# Monocular Multiview Object Tracking with 3D Aspect Parts

Yu Xiang[1,2,⋆], Changkyu Song[2,⋆], Roozbeh Mottaghi[1], and Silvio Savarese[1]

[1] Computer Science Department, Stanford University, Palo Alto, USA
{yuxiang,roozbeh}@cs.stanford.edu, ssilvio@stanford.edu
[2] Department of EECS, University of Michigan at Ann Arbor, Ann Arbor, USA
changkyu@umich.edu

**Abstract.** In this work, we focus on the problem of tracking objects under significant viewpoint variations, which poses a big challenge to traditional object tracking methods. We propose a novel method to track an object and estimate its continuous pose and part locations under severe viewpoint change. In order to handle the change in topological appearance introduced by viewpoint transformations, we represent objects with 3D aspect parts and model the relationship between viewpoint and 3D aspect parts in a part-based particle filtering framework. Moreover, we show that instance-level online-learned part appearance can be incorporated into our model, which makes it more robust in difficult scenarios with occlusions. Experiments are conducted on a new dataset of challenging YouTube videos and a subset of the KITTI dataset [14] that include significant viewpoint variations, as well as a standard sequence for car tracking. We demonstrate that our method is able to track the 3D aspect parts and the viewpoint of objects accurately despite significant changes in viewpoint.

**Keywords:** multiview object tracking, 3D aspect part representation.

## 1 Introduction

Traditional object tracking methods focus on accurately identifying the 2D location of objects in the image and associating those locations across frames. While this capability is a critical ingredient in many application scenarios, it is often not sufficient. There are numerous situations (e.g., in autonomous driving) where not only does one need to track the location of an object (e.g., a car) but also infer its 3D pose in time – for instance, if one needs to predict a potential collision, estimating other cars' pose and angular velocities is crucial. Moreover, there are situations (e.g., in robotics or augmented reality) where one needs to identify portions of the object such as its aspects or affordance. For instance, this is critical when an autonomous agent needs to interact with, say, a car and wants to figure out where a door or a window is.

⋆ Indicates equal contribution.

D. Fleet et al. (Eds.): ECCV 2014, Part VI, LNCS 8694, pp. 220–235, 2014.

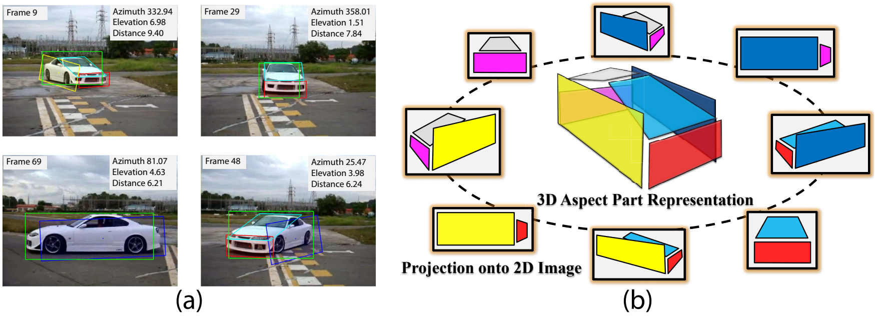

**Fig. 1.** (a) An example output of our tracking framework. Our multiview tracker provides the estimates for continuous pose and 3D aspect parts of the object. (b) An example of the 3D aspect part representation of a 3D object (car) and the projections of the object from different viewpoints.

Unfortunately, most of the existing tracking methods are not capable of (or at least not designed for) estimating the 3D object pose nor tracking portions of the target. In this paper, we seek to address this limitation and propose a new tracking framework that not only tracks the object in 2D, as most the state-of-the-art methods do, but also returns, as part of a joint inference problem, a continuous estimation of the viewpoint in time. Moreover, it is also able to identify and track portions of the object such as its aspects, in time (see Fig. 1(a)).

Our proposed tracker follows and generalizes the philosophy of "tracking by detection" (whereby a track is inferred by using detection hypotheses as observations) and leverages existing 3D (multiview) object representations [39,36,25,37,43,31,13,26] for detecting and estimating the 3D pose of object categories. Unlike traditional tracking by detection methods, however, that just focus on tracking the 2D or 3D location of the object, our approach also "tracks" the 3D pose and parts of the target. We leverage the 3D aspect part representation (see Fig. 1(b)) and use it in a novel particle filtering framework for multiview tracking, where combining viewpoint estimation and the 3D aspect parts enables us to predict the visibility and shape of each 3D aspect part. In particular, we leverage two state-of-the-art object detectors to train the category-level part templates in our multiview tracking framework: Deformable Part Model (DPM) [12] and Aspect Layout Model (ALM) [43]. We believe these are reasonable choices in that: i) DPM achieves state-of-the-art object detection performance and it is suitable for "tracking by detection" implementation as shown in [7,33] ii) ALM achieves state-of-the-art pose estimation results and provides a good platform for injecting 3D information to the 3D pose tracking problem; iii) ALM can recover the object layout in term of the distribution of object aspects in 3D.

