Reconstructing Vechicles from a Single Image: Shape Priors for Road Scene Understanding

J. Krishna Murthy[1], G.V. Sai Krishna[1], Falak Chhaya[1], and K. Madhava Krishna[1]

***Abstract*— We present an approach for reconstructing vehicles from a single (RGB) image, in the context of autonomous driving. Though the problem appears to be ill-posed, we demonstrate that prior knowledge about how 3D shapes of vehicles project to an image can be used to reason about the reverse process, i.e., how shapes (back-)project from 2D to 3D. We encode this knowledge in *shape priors*, which are learnt over a small keypoint-annotated dataset. We then formulate a shape-aware adjustment problem that uses the learnt shape priors to recover the 3D pose and shape of a query object from an image. For shape representation and inference, we leverage recent successes of Convolutional Neural Networks (CNNs) for the task of object and keypoint localization, and train a novel cascaded fully-convolutional architecture to localize vehicle *keypoints* in images. The shape-aware adjustment then robustly recovers shape (3D locations of the detected keypoints) while simultaneously filling in occluded keypoints. To tackle estimation errors incurred due to erroneously detected keypoints, we use an Iteratively Re-weighted Least Squares (IRLS) scheme for robust optimization, and as a by-product characterize noise models for each predicted keypoint. We evaluate our approach on autonomous driving benchmarks, and present superior results to existing monocular, as well as stereo approaches.**

I. INTRODUCTION

With the advent of autonomous driving, the robot vision community has been devoting siginficant attention to understanding road scenes. Many of the successful approaches to road scene understanding make extensive use of LiDAR or stereo camera rigs. However, there has been increasing interest in replacing these expensive systems with cheap off-the-shelf cameras. This poses many interesting challenges, which are primary motivating factors for the current paper.

We focus on the specific problem of recovering 3D shape and pose of vehicles, given a single (RGB) image. Our approach is based on the premise that humans are able to perceive 3D structure from a single image, owing to their vast prior knowledge on how various 3D shapes project to 2D. Using this prior knowledge, they perform efficient inference of the reverse process, i.e., the 3D structure from 2D appearance. We attempt to endow machines with the same capability, by learning *shape priors*, and using them along with other constraints arising from projective imaging geometry in a robust optimization framework.

Ability to accurately detect objects and their corresponding part locations (*keypoints*) has been shown to benefit pose and shape estimation [1]. To this end, we leverage recent successes of Convolutional Neural Networks (CNNs) in related visual perception tasks [2], [3], [4], and train fully-convolutional regressors for extracting keypoints from a detected object. We show that a small keypoint-annotated training set suffices to train keypoint regressors, and in the subsequent learning of shape priors.

[1]J. Krishna Murthy, G. Sai Krishna, Falak Chhaya, and K. Madhava Krishna are with Robotics Research Center, International Institute of Information Technology, Hyderabad, India. krrish94@gmail.com mkrishna@iiit.ac.in

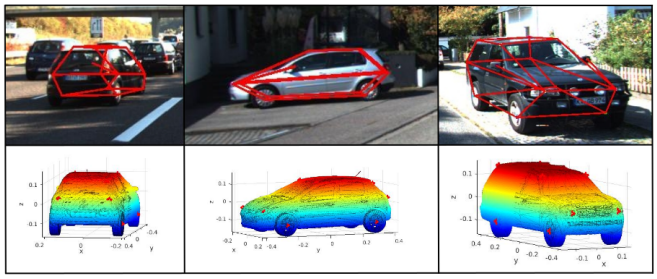

Fig. 1. Examples of 3D shapes estimated by the proposed pipeline for instances with varying shape and pose. Just using a single image, the algorithm extracts the locations of keypoints in 3D. The algorithm also extracts 3D (and subsequent 2D) locations for occluded keypoints. **Top:** Estimated 3D shapes projected down to the image. **Bottom**: Estimated pose and shape shown by rendering the *closest* CAD model to the query instance.

Contributions: The primary contribution of this paper is a novel *shape-aware adjustment* scheme which estimates the 3D pose and shape of a vehicle, given the shape prior and keypoint detections. The proposed formulation works even when some keypoints are not detected (due to occlusion) and robustly fills in the missing keypoints. Moreover, it accounts for imperfect keypoint localization and recovers noise models for each detected keypoint by using an Iteratively Reweighted Least Squares (IRLS) scheme. We quantitatively demonstrate that such a reweighting scheme can applied to standard pose estimation pipelines such as [5] and it boosts accuracy by more than an astounding 90%!

Another contribution of the paper is a novel architecture for keypoint localization which we call a *CNN-cascade*. CNN-cascade performs the task of accurate keypoint localization and aids the 3D pose and shape adjustment. To guide future efforts towards keypoint localization, we release a dataset comprising of keypoint annotations for a subset of vehicles from the KITTI [6] autonomous driving benchmark.

