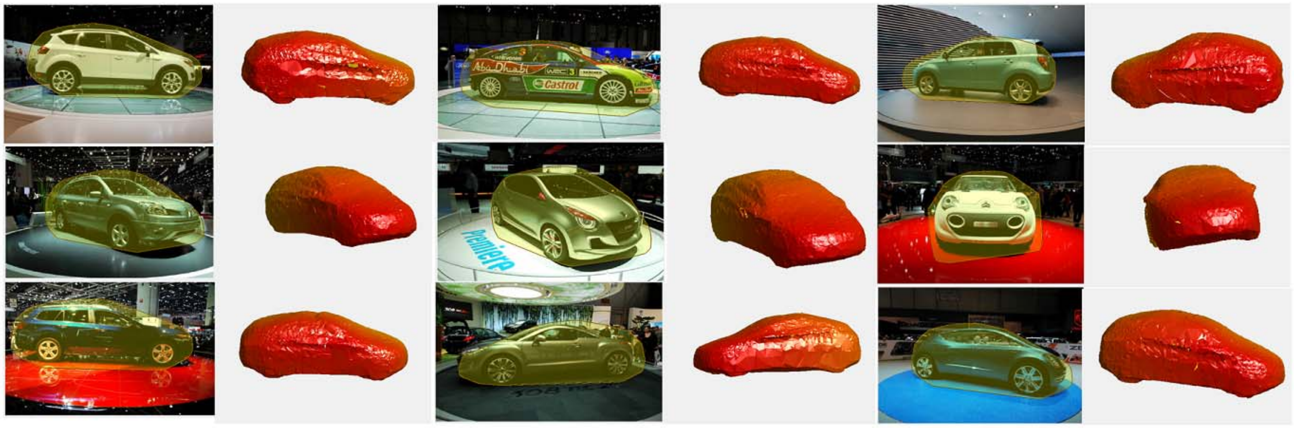

Figure 4: Qualitative results. *Left:* Test images overlaid with the projection (2D convex hull) of the estimated shape and viewpoint. *Right:* Full 3D reconstruction of the test object instance obtained by deforming basis shapes (meshes) using estimated shape parameters rotated to the estimated pose.

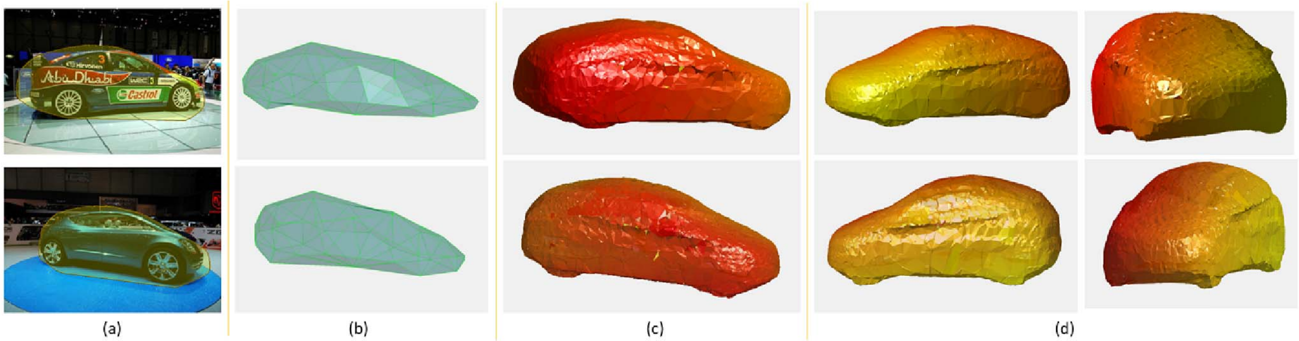

Figure 5: Qualitative results. *(a)* Test images overlaid with the projection (2D convex hull) of the estimated shape and viewpoint. *(b)* Wire-frame reconstruction (of estimated part positions) *(c)* Full 3D reconstruction (mesh) of the test object instance rotated *wrt.* estimated camera parameters*(d)* The full 3D reconstruction rotated by different angles to demonstrate the details of the estimated shape better.

would project compactly into the object's (groundtruth) silhouette, resulting in a high overlap (close to 100%). We use 220 manually segmented images from the test set of Epfl-cars dataset [25] as the groundtruth to evaluate the full 3D reconstruction performance. Our 3D reconstruction framework has a *precision* (pixel-level) of 81.39% and a *recall* of 88.82%.

Figures 4 & 5 show qualitative results of the full 3D reconstructions on Epfl-cars dataset. Our framework does a pretty good job in reasoning the unknown shape of the test object instance based on the visual evidence. As shown, it is able to deform the shape estimates to render an accurate 3D reconstruction of the test object instance.

6. Conclusion

As promised, we have demonstrated an end-to-end pipeline that learns a two-pronged class model automatically from arbitrary class image sequence data. The dense shape representation allows for realistic deformable reconstruction and occlusion modelling, while the sparse model discovers a class part model based on class characteristics, which help in efficient and reliable view estimation and object detection and bootstrapping. We achieve qualitatively and quantitatively pleasing results, thanks to modelling the problem using continuous variables, realistic physics and an incremental and (largely) continuous optimization that learns completely from natural, 2D data (without tedious content creation, annotation and minimizing human bias). Going forward, we would like to test the power of this method for unorganized data from more classes and possibly leverage image sequences for improving reconstructions via temporal reasoning. The proposed two-pronged representation lend themselves readily to real world applications such as self-driving cars, mixed reality or motion capture based gaming *etc.*, that rely on modelling objects in 3D, for more accurate scene understanding, that in turn, aids decision making.