Moreover, in order to increase the robustness of our tracker to viewpoint changes as well as occlusions, we propose to inject to our tracker the ability to learn the appearance of the object in an online learning fashion, similar to [2,15,20,3,38,45]. Unlike traditional online learning tracking methods, however, which focus on learning a holistic description of the entire object as the tracking goes by (an exception is the recent work by [45]), we propose to update the appearance model only for the *visible* parts of the object. Part visibility is readily available as a result of the fact that we also estimate the 3D pose of the object in time. A key strength of our approach is that we combine tracking by detection and online learning in a coherent probabilistic framework.

In our experiments, we provide results for viewpoint estimation and 3D aspect part localization. Besides, to demonstrate the usefulness of 3D pose and viewpoint during tracking, we compare our method with some of the state-of-the-art online learning methods that do not use 3D information and show significant improvement. Furthermore, we illustrate that our framework is effective in leveraging temporal information to provide continuous estimates for the object pose with and without online learning. Finally, we show that in the presence of occlusions, online learning helps increase the robustness and accuracy.

Since the current benchmark datasets for online object tracking [41] are not designed to test the ability of the trackers on handling topological appearance changes and do not show significant viewpoint variations, we collected a new challenging dataset with 9 multiview car video sequences from YouTube for experiments. We also test our method on a subset of the KITTI dataset [14] which comprises videos with significant viewpoint changes. Furthermore, we evaluate our method on a standard sequence for car tracking without viewpoint variations [20]. We demonstrate the ability of our method to accurately track viewpoints and 3D aspect parts in videos. Fig. 1(a) shows the tracking results of our method.

**Contributions.** 1) We propose a multiview tracker to handle the topological appearance change of rigid objects during tracking, which estimates continuous 3D viewpoint in a monocular setting. 2) Our multiview tracker is able to track the 3D aspect parts of an object. 3) We combine category-level pre-trained 3D object detectors and instance-level online-learned part appearance models in a principled way. 4) We contribute a new dataset with 9 car video sequences for multiview object tracking, and show promising tracking results on it.

## 2 Related Work

**Tracking by Detection.** Our approach falls in the category of tracking by detection methods [4,5,7,33,44], where category-level detectors are utilized to track the target of interest. However, in contrast to these methods, our focus is on tracking continuous 3D pose and 3D aspect parts.

**Online Object Tracking.** Online trackers focus on constructing appearance models which adapt to appearance changes during tracking [2,15,20,3,45,38]. By leveraging online learning techniques, such as online multiple instance learning [2], online structural learning [45] and self-paced learning [38], these methods have achieved robust tracking results on benchmark datasets [41]. Since they are able to track generic objects, they are referred to as model-free trackers. However, as shown in our experiments, they cannot handle the topological appearance change of objects caused by severe viewpoint transformations. An exception is the recent work by [29] which extends the Lucas-Kanade algorithm [28] with pixel object/background likelihoods. It shows competitive performance on a vehicle tracking dataset with severe viewpoint changes.

**Multiview Object Recognition.** Our tracker builds upon the idea of multiview recognition. The goal of multiview object recognition is to recognize objects from arbitrary viewpoints, which dates back to the early works in computer vision (e.g., [27,9]). Recent works in multiview object recognition either represent objects as collections of parts or features which are connected across views [39,36,37], or utilize explicit 3D models with associated visual features to represent objects [25,43,31,13,26]. Our method benefits from the 3D aspect part representation introduced in [43]. While [43] focuses on object detection and pose estimation from single images in a discretized viewpoint space, we show that the 3D aspect part representation can be utilized to estimate continuous object pose and 3D aspect part locations in multiview object tracking.

**3D Model-Based Tracking.** Multiview object recognition methods have been extended and applied to 3D tracking [35,10,24,6,34,30]. Most of the previous works aim at tracking the 3D pose of an object *instance* using its 3D CAD model, e.g., [10,6,34]. In contrast, we focus on 3D tracking of object *categories* with a 3D object category representation, which is able to handle the intra-class variability among object instances in the same category.