Central Idea: Estimating shape and pose simultaneously from a single image is an ill-posed problem, and suffers from several ambiguities as demonstrated in [7]. Most notably, when the object pose is unconstrained, optimizing on shape reprojection error results in arbitrary deformations in the estimated 3D shape. The central idea of this paper is to *decouple the pose and shape estimation problems*. Our approach can be viewed as a coarse-to-fine adjustment, where we first obtain the coarse shape of the object by aligning the

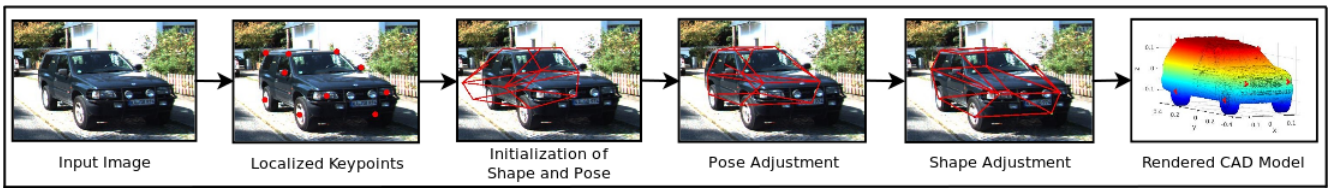

Fig. 2. Illustration of the proposed pipeline. **Left to Right:** The input image is passed through the CNN-cascade to accurately localize keypoints. The initialization is (usually) with an incorrect pose and shape. Pose adjustment aligns the initial wireframe with the actual shape. Shape adjustment optimizes for the 3D shape that best explains the 2D keypoint evidence. Note how the headlights and wheels do not project correctly after pose adjustment, but are refined by shape adjustment. We also render the *closest* CAD model (under the Hausdorff distance metric) to the estimated wireframe.

mean shape for the object category with the query instance. Then, we capture keypoint evidence which is specific to the instance, and run a novel shape-adjustment procedure which ensures that the optimized shape satisfies geometric constraints for the object category (such as planarity, symmetry, etc.), and does not deviate much from the shape prior for the object category. A set of sample outputs from our approach is shown in Fig. 1. The pipeline is summarized in Fig. 2.

Evaluation: We perform an extensive analysis of the proposed approach on the KITTI [6] benchmark for autonomous driving. We evaluate our approach on more than $14,000$ vehicles and demonstrate superior performance with respect to published monocular and stereo leaders by upto 27% in terms of pose accuracy. [1]

II. RELATED WORK

The last decade-and-a-half has witnessed a huge volume of work on high-level perception for autonomous driving. Many successful approaches make use of information from multiple cues, most notably Stereo and LIDAR. In this section we briefly review work pertaining to monocular perception, while discussing the differences in the underlying assumptions/approach.

Philosophically, the closest approach to ours is the one by Zia, et al. [1], where 3D shape representations for vehicles were learnt from annotated CAD models. Vehicle shape was represented as an ordered collection of 3D part locations and random forest classifiers were employed for localizing 2D parts given an object bounding box. The approach was extended to handle information from multiple views in [8]. However, the shape inference mechanism used in [1], [8] was stochastic hill climbing, owing to the highly non-convex and non-smooth cost function, thereby resulting in a very slow, iterative optimization procedure. Moreover, part localization was performed using random forests, which was slow too. In contrast, our optimization formulation comprises of smooth functions and can be solved using non-linear least squares, and our part localization pipeline is fast.

In a sequence of works, [9], [10] have developed a real-time monocular SfM system for autonomous driving. However, a vehicle is represented as a 3D bounding box; no shape information is involved. Further, information from multiple frames is used. Our approach attempts to use information from only a single image, as opposed to a video sequence.

[1]The code and dataset used in this work will be made publicly available.

In [11], Non-Rigid SfM [12] is used to learn shape representations over a keypoint-annotated dataset. Using the learnt *average shape* representations and *deformation modes*, dense point clouds of vehicles are reconstructed from a single image. We differ from [11] in that we capture inherent geometric constraints that vehicles exhibit (symmetry and planarity, for instance) which would result in more meaningful shape estimates. This was partially addressed in [7], which tries to use symmetry and Manhattan properties of objects to recover more meaningful shapes. However, the reconstructions assume a weak-orthographic projection model, whereas we use a projective camera model and a globally optimal pose estimation pipeline.

All the above methods rely on some form of object part detection to produce meaningful reconstructions. For instance, Zia, et. al. [1] rely on Deformable Part Models (DPMs) and random forests. With the promise demonstrated by Convolutional Neural Networks (CNNs) in object detection tasks, they have also been used for the task of viewpoint and keypoint prediction in [2]. Apart from [2], convolutional architectures for keypoint localization have been proposed in [3], [4]. Motivated by the promise shown by CNNs, we train a cascaded architecture of fully-convolutional networks for keypoint localization.

Current published monocular and stereo competitors for 3D object detection are [13] and [14] respectively. Both these approaches operate at the level of estimating 3D oriented bounding boxes for car detection. We comprehensively analyze the performance of the proposed approach and demonstrate a significant performance boost over [13], [14]. Note that we also learn a more detailed 3D representation than 3D bounding boxes, which is desirable for reconstruction.

III. OUR APPROACH

Simultaneous pose and shape estimation of vehicles is ill-posed when only a single image is available [7]. Guided by the motivation that humans make use of prior information about the vehicle to reason about 3D shape and pose, we decouple the pose and shape estimation problems. Fig. 2 illustrates the overall picture of the proposed pipeline.