References

[1] Autodesk 123d catch. `https://en.wikipedia.org/wiki/Autodesk_123D`. 3

[2] Y. Bao, M. chandraker, Y. Lin, and S. Savarese. Dense object reconstruction using semantic priors. In *Proc. CVPR*, 2013. 2

[3] A. C. S. Kumar, A. B/odis-Szomur/u, S. Bhandarkar, and M. Prasad. Class-specific object pose estimation and reconstruction using 3d part geometry. In *Proceedings of the ECCV Workshops*, pages xx–yy, Oct 2016. 3, 7

[4] A. Chang, F. Thomas, G. Leonidas, H. Pat, H. Qixing, L. Zimo, and S. Silvio. Shapenet: An information-rich 3d model repository. In *arXiv preprint arXiv:1512.03012*, 2015. 6

[5] C. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *Proc. ECCV*, 2016. 2

[6] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *Proc. ICML*, 2014. 6

[7] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. 2, 6

[8] H. Fan, H. Su, and L. Guibas. A point set generation network for 3d object reconstruction from a single image. 2016. 2, 3

[9] P. Felzenszwalb and D. Huttenlocher. Pictorial structures for object recognition. *IJCV*, 61(1), 2005. 2

[10] P. F. Felzenszwalb, R. B. Girshick, D. A. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE PAMI*, 32:1627–1645, 2010. 2, 5, 6

[11] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proc. CVPR*, volume 2, pages 264–271, Jun 2003. 2

[12] A. W. Fitzgibbon, G. Cross, and A. Zisserman. Automatic 3D model construction for turn-table sequences. In R. Koch and L. Van Gool, editors, *3D Structure from Multiple Images of Large-Scale Environments, LNCS 1506*, pages 155–170. Springer-Verlag, Jun 1998. 3, 4

[13] R. Girshick, F. Iandola, T. Darrell, and J. Malik. Deformable part models are convolutional neural networks. In *Proc. CVPR*, 2015. 6

[14] D. Glasner, A. Galun, M., R. S., Basri, and G. Shakhnarovich. Viewpoint-aware object detection and pose estimation. In *Proceedings of the ICCV Workshops*, 2011. 2, 6, 7

[15] M. Hejrati and D. Ramanan. Analysis by synthesis: 3d object recognition by object reconstruction. In *Proc. CVPR*, 2014. 2, 6

[16] J. Hesch and S. Roumeliotis. A direct least-squares (dls) method for pnp. In *Proc. ICCV*, pages 383–390, 2011. 2, 5

[17] G. J., C. Choy, A. Garg, M. Chandraker, and S. Savarese. Weakly supervised generative adversarial networks for 3d reconstruction. 2017. 2, 3

[18] M. Jaderberg, K. Simonyan, and A. Zisserman. Spatial transformer networks. In *NIPS*, 2015. 3

[19] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In P. Bartlett, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors, *NIPS*, pages 1106–1114. 2012. 2

[20] A. Kurenkov, J. Ji, A. Garg, V. Mehta, J. Gwak, C. Choy, and S. Savarese. Deformnet: Free-form deformation network for 3d shape reconstruction from a single image. 2017. 3

[21] B. Leibe, A. Leonardis, and B. Schiele. Robust object detection with interleaved categorization and segmentation. *IJCV*, 77:259–289, 2008. 2

[22] M. Leotta and J. Mundy. Vehicle surveillance with a generic, adaptive, 3d vehicle model. *IEEE PAMI*, 33(7):1457–1469, 06 2011. 2

[23] Q. Lin, Chen, and S. Yan. Automatic 3d model construction for turn-table sequences. In *Proc. ICLR*, 2014. 2

[24] R. J. López-Sastre, T. Tuytelaars, and S. Savarese. Deformable part models revisited: A performance evaluation for object category pose estimation. In *Proceedings of the ICCV Workshops*, 2011. 6, 7

[25] M. Ozuysal, V. Lepetit, and P. Fua. Pose estimation for category specific multiview object localization. In *Proc. CVPR*, 2009. 6, 7, 8

[26] B. Pepik, P. Gehler, M. Stark, and B. Schiele. 3d^{2}pm - 3d deformable part models. In *Proc. ECCV*, 2012. 2, 6, 7

[27] B. Pepik, M. Stark, P. Gehler, T. Ritschel, and B. Schiele. 3d object class detection in the wild. In *CVPR Workshops*. IEEE, 2015. 2

[28] B. Pepik, M. Stark, P. Gehler, and B. Schiele. Teaching 3d geometry to deformable part models. In *Proc. CVPR*, 2012. 5

[29] J. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. 6

[30] M. Prasad, A. Zisserman, and A. W. Fitzgibbon. Single view reconstruction of curved surfaces. In *Proc. CVPR*, volume 2, pages 1345–1354, June 2006. 2

[31] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. In *IJCV*, 2015. 6

[32] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *arXiv preprint*, 2014. 6

[33] A. Thomas, V. Ferrari, B. Leibe, T. Tuytelaars, and L. Van Gool. Shape-from-recognition: Recognition enables meta-data transfer. *CVIU*, 113(12):1222–1234, 2009. 2

[34] Y. Xiang, R. Mottaghi, and S. Savarese. Beyond pascal: A benchmark for 3d object detection in the wild. In *Proc. WACV*, 2014. 6

[35] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *Proc. CVPR*, 2016. 2, 3, 4

[36] M. Z. Zia, M. Stark, and K. Schindler. Towards scene understanding with detailed 3d object representations. *IJCV*, 112(2):188–203, 2015. 2