**Monocular vs. Multi-Camera Multiview Object Tracking.** An alternative way to achieve multiview object tracking is to utilize multi-camera settings, where the target is observed from multiple cameras simultaneously [21,23,17]. Tasks such as occlusion reasoning [21] and 3D reconstruction [17] which are challenging in monocular settings can be solved efficiently in multi-camera environments. Since multiple cameras are only available in specific scenarios, we focus on monocular multiview tracking in this work.

**3D Tracking and Reconstruction** In contrast to methods that track targets in 3D (e.g., [19,11,32]), we have access only to videos and do not use other sensor modalities such as range data. Compared with methods that perform joint 3D reconstruction and tracking (e.g., [16,18]), we are interested mainly in estimating the 3D pose and shape extent of the target in terms of its part layout.

## 3 Multiview Tracking Framework

The primary goal of multiview object tracking is to estimate the posterior distribution of the target's state $P(X_t, V_t|Z_{1:t})$ at the current time step $t$ given all observations $Z_{1:t}$ up to that time step, where $X_t$ and $V_t$ denote the location and viewpoint of the target at time $t$ respectively. Instead of tracking the object as a whole, which cannot handle the topological appearance change of object, we propose to track the 3D aspect parts of the object and its viewpoint jointly while modeling the relationship between these parts. By using a 3D aspect part representation of the object (Fig. 1(b)), we can predict the visibility and shape of the parts in arbitrary viewpoints. In this way, the tracking framework is able to handle the appearance change introduced by viewpoint transitions, especially in cases when a part disappears or reappears due to self-occlusion. Consequently, the location of the object at time $t$ is determined by the locations of the 3D aspect parts, i.e., $X_t = \{X_{it}\}_{i=1}^{n}$, where $n$ is the number of parts and $X_{it}$ denotes the location of part $i$ at time $t$. The viewpoint $V_t$ is represented by the azimuth $a_t$, elevation $e_t$ and distance $d_t$ of the camera position in 3D with respect to the object, i.e., $V_t = (a_t, e_t, d_t)$ as shown in Fig. 2(a).

By applying Bayes rule, the posterior distribution can be decomposed as

$$P(X_t, V_t|Z_{1:t}) \propto \tag{1}$$
$$\underbrace{P(Z_t|X_t, V_t)}_{\text{likelihood}} \int \underbrace{P(X_t, V_t|X_{t-1}, V_{t-1})}_{\text{motion prior}} \underbrace{P(X_{t-1}, V_{t-1}|Z_{1:t-1})}_{\text{posterior at time t-1}} dX_{t-1}dV_{t-1},$$

where the likelihood $P(Z_t|X_t, V_t)$ measures the probability of observing measurement $Z_t$ given the state of the target $(X_t, V_t)$ at time $t$, the motion prior $P(X_t, V_t|X_{t-1}, V_{t-1})$ predicts the state of the target at time $t$ given its previous state, and $P(X_{t-1}, V_{t-1}|Z_{1:t-1})$ is the posterior at time $t-1$.

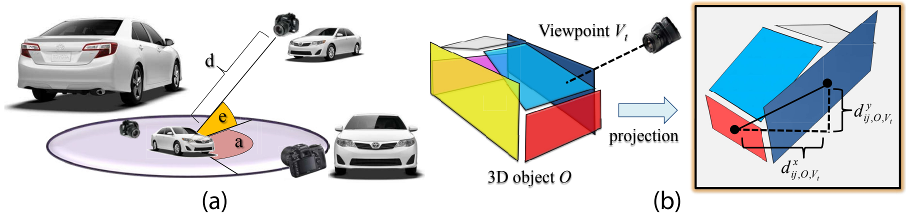

**Fig. 2.** (a) The viewpoint of the object is represented by the azimuth, elevation, and distance of the camera pose in 3D, $V = (a, e, d)$. (b) Illustration of the relative distance between two parts by projecting the 3D object onto a 2D image.