A. Shape Priors

Our approach uses *shape priors* to model the 3D representation of an object category. More specifically, a shape prior is an ordered collection of 3D vertices (*keypoints*, or *parts*) for the *average shape* of an instance from the object category. By *keypoints*, we refer to various semantic parts of an object

category that are common to all instances of that category. For example, in the case of cars, potential candidates for keypoints are wheel centers, headlights, rooftop corners, etc. Previous approaches [1], [8] made use of a large database of keypoint-annotated CAD models to define shape priors. On the other hand, we make use of a small 2D keypoint-annotated image set consisting of nearly 300 instances from the PASCAL3D+ [15] dataset.

Learning Shape Priors from 2D Data: Given a keypoint-annotated dataset, the conventional approach to reconstructing 3D keypoints is to run a Structure-from-Motion (SfM) pipeline using keypoints as correspondences. However, this is not a suitable approach since keypoint correspondences give a valid reconstruction only when the *same rigid* instance is observed across multiple images. We would like to have as many *different* instances as possible, to capture a large part of the intra-class shape variations. But, when we have different instances, it is no longer possible to use keypoints as correspondeces. To overcome this difficulty, we use the EM-PPCA method of [11], [12] and cast the problem of *lifting the dataset from 2D to 3D* into the framework of Non-Rigid Structure-from-Motion (NRSfM). Consider that we are given M instances each annotated with K keypoints (2D shapes), where $\mathbf{s}_{i,m}$ ($i \in \{1..K\}, m \in \{1..M\}$) represents the i^{th} keypoint of instance m. As in [11], we assume that the projection model is weakly-orthographic (note that this assumption is made only until we learn shape priors; we relax this soon). We denote the 3D shape of an instance m by $\mathbf{S}_m$, i.e., $\mathbf{S}_m = \left[\mathbf{s}_{1,m}^T, ..., \mathbf{s}_{K,m}^T\right]^T$ and the (orthographic 2×3) rotation and translation of the camera for the instance by $\mathbf{R}_m$ and $\mathbf{t}_m$. NRSfM hypothesizes that real-world shapes are confined to a low-dimensional *basis* of the entire shape space. We express a specific shape instance as the sum of the mean shape of the object category $\bar{\mathbf{S}}$ deformed using a linear combination of vectors from a set of N *basis shapes* $\mathbf{V} = [\mathbf{V}_1, ..., \mathbf{V}_N]$. Then, the NRSfM problem [12] involves maximizing the likelihood of the following model.

$$\begin{aligned} \mathbf{s}_{i,m} &= c_m \mathbf{R}_m (\mathbf{S}_{i,m} + \mathbf{t}_m) + \delta_{i,m} \\ \mathbf{S}_m &= \bar{\mathbf{S}} + \sum_{j=1}^{N} \lambda_{j,m} \mathbf{V}_j \\ \delta_{i,m} &\sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}) \quad \lambda_{j,m} \sim \mathcal{N}(0,1) \quad \mathbf{R}_i^T \mathbf{R}_i = \mathbf{I} \end{aligned} \tag{1}$$

In the above model, $\lambda_{j,m}$ are the scalar weights (latent variables) of the shape combination, and $\delta_{i,f}$ models observation noise. c_m is the weak-orthographic scaling factor.

Note that all instances would have some missing keypoints, due to occlusion. Hence, the likelihood of the model is maximized using the EM algorithm. Missing data is filled in the E-step using the method of [12]. We then have a mean shape and basis vectors that characterize various deformation modes of the mean shape. For the case of a car, choosing the first five basis vectors captured more than 99.99% of the shape variations.

B. Pose Adjustment

For correctly estimating the 3D pose of an object, we propose a variant of Perspective n-Point (PnP) [5] to obtain a robust, globally optimal solution. The pose adjustment pipeline operates in two phases, viz. keypoint localization and robust alignment.

Keypoint localization: The approach relies on localization of the keypoints for the object class in the image. Leveraging the recent successes of Convolutional Neural Networks for the task of keypoint localization [2], [3], [4], we train a fully convolutional regressor to accurately localize keypoints given a detected object instance as input. Rather than relying entirely on monolithic networks that operate at the instance level as in [2], [4], we propose augmenting such networks with a cascaded architecture consisting of small, keypoint-specific subnetworks, following the broad idea of [3]. Specifically, we use the keypoints detected by [2] (denoted KNet) and further refine them using keypoint-specific finetuning networks to reduce mis-predictions and to obtain a more precise localization. Around every keypoint location output by KNet, we sample N random patches of size 32×32, which is input to the finetuning network for the corresponding keypoint. The finetuning networks are trained in such a way that they output a non-parametric probability distribution of keypoint likelihoods over the input patch, rather than regressing to a vector denoting keypoint coordinates, as in [3]. We aggregate and normalize keypoint likelihoods over the N patches, and choose the location of the maxima of the normalized keypoint likelihood map to be the estimated location of the keypoint. The architecture of our network is shown in Fig. 3.

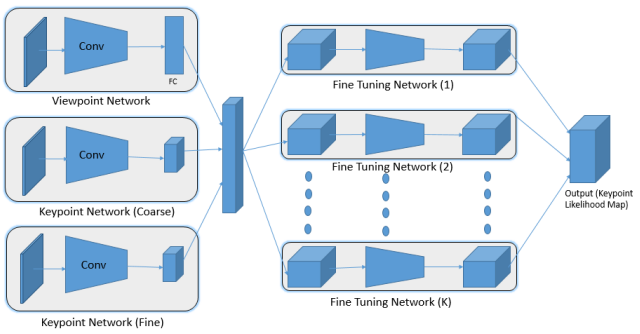

Fig. 3. Architecture of our Keypoint Localization Network

Robust Alignment: Given keypoint locations from the CNN-cascade in the earlier stage, we now formulate a robust PnP problem and solve it to obtain a globally optimal solution using a Gröebner basis solver. The usual PnP problem for estimating the camera extrinsics $(\mathbf{R}, \mathbf{t})$, given a set of n 3D points $\mathbf{X}_i$ and their 2D correspondences $\mathbf{x}_i$ can be stated as the following optimization problem (Here, $\mathbf{K}$ is the *projective* camera intrinsic matrix).

$$\begin{aligned} \min_{\mathbf{R},\mathbf{t}} \sum_{i=1}^{n} \|\mathbf{K}(\mathbf{R}\mathbf{X}_i + \mathbf{t}) - \mathbf{x}_i\|^2 \\ \text{s.t.} \quad \mathbf{R}^T\mathbf{R} = \mathbf{I} \end{aligned} \tag{2}$$

It is hard to achieve a global optima for the above problem, owing to the rotation matrix constraint. However, [5] formulates a polynomial cost function by parametrizing the rotation matrix as a non-unit quaternion and uses a Gröebner basis solver to obtain a global optimum. It can briefly be summarized as

$$\min_{a,b,c,d} \|\mathbf{M}\alpha\|^2 \tag{3}$$

where α is the Gröebner basis containing the parameters of the quaternion, namely a, b, c, d and the translation, and M is the matrix that holds the coefficients of the Gröebner basis.

This method has one major drawback in that it is intolerant to keypoint localization errors. We extend the formulation in [5] to provide a more robust solution in presence of keypoint localization errors. Specifically, we use the method of Iteratively Re-weighted Least Squares (IRLS) to weigh each observation (keypoint) in the cost function and solve it iteratively, updating the weights of each observation. To assign weights to a particular observation, we take into account two confidence measures. First, we consider the keypoint likelihood as output from the CNN-cascade w_{cnn}. Second, we consider the prior probability of the keypoint being visible from the pose estimate at time t $w_{vis}(t)$, computed by ray-tracing. The weight assigned to the i^{th} keypoint at time t is

$$w_i(t) = \mu_0 w_{cnn} + (1 - \mu_0) w_{vis}(t) \tag{4}$$

In each iteration, we solve the PnP problem to obtain estimates for $(\mathbf{R}, \mathbf{t})$. We update the weights for each observation in the following manner.

$$w_i(t+1) = \mu_1 w_i(t) + (1 - \mu_1) \left[\mu_2 e_i(t) + (1 - \mu_2) w_{vis}(t)\right] \tag{5}$$

Here, $e_i(t)$ is the reprojection error in the i^{th} keypoint, normalized over all instances. μ_0, μ_1 and μ_2 are hyperparameters that determine the importance of each term. The final optimization problem can be written in the form shown in Eq. 6, where $\mathbf{W}$ is a diagonal matrix of weights. Algorithm 1 presents an overview of the process.

$$\min_{a,b,c,d} \|\mathbf{M}\mathbf{W}\alpha\|^2 \tag{6}$$

Algorithm 1 ASPnP + IRLS

```
Initialize weights as in Eq. 4
Initialze (R, t) by solving the PnP problem in Eq. 6
t ← 0
while t ≤ MAX_ITERS (usually set to 5) do
    Compute reprojection error err_i(t) for each keypoint
    Normalize all reprojection errors
    Update weights as in Eq. 5
    Solve for (R, t) by solving Eq. 6
    t ← t + 1
end while
```

Note that this variant of IRLS converges in just 4-5 iterations in all cases. We use the final set of weights obtained from IRLS as an indicator of how confident (or noisy) the keypoint is.

C. Shape-Aware Adjustment

Using the pose estimated from the previous stage as an initialization, we now formulate a shape-aware adjustment problem that captures—in addition to standard reprojection error—constraints generated from planarity and symmetry in the object category.