### 3.1 Likelihood

The likelihood $P(Z_t|X_t, V_t)$ measures the compatibility between the state of the target $(X_t, V_t)$ with the observation $Z_t$ at time $t$. Since we track an object by its 3D aspect parts, the likelihood of the object is decomposed as the product of the likelihoods of the 3D aspect parts:

$$P(Z_t|X_t, V_t) = \prod_{i=1}^{n} P(Z_t|X_{it}, V_t), \tag{2}$$

where $P(Z_t|X_{it}, V_t)$ denotes the appearance likelihood of part $i$. The likelihood is measured based on category-level pre-trained part appearance models. To make the likelihood more robust in some difficult scenarios (e.g., occlusion), we also use instance-level online-learned part appearance models in computing the likelihoods for 3D aspect parts. In traditional online object tracking, the likelihood of a part is computed using the appearance model of that part learned online, where the assumption is that the part is always visible during tracking. However, this is not necessarily true when the viewpoint changes. When parts with learned appearance models disappear and unseen parts become visible, the tracker loses the target. In our case, when new parts appear, if no online appearance models have been learned for them before, we resort to the category-level part templates to compute the likelihood. Subsequently, the online appearance models for the new parts are initialized according to the tracking output and updated afterwards. The online appearance model is updated according to the 3D pose, i.e., we only update the model for the visible parts. Specifically, we define the likelihood as:

$$P(Z_t|X_{it}, V_t) \propto \exp\Big(\Lambda_{\text{category}}(Z_t, X_{it}, V_t) + \Lambda_{\text{online}}(Z_t, X_{it}, V_t)\Big), \qquad (3)$$

where $\Lambda_{\text{category}}(Z_t, X_{it}, V_t)$ is the potential from the category-level part template for part $i$, and $\Lambda_{\text{online}}(Z_t, X_{it}, V_t)$ is the potential from the online appearance model for part $i$.

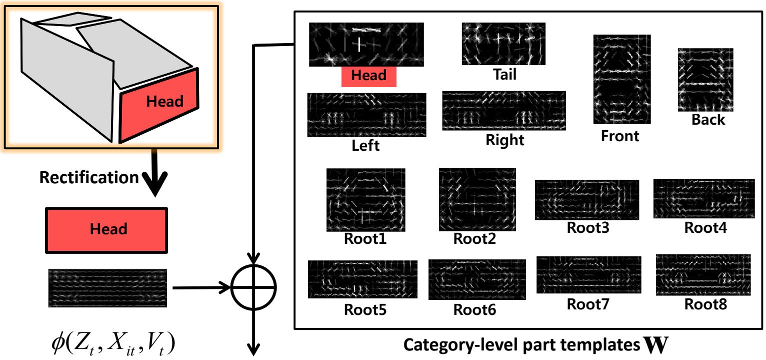

**Fig. 3.** Illustration of the category-level part templates and the computation of the potential for the Head part, where rectified HOG features are used

A category-level part template is trained with various instances in the same category, which captures the general shape of the part. We define the potential from the category-level part template as

$$\Lambda_{\text{category}}(Z_t, X_{it}, V_t) = \begin{cases} \mathbf{w}_i^T \phi(Z_t, X_{it}, V_t), & \text{if visible} \\ \alpha_i, & \text{if self-occluded,} \end{cases} \qquad (4)$$

where $(\mathbf{w}_i, \alpha_i)$ denotes the weights of the part template, and $\phi(Z_t, X_{it}, V_t)$ is the feature vector. The part template $\mathbf{w}_i$ is applied only if the part is visible.

Otherwise, an occlusion weight $\alpha_i$ is assigned to the part. We use rectified HOG features as $\phi(Z_t, X_{it}, V_t)$, where HOG features [8] are extracted after rectifying the image into the frontal view of the part according to the viewpoint $V_t$. Therefore, the part template $(\mathbf{w}_i, \alpha_i)$ corresponds to the frontal view of the part. This property is critical for continuous viewpoint estimation. In learning the part template from training images, the viewpoint space is discretized. During tracking, we can always first rectify the image into the frontal view of the part from arbitrary continuous viewpoint, and then apply the learned template. In this way, we are able to compute the likelihoods for continuous viewpoints during the Bayesian filtering tracking. All the part templates for 3D aspect parts are jointly learned from training images using a Structural SVM optimization as in [43]. Fig. 3 illustrates the learned category-level part templates and the rectified HOG features. Note that, besides training part templates for 3D aspect parts, we also introduce root templates which correspond to the whole object in different view sections and are obtained from DPM [12].

The online appearance models capture instance-level characteristics of part appearance, which are specialized to the current target. Moreover, the models are updated during tracking to accommodate appearance change. The potential of the online appearance model in Eq. (3) is defined as

$$\Lambda_{\text{online}}(Z_t, X_{it}, V_t) = \begin{cases} \mathbf{H}_i(\psi(Z_t, X_{it}, V_t)), & \text{if visible} \\ \lambda_0, & \text{if self-occluded,} \end{cases} \tag{5}$$

where $\mathbf{H}_i$ is the classifier for part $i$, $\psi(Z_t, X_{it}, V_t)$ is the feature vector and $\lambda_0$ is a constant assigned to the part if it is self-occluded. We utilize the multiple instance boosting algorithm [2] for training and updating the classifier $\mathbf{H}_i$ during tracking. The classifier is applied and updated only if the part is visible under the predicted viewpoint, which prevents the classifier from learning with incorrect appearance features. Similar to the rectified HOG features used in constructing the category-level part templates, we rectify the image to the frontal view of the part according to $V_t$ before extracting Haar-like features as in [40] for $\psi(Z_t, X_{it}, V_t)$. In this way, the online appearance model is robust to viewpoint distortions, and we can compute part likelihoods for continuous viewpoints.

### 3.2 Motion Prior

The motion prior $P(X_t, V_t | X_{t-1}, V_{t-1})$ predicts the current state of the target based on its previous state. We decompose the motion prior according to part location and viewpoint:

$$\begin{aligned} & P(X_t, V_t | X_{t-1}, V_{t-1}) \\ = & P(X_t | X_{t-1}, V_{t-1}, V_t) P(V_t | X_{t-1}, V_{t-1}) \\ = & P(X_t | X_{t-1}, V_t) P(V_t | V_{t-1}), \end{aligned} \tag{6}$$

where $P(X_t | X_{t-1}, V_t)$ models the change in location, and $P(V_t | V_{t-1})$ is the viewpoint motion. Note that in Eq. (6), two assumptions of conditional independence are imposed to simplify the motion prior. Inspired by [22] which uses a Markov Random Field (MRF) motion prior to capture the interaction between targets, we model the change in location using an MRF that is able to capture the relationships between parts:

$$P(X_t|X_{t-1}, V_t) \propto \prod_{i=1}^{n} P(X_{it}|X_{i(t-1)}) \prod_{(i,j)} \Lambda(X_{it}, X_{jt}, V_t), \tag{7}$$

where $P(X_{it}|X_{i(t-1)})$ is the motion model for part $i$ and $\Lambda(X_{it}, X_{jt}, V_t)$ is the pairwise potential which constrains the relative location of two parts according to the 3D aspect part representation and the viewpoint.

In order to handle abrupt location and viewpoint changes or occlusion, we do not impose a strong motion prior such as the constant velocity motion prior in our multiview tracker. The location motion of a part in Eq. (7) and the viewpoint motion in Eq. (6) are both modeled with Gaussian distributions centered on the previous location and the previous viewpoint respectively:

$$P(X_{it}|X_{i(t-1)}) \sim \mathcal{N}(X_{i(t-1)}, \sigma_x^2, \sigma_y^2) \tag{8}$$

$$P(V_t|V_{t-1}) \sim \mathcal{N}(V_{t-1}, \sigma_a^2, \sigma_e^2, \sigma_d^2), \tag{9}$$

where $\sigma_x^2$, $\sigma_y^2$, $\sigma_a^2$, $\sigma_e^2$ and $\sigma_d^2$ are the variances of the Gaussian distributions for 2D part center coordinates, azimuth, elevation and distance respectively.

To define the pairwise potential between part locations in Eq. (7), we utilize the 3D aspect part representation (Fig. 1(b)). Let $O$ denote the 3D object representation. Given the viewpoint $V_t$ at time $t$, we can project the 3D object onto the image according to $V_t$. Then we obtain the ideal relative distance $d_{ij,O,V_t}$ between part $i$ and part $j$ as shown in Fig. 2(b). We define the pairwise potential to penalize large deviations between the observed relative part locations from the ideal ones with Gaussian priors:

$$\begin{aligned} \Lambda(X_{it}, X_{jt}, V_t) &= P(\Delta_t(x_i, x_j)|V_t)P(\Delta_t(y_i, y_i)|V_t), \\ P(\Delta_t(x_i, x_j)|V_t) &\sim \mathcal{N}(d^x_{ij,O,V_t}, \sigma^2_{dx}), \\ P(\Delta_t(y_i, y_j)|V_t) &\sim \mathcal{N}(d^y_{ij,O,V_t}, \sigma^2_{dy}), \end{aligned} \tag{10}$$

where $X_{it} = (x_{it}, y_{it})$ and $X_{jt} = (x_{jt}, y_{jt})$ denote the 2D center coordinates of the two parts, $\Delta_t(x_i, x_j) = |x_{it} - x_{jt}|$, $\Delta_t(y_i, y_j) = |y_{it} - y_{jt}|$, $d^x_{ij,O,V_t}$ and $d^y_{ij,O,V_t}$ are the ideal relative distances between the two parts in the $x$ and $y$ directions respectively (Fig. 2(b)), and $\sigma^2_{dx}$ and $\sigma^2_{dy}$ are the variances of the Gaussian distributions for 2D relative distances, which are set proportionally to the size of the part in the image. The pairwise potential (10) allows the 3D shape of the target to deviate from the 3D object model with some deformation cost. Note that we use a general 3D aspect part representation for an object category and apply it to different instances of that category.