Recall that in Eq. 1 we defined the shape of an instance to be expressible in terms of the sum of the mean shape and a linear combination of the basis vectors. If $\mathbf{X}_i$ denotes the i^{th} 3D keypoint location, it can be expressed in terms of the mean location for $\mathbf{X}_i$ (denoted $\bar{\mathbf{X}}_i$), and its deformation basis and coefficients $\mathbf{V}_i = [\mathbf{V}_{i,1}, ..., \mathbf{V}_{i,N}]$ and λ_j respectively. Note that the deformation basis $\mathbf{V}_i$ is specific to each keypoint (obtained by extracting specific rows from the $\mathbf{V}$ matrix resulting from NRSfM), but the coefficients λ_j are common to all keypoints. If this was not the case, the resultant deformations of varying λ_j would be unconstrained.

$$\mathbf{X}_i = \bar{\mathbf{X}}_i + \sum_{j=1}^{N} \lambda_{i,j} \mathbf{V}_{i,j} \tag{7}$$

Reprojection Error: We define the reprojection error of a set of points formally as

$$E_{reproj} = \sum_{i=1}^{K} \|\mathbf{K}\mathbf{X}_i - \mathbf{x}_i\|^2 \tag{8}$$

Planarization: We exploit the fact that each vehicle consists of a set of planar (or quasi-planar) surfaces. For instance, the centers of the wheels of a car are planar; so are the corners of the rooftop in most cases. We consider the shape $\mathbf{S}$ to be a quad-mesh consisting of a few planar faces $\mathbf{F}$. For each face $f \in \mathbf{F}$, we define four variables $(\mathbf{n}_f, d_f)$, where $\mathbf{n}_f$ is a 3-vector representing the normal of the plane, and d_f is a scalar that represents the distance of the plane from the origin. We solve for these variables by defining a planarization energy as

$$E_{planar} = \sum_{f \in \mathbf{F}} \left(\sum_{v \in f} \|\mathbf{n}_f^T \mathbf{X}_v + d_f\|^2 \right) - \mu_f \left(1 - \mathbf{n}_f^T \mathbf{n}_f\right) \tag{9}$$

The first term encourages all four points on the face (four points, since we assume a quad mesh) to be planar, while the second term encodes the constraint that the normals must be of unit magnitude. Wherever applicable, we also constrain the normals so that they are parallel to the ground plane normal. Furthermore, it is not enough if certain points are planar; they must also be rectangular. These requirements are imposed as further constraints.

Symmetrization: All real-world cars exhibit symmetry about their medial plane. However, we find this fact rarely exploited for monocular reconstruction. Symmetry constraints are imposed for reconstruction in [7], but they propose a factorization-based solution for the weak-orthographic

projection case. On the other hand, we impose symmetry constraints for the projective case and solve it using non-linear least squares. We consider two sets of corresponding symmetric points - $\mathbf{X}_r \in \mathbf{R}$ to the right of the medial plane, and correspondingly $\mathbf{X}_l \in \mathbf{L}$ to the left of the medial plane. If $\mathbf{T} = (\mathbf{n}_{med}, d_{med})$ is the medial plane, we define the symmetrization energy as prescribed in [16].

$$E_{sym} = \sum_{\mathbf{X}_r \in \mathbf{R}} \|\mathbf{X}_r + 2(d_{med} - \mathbf{n}_{med}^T \mathbf{X}_r)\mathbf{n}_{med} - \mathbf{X}_l\|^2 \quad (10)$$

Regularization: To ensure that the energies proposed herewith result in realistic output values, we define a set of regularization terms. We encourage the length $\mathcal{L}(\mathbf{S})$, width $\mathcal{W}(\mathbf{S})$, and height $\mathcal{H}(\mathbf{S})$ of each shape $\mathbf{S}$ to be *close* to their prior values, computed over the dataset. If $\bar{\mathbf{l}}$, $\bar{\mathbf{w}}$, and $\bar{\mathbf{h}}$ are the prior lengths, widths, and heights respectively, then the regularizers for dimensions are

$$E_{dim} = \mu_l \|\mathcal{L}(\mathbf{S}) - \bar{\mathbf{l}}\|^2 + \mu_w \|\mathcal{W}(\mathbf{S}) - \bar{\mathbf{w}}\|^2 + \mu_h \|\mathcal{H}(\mathbf{S}) - \bar{\mathbf{h}}\|^2 \quad (11)$$

Further, we use a modification of the well-known Laplacian smoothing regularizer which encourages each vertex to remain close to the centroid of its neighbors. We have found it to perform slightly better than the usual Euclidean distance regularizer.

$$E_{lap} = \sum_{i=1}^{K} \|\mathbf{X}_i - \left(\sum_{j \in \text{Nbd}(\mathbf{X_i})} e^{-\|\mathbf{X}_i - \mathbf{X}_j)\|^2} \mathbf{X}_j \right) \|^2 \quad (12)$$

Initialization: To initialize our pose adjustment pipeline, we use 3D bounding boxes from [13]. We rotate and translate the mean shape to the center of the estimated bounding box, and scale the wireframe such that it fits the box. Although such an initialization is not mandatory (the ASPnP + IRLS is fairly robust to initialization errors (see Fig. 2)), we find that this results in a slight performance gain.

Shape-Aware Adjustment: Our final shape-aware adjustment is the conglomeration of all the above terms (with appropriate weighing coefficients), and can be written as

$$E_{total} = \eta_1 E_{reproj} + \eta_2 E_{planar} + \eta_3 E_{sym} + \eta_4 E_{dim} + \eta_5 E_{lap} \quad (13)$$

All terms in the equations are easily differentiable (unlike those in [1], [8]), and can be solved using a standard non-linear least squares solver. Moreover, since we are solving for the shape of one instance, the number of variables involved is small, which results in near real-time performance. Again, we emphasize that, we do not solve for the locations of the 3D points $\mathbf{X}_i$; we instead solve for the weights $\lambda_{j,m}$ of the basis vector combination (see Eq. 1). We apply 5 iterations of IRLS weight updates, to reduce the effect of incorrect observations on the estimated shape.

IV. RESULTS

We perform a thorough qualitative and quantitative analysis of the proposed approach on the *Car* class of the challenging KITTI object detection and pose estimation benchmark [6]. Since we decouple the problem of shape optimization from that of pose estimation, we evaluate each of them independently. To indicate that this evaluation is fair, we also conduct experiments to verify that the shape optimization does not result in significant changes in object pose. We also present an analysis of per-keypoint localization errors. Finally, we show qualitative results (Fig. 4) which indicate that the proposed approach works over a wide range of vehicle shapes.

Dataset: The dataset consists of 7481 training images, with specified train and validation splits. Based on the levels of occlusion and truncation, the evaluation has been split into indicated in the *Easy*, *Moderate*, and *Hard* regimes. We train the CNN-cascade by annotating data from the train split. We evaluate our approach against other competitors on the validation split.

To evaluate the pose estimated by the proposed approach, we use all the 3424 images (14327 instances) from the validation split. However, for the evaluation of part detectors, we manually annotate 500 images from the validation split, and evaluate our algorithm on it, akin to [1].

Keypoint-annotations for KITTI: Keypoint localization is increasingly being recognized as a tool to develop finer understanding of objects. To guide further research in this direction, especially in the context of autonomous driving, we annotate keypoints for cars in a subset of the KITTI object dataset. To ensure consistency of the annotated keypoints with their 3D locations, we follow the procedure adopted in creating the PASCAL3D+ dataset [15]. Specifically, we use the 2D keypoint coordinates and initialize a PnP (Perspective n-Point) algorithm that uses reference CAD models to correct incorrectly marked 2D keypoints. We intend to make the annotations publicly available.

Metrics: To evaluate the pose estimated by our approach, we use the Average Orientation Precision (AOP) metric proposed in [15], as well as the mean absolute difference in orientation estimates. For our AOP criterion, the detection output is considered correct if and only if the overlap ratio with the ground-truth bounding box is greater than 70% and the difference between the predicted and ground-truth viewpoints is less than a particular threshold. We characterize the performance of various approaches by evaluating them for various values of the threshold.

To evaluate error in estimated keypoint locations, we use the APK (Average Precision of Keypoints) metric introduced in [17]. Under this metric, a keypoint is assumed to be accurately localized if it lies within $\alpha * \max(h, w)$, where h, w are the height and width of the bounding box respectively, and we set α to 0.1, following [2].

To evaluate quantitatively the distance between two shapes (meshes), we use the Hausdorff distance metric.

Implementation: The CNNs used for keypoint detection were based on the fully convolutional architecture presented in [2], with the only change that the loss function was defined in terms of a non-parametric Gaussian over the true keypoint location. The networks were trained in Caffe [18], over annotated data from the train split. The provided train/validation split ensured that there was no overlap be-

Approach	**Pose Estimation Accuracy** in terms of Average Orientation Precision (**AOP**)								
	Easy			*Moderate*			*Hard*		
	$\leq 5^\circ$	$\leq 15^\circ$	$\leq 30^\circ$	$\leq 5^\circ$	$\leq 15^\circ$	$\leq 30^\circ$	$\leq 5^\circ$	$\leq 15^\circ$	$\leq 30^\circ$
ObjProp3D [14]	40.04	81.37	91.83	36.76	77.41	89.31	33.98	73.22	85.81
Mono3D [13]	42.20	84.59	92.90	38.52	80.60	90.08	35.78	76.27	86.67
ASPnP [5]	1.06	6.18	27.26	1.59	6.35	25.40	2.87	6.90	25.29
Pose Adjustment (Ours)	**53.62**	**90.44**	**95.95**	**48.50**	**85.67**	**94.44**	**44.72**	**80.98**	**89.08**
Percentage Improvement	$+\mathbf{27.06}\%$	$+\mathbf{6.91}$	$+\mathbf{3.28}\%$	$+\mathbf{25.91}\%$	$+\mathbf{5.41}\%$	$+\mathbf{9.92}\%$	$+\mathbf{14.34}\%$	$+\mathbf{6.17}\%$	$+\mathbf{3.77}\%$

TABLE I

ORIENTATION PRECISION EVALUATION FOR VARIOUS ERROR RANGES ON THE KITTI OBJECT DETECTION BENCHMARK. THE PERCENTAGE IMPROVEMENT ROW IS COMPUTED WITH RESPECT THE NEXT BEST PERFORMER IN EACH CATEGORY (COLUMN).

Approach	**APK**			**Hausdorff Distance from Mean Shape**		
	Easy	Moderate	Hard	Easy	Moderate	Hard
Initialization	47.31	51.65	43.17	0.00	0.00	0.00
Pose Adjustment	62.42	62.72	59.89	3.71	4.64	4.79
Shape Adjustment	**80.72**	**81.81**	**71.39**	3.75	4.69	4.83
Between Pose and Shape Adjustment				**0.27**	**0.28**	**0.04**

TABLE II

EFFECT OF EACH STEP IN THE PROPOSED PIPELINE ON SHAPE ESTIMATION

tween images/sequences.