### 3.3 Particle Filtering Tracking

In order to track the continuous pose of the target, we employ the particle filtering technique to infer the posterior distribution in Eq. (1). We use Markov Chain

Monte Carlo (MCMC) sampling, where the posterior $P(X_{t-1}, V_{t-1}|Z_{1:t-1})$ at time $t-1$ is represented as a set of $N$ unweighted samples $P(X_{t-1}, V_{t-1}|Z_{1:t-1}) \approx (X_{t-1}^{(r)}, V_{t-1}^{(r)})_{r=1}^{N}$. So we obtain the following Monte Carlo approximation to the Bayesian filtering distribution:

$$P(X_t, V_t|Z_{1:t}) \propto P(Z_t|X_t, V_t) \sum_{r=1}^{N} P(X_t, V_t|X_{t-1}^{(r)}, V_{t-1}^{(r)}), \tag{11}$$

where $P(Z_t|X_t, V_t)$ is the likelihood and $P(X_t, V_t|X_{t-1}^{(r)}, V_{t-1}^{(r)})$ is given by the motion prior. At time $t$, we obtain a set of new samples by sampling from Gaussian proposal distributions on both part locations and viewpoint centered on samples at time $t-1$. Then the state of the target at time $t$, i.e., 3D aspect part locations and viewpoint, is predicted as the MAP of the posterior at time $t$, which is given by the sample with the largest posterior probability in Eq. (11). By sampling new viewpoints, we are able to predict the topological appearance change of the target, so as to apply and update the part templates accordingly. To initialize the tracker, we use the ground truth viewpoint in the first frame of the video, and aspect parts are initialized automatically by projecting the 3D aspect part model according to the viewpoint. Algorithm 1 summarizes our multiview tracking method using Bayesian particle filtering.

```
input : A video sequence $Z_{1:T}$, initial 3D aspect parts and viewpoint $(X_1, V_1)$
output: 3D aspect parts and viewpoints for the target in the video $(X_t, V_t)_{t=1}^{T}$

Initialize samples $(X_1^{(r)}, V_1^{(r)})_{r=1}^{N}$ for the first frame by sampling viewpoints and part
locations according to the motion prior (6) based on $(X_1, V_1)$;
for $t \leftarrow 2$ to $T$ do
    Initialize the MCMC sampler: randomly select a sample $(X_{t-1}^{(r)}, V_{t-1}^{(r)})$ as the initial
    state of the $(X_t, V_t)$ Markov chain;
    repeat
        Sample a new viewpoint from the Gaussian proposal density $Q(V_t'; V_t)$;
        Compute the visibility of 3D aspect parts under viewpoint $V_t'$;
        foreach part $i$ visible in both $V_t'$ and $V_t$ do
            Sample its location from the Gaussian proposal density $Q(X_{it}'; X_{it})$;
        end
        foreach part $i$ visible in $V_t'$ but not in $V_t$ do
            Compute its location $X_{it}'$ using the mean distance with respect to other
            visible parts according to the pairwise distributions (10);
        end
        Compute the acceptance ratio
            $a = \min\Big(1, \frac{P(X_t', V_t'|Z_{1:t})Q(X_t; X_t')Q(V_t; V_t')}{P(X_t, V_t|Z_{1:t})Q(X_t'; X_t)Q(V_t'; V_t)}\Big);$   (12)
        Accept the sample $(X_t', V_t')$ with probability $a$. If accepted, $(X_t, V_t) \leftarrow (X_t', V_t')$.
        Otherwise, leave $(X_t, V_t)$ unchanged;
    until $N$ samples are accepted;
    Obtain the new sample set $(X_t^{(r)}, V_t^{(r)})_{r=1}^{N}$, and find the MAP among it as the
    tracking output for frame $t$;
end
```

**Algorithm 1.** Multiview particle filtering object tracking

## 4 Experiments

We evaluate the performance of our multiview tracker on car tracking, since the ability to track cars is critical for various real world applications and it represents an informative case study in handling topological appearance change.

### 4.1 Datasets

The current benchmarks for evaluating trackers that handle appearance changes (e.g., [41]) are not built to emphasize the 'topological' appearance change of the target. So they are not suitable for evaluating our method whose main goal is to handle the topological appearance changes. Hence, we collected a new car tracking dataset of 9 video sequences that contain significant viewpoint change from YouTube. Each video contains one car to be tracked. To provide ground truth annotations for viewpoints and 3D aspect parts, we use the pose annotation tool proposed in [42], which computes accurate viewpoints and 3D aspect part locations of the targets using correspondences between 2D image points and 3D anchor points of CAD models. In order to test our multiview tracker in challenging real world scenarios, we also selected 11 sequences from the KITTI dataset [14] that contain significant viewpoint change. There can be multiple cars in each sequence, but we specify one car to track. In some sequences, the target is occluded temporarily which makes these sequences challenging. Finally, we evaluate our method on a standard sequence for car tracking from [20]. Unfortunately, this sequence does not contain significant viewpoint variations. Refer to the technical report in [1] for details of the annotation process and the statistics for the YouTube and the KITTI sequences.

### 4.2 Evaluation Measures

Our multiview tracker outputs not only the 2D bounding box of the target, but also its 3D pose and the 2D locations of the 3D aspect parts. So we evaluate the performance of our tracker on these three tasks and compare it with corresponding baselines. For 2D tracking, we report the Pascal VOC overlap ratio, which is defined as $R = Area(B_T \cap B_{GT})/Area(B_T \cup B_{GT})$, where $B_T$ is the predicted bounding box of the target and $B_{GT}$ is the ground truth bounding box.

For viewpoint estimation, we report two metrics. The first metric is the viewpoint accuracy, where an estimated viewpoint is considered to be correct if the deviation between the estimated azimuth and the ground truth azimuth is within $15°$. The second metric is the absolute difference in azimuth between the ground truth viewpoint and the estimated viewpoint. Since the elevation change is small in the sequences in our experiments, we do not present detailed evaluation in elevation estimation.

For 3D aspect part localization, we also use the Pascal VOC overlap ratio, where the intersection over union is computed between the predicted part shape and the ground truth part shape. If a visible part is predicted as self-occluded, the overlap ratio is zero. So we penalize incorrect aspect estimation of the target.

We measure the viewpoint and part locations for the target in one frame only if the target is correctly tracked in the frame, i.e., its overlap ratio with ground truth bounding box is larger than 0.5.

### 4.3 Experimental Settings

The following parameters have been set experimentally and remain fixed for all of the experiments with different sequences. In the motion prior, the standard deviations of part center coordinates in Eq. (8) are set to $\sigma_x = 4 \cdot w$ and $\sigma_y = 4 \cdot h$, where $w$ and $h$ denote the width and height of the part respectively. The standard deviations of viewpoint in Eq. (9) are set to $\sigma_a = 135°$, $\sigma_e = 5°$ and $\sigma_d = 10$. We use large standard deviations for both part location and viewpoint in order to recover from tracking failures due to occlusions or noisy responses from part templates. In the pairwise potential, both the standard deviations in Eq. (10) are set to $\sigma_{dx} = \sigma_{dy} = h/4$. In Eq. (5), the constant $\lambda_0$ can be arbitrary since we only compare the common visible parts of two samples when selecting the MAP sample (Algorithm 1). We compute 40 (viewpoints) × 200K (part locations) samples per frame since the joint space of viewpoint and all parts is huge. To train the templates for 3D aspect parts, we use the 3DObject dataset [36]. For the templates in DPM, we use the car model pre-trained on PASCAL'07 [12].

### 4.4 Results

**2D Object Tracking.** Tab. 1 shows the 2D object tracking results in terms of average bounding box overlap ratio on our new car tracking dataset, the KITTI sequences and the 06_car sequence from [20], where we compare our multiview tracker with several baselines. First, four state-of-the-art online tracking methods, MIL [2], L1 [3], TLD [20] and Struct [15], perform poorly on our new dataset and the KITTI sequences. Their mean overlap ratios are below 0.5. This is mainly because these online tracking methods cannot handle the topological appearance change of the cars. When the viewpoint changes, the online trackers keep tracking just a single portion of the object or even lose the target (Fig. 4).

It is evident that the category-level part templates contribute significantly in the multiview tracking setting. In Tab. 1, "Category Model" column shows the case that we use only the category-level part templates in our particle filtering framework without using online learning (refer to Eq. (3)). We can see that "Category Model" improves over the best online tracker by 30% on the new dataset and 19% on the KITTI sequences in terms of mean overlap ratio. By leveraging the 3D aspect part representation and estimating the viewpoint, our "Category Model" is able to predict the aspect change of the target and track the target in different views.

Our full model takes advantages of both category-level part templates and online-learned part appearance models, and it achieves the best mean overlap ratio on the YouTube dataset and the KITTI sequences. The highest improvement is for Race5 and KITTI03, where "Category Model" fails to track the car due to occlusion by smoke and another car, respectively. By combining online

**Table 1.** 2D object tracking performance using average bounding box overlap ratio

| Video | MIL [2] | L1 [3] | TLD [20] | Struct [15] | DPM [12]+PF | Category Model | Full Model |
|---|---|---|---|---|---|---|---|
| Race1 | 0.34 | 0.39 | 0.20 | 0.36 | 0.68 | 0.68 | **0.69** |
| Race2 | 0.49 | 0.49 | 0.28 | 0.50 | **0.74** | **0.74** | 0.73 |
| Race3 | 0.36 | 0.26 | 0.25 | 0.44 | 0.74 | 0.74 | **0.77** |
| Race4 | 0.53 | 0.56 | 0.47 | 0.63 | **0.76** | **0.76** | **0.76** |
| Race5 | 0.29 | 0.54 | 0.28 | 0.26 | 0.63 | 0.63 | **0.68** |
| Race6 | 0.27 | 0.53 | 0.48 | 0.29 | 0.76 | 0.76 | **0.77** |
| SUV1 | 0.58 | **0.81** | 0.56 | 0.60 | 0.78 | 0.78 | 0.78 |
| SUV2 | 0.18 | 0.12 | 0.53 | 0.24 | **0.77** | **0.77** | **0.77** |
| Sedan | 0.26 | 0.23 | 0.33 | 0.30 | **0.78** | **0.78** | **0.78** |
| **Mean** | 0.37 | 0.44 | 0.38 | 0.40 | 0.74 | 0.74 | **0.75** |
| KITTI01 | 0.20 | 0.40 | 0.44 | 0.33 | 0.65 | 0.64 | **0.69** |
| KITTI02 | 0.28 | 0.18 | 0.20 | 0.12 | 0.26 | 0.26 | **0.32** |
| KITTI03 | 0.37 | **0.59** | 0.42 | 0.36 | 0.20 | 0.19 | 0.50 |
| KITTI04 | 0.31 | 0.12 | **0.36** | 0.34 | 0.67 | 0.33 | 0.33 |
| KITTI05 | 0.40 | 0.32 | 0.51 | 0.41 | 0.54 | **0.73** | 0.72 |
| KITTI06 | 0.64 | 0.21 | 0.54 | **0.65** | **0.65** | **0.65** | 0.56 |
| KITTI07 | 0.12 | 0.33 | 0.03 | 0.28 | **0.66** | 0.65 | **0.66** |
| KITTI08 | 0.58 | 0.13 | 0 | 0.66 | **0.74** | **0.74** | 0.72 |
| KITTI09 | 0.18 | 0.15 | 0 | 0.17 | 0.18 | 0.51 | **0.52** |
| KITTI10 | 0.33 | 0.46 | 0.41 | 0.35 | **0.68** | **0.68** | **0.68** |
| KITTI11 | 0.28 | 0.23 | 0.24 | 0.28 | **0.71** | **0.71** | 0.68 |
| **Mean** | 0.34 | 0.28 | 0.29 | 0.36 | 0.54 | 0.55 | **0.58** |
| 06_car [20] | 0.19 | 0.52 | **0.85** | 0.48 | 0.70 | 0.67 | 0.70 |

appearance models, the full model can recover from occlusion and track the car by adapting its appearance models. Fig. 4 shows some tracking outputs from our multiview tracker on SUV1 and Race1. Fig. 5 displays some tracking results on KITTI03, where our full Model recovers from occlusion, but the "Category Model" switches to the occluder.

We also compare our method with a tracking-by-detection baseline, which applies particle filtering to the output of a detector (DPM [12]). Our result is on par with this baseline for 2D object localization in the YouTube and 06_car sequences, and we provide 4% improvement on the KITTI dataset. However, note that this baseline and the online trackers baselines are not able to provide the estimates for the viewpoint and aspect part locations.

The results on the 06_car sequence from [20] demonstrate that our multiview tracker can handle the degenerate case where the viewpoint of the target does not change. MIL, L1 and Struct drift due to occlusion by trees, while TLD is well designed to recover from occlusion and achieves