The shape-aware adjustment was implemented using Ceres Solver [19], and was solved using a dense Schur linear system solver and a Jacobi preconditioner.

A. Pose Estimation

To analyze the efficiency of our pose estimation approach , we evaluate it against Mono3D [13], which is the current monocular competitor for 3D object detection and pose estimation. We also show that, due to inaccuracies in keypoint estimates, the standard ASPnP [5] formulation struggles to find a suitable rotation and translation for the shape prior, whereas the proposed approach (ASPnP + IRLS) achieves precise pose (to within 5 degrees) for more than 50% of the samples. Table I compares our results with that of Mono3D [13] and ObjProp3D [14]. Note that ObjProp3D is a stereo-based approach and the comparision is thus unfair. On an average, we improve the pose estimation performance with respect to the next-best competitor (for each evaluation category) by 11.42%.

We see that, in most cases, we are within $\pm 30^\circ$ of the original azimuth, while more than half of the time, we are within $\pm 5^\circ$ of the actual azimuth. In Table III, we show the mean absolute difference (error) in the pose estimates. Here, we see the robustness that IRLS renders to the pose adjustment pipeline. ASPnP [5]—though claims global optimiality of its solution—struggles in the presence of noise. It turns out that in all instances of cars, one or more keypoints is self-occluded and hence filled-in only approximately by the keypoint localization module. In our formulation of the IRLS weight update, weights assigned to incorrectly estimated keypoints get reduced every iteration, and have less effect on the final pose estimate.

B. Shape-Aware Adjustment

We now demonstrate the performance of the shape adjuster by quantitifying the precision in part locations before and after shape optimization. Note that reprojection error alone is not a measure of 3D shape correctness, as we could always have an arbitrary shape that could lead to zero reprojection error. To demonstrate that the optimized wireframe is *close* to the shape space of cars, we evaluate a mesh error metric based on the Hausdorff distance of the optimized shape from the shape prior. Further, following [1], we show qualitative results for 3D shape estimation (Fig. 4). Table II shows the perfomance of our shape adjuster on subsets of the easy, moderate, and hard splits. We sampled every 50th car in the easy split, every 75th car in the moderate split, and every 100th car in the hard split (this was done to get a nearly equal number of images in each resultant set). As shown in Table II, there is a significant change between the shape prior (initialization) and the output from the pose adjustment module. This is due to the fact that the initialization is erroneously oriented most of the time, and the pose adjustment results in significant rotations being applied to it. Table II validates an important claim of this paper, viz. *pose and shape estimation problems can be decoupled*. This claim is supported by the fact that shape adjustment results in small, local changes to the shape model, in comparison with pose adjustment, which results in (possibly) large, global transformations. Due to the scale of the solutions to the pose estimation and the shape estimation problems, it is best that

Approach	**Mean Absolute Error** (in degrees)		
	Easy	Moderate	Hard
ObjProp3D [14]	17.32	21.86	26.87
Mono3D [13]	15.14	15.15	17.82
ASPnP [5]	98.17	98.82	98.83
Pose Adjustment (Ours)	**8.79**	**12.57**	**16.19**

TABLE III

MEAN ABSOLUTE ERROR IN POSE ESTIMATION (DEVIATION FROM GROUND-TRUTH POSE, IN DEGREES)

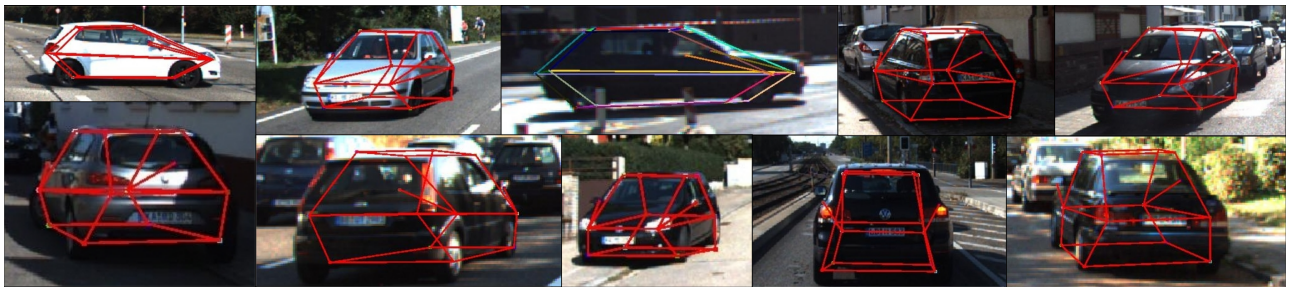

Fig. 4. Qualitative results showing shape reconstructions from our pipeline.

they are solved sequentially.

Fig. 5 provides an analysis of APK for each keypoint. This plot strongly justifies the need for shape adjustment. We can observe that, by pose adjustment alone, we improve on APK, but this improvement is strictly *on an average*, i.e., APK averaged over all parts improves, but there are parts (eg. parts 2, 3) where the initialization has a lower APK. After shape adjustment, however, the APK improves significantly for all parts. Note that part 2 (which corresponds to the center of the right front wheel) is one of the least frequently occurring parts in the KITTI object dataset.

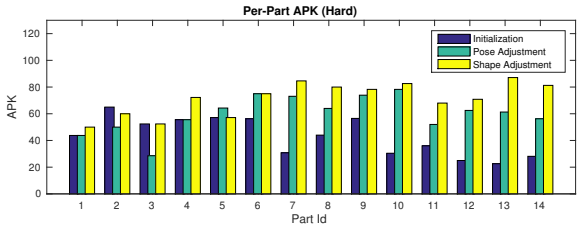

Fig. 5. Per-Part APK Measure for the *Hard* split. Parts 1-4 correspond to the wheels, 5-6 correspond to the headlights, 7-8 correspond to the taillights, 9-10 correspond to the side-view mirrors, and 11-14 correspond to four corners of the rooftop.

C. *Run-Time Analysis*

All the pipelines we proposed have been designed with the goal of being applicable in autonomous driving scenarios. Barring the CNN-cascade, all other modules run on a single CPU core. The CNN-cascade runs on an NVIDIA GeForce GTX. Details of running times of various components of the pipeline are furnished in Table IV.

Module	Runs on	Runtime per instance (in ms)
Pose Adjustment	CPU	9.97 ms
CNN-cascade	GPU	200.79 ms
Shape Adjustment	CPU	52 ms

TABLE IV

RUNNING TIMES OF VARIOUS COMPONENTS OF THE PIPELINE

V. CONCLUSIONS

In this work, we presented an approach for estimating the 3D shape and pose of a vehicle from a single image. We decoupled the pose estimation problem from that of shape estimation, and proposed fast, robust algorithms for each of them. We evaluated our approach against published monocular and stereo competitors and demonstrated superior performance, thereby advancing the state-of-the-art. In retrospect, we feel that the bottleneck to achieving near-perfect performance is imprecise keypoint localization, and we identify that as a future research direction. We would also like to explore the benefits of temporal information, wherever available.

ACKNOWLEDGMENT

The authors would like to thank the following people for spending some of their valuable time annotating keypoints on cars from the KITTI object detection benchmark: Abhishek Raj, Sheetal Reddy, Falak Chhaya, Sarthak Sharma, Abhineet Jain, Parv Parkhiya, and Akanksha Baranwal.

REFERENCES

[1] M. Z. Zia, M. Stark, and K. Schindler, "Towards scene understanding with detailed 3d object representations," *International Journal of Computer Vision*, vol. 112, no. 2, pp. 188–203, 2015.

[2] S. Tulsiani and J. Malik, "Viewpoints and keypoints," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2015, pp. 1510–1519.

[3] Y. Sun, X. Wang, and X. Tang, "Deep convolutional network cascade for facial point detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 3476–3483.

[4] T. Pfister, J. Charles, and A. Zisserman, "Flowing convnets for human pose estimation in videos," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1913–1921.

[5] Y. Zheng, Y. Kuang, S. Sugimoto, K. Astrom, and M. Okutomi, "Revisiting the pnp problem: A fast, general and optimal solution," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 2344–2351.

[6] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[7] Y. Gao and A. L. Yuille, "Exploiting symmetry and/or manhattan properties for 3d object structure estimation from single and multiple images," *arXiv preprint arXiv:1607.07129*, 2016.

[8] F. Chhaya, D. Reddy, S. Upadhyay, V. Chari, M. Z. Zia, and K. M. Krishna, "Monocular reconstruction of vehicles: Combining slam with shape priors," in *Proceedings of the IEEE Conference on Robotics and Automation*, 2016.

[9] S. Song and M. Chandraker, "Robust scale estimation in real-time monocular sfm for autonomous driving," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1566–1573.

[10] S. Song and M. Chandraker, "Joint sfm and detection cues for monocular 3d localization in road scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3734–3742.

[11] S. Tulsiani, A. Kar, J. Carreira, and J. Malik, "Learning category-specific deformable 3d models for object reconstruction." *IEEE transactions on pattern analysis and machine intelligence*, 2016.

[12] L. Torresani, A. Hertzmann, and C. Bregler, "Nonrigid structure-from-motion: Estimating shape and motion with hierarchical priors," *IEEE transactions on pattern analysis and machine intelligence*, vol. 30, no. 5, pp. 878–892, 2008.

[13] X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun, "Monocular 3d object detection for autonomous driving," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2147–2156.

[14] X. Chen, K. Kundu, Y. Zhu, A. G. Berneshawi, H. Ma, S. Fidler, and R. Urtasun, "3d object proposals for accurate object class detection," in *Advances in Neural Information Processing Systems*, 2015, pp. 424–432.

[15] Y. Xiang, R. Mottaghi, and S. Savarese, "Beyond pascal: A benchmark for 3d object detection in the wild," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2014.

[16] N. J. Mitra, L. J. Guibas, and M. Pauly, "Symmetrization," *ACM Transactions on Graphics (TOG)*, vol. 26, no. 3, p. 63, 2007.

[17] Y. Yang and D. Ramanan, "Articulated human detection with flexible mixtures of parts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 12, pp. 2878–2890, 2013.

[18] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.

[19] S. Agarwal, K. Mierle, and Others, "Ceres solver," http://ceres-solver.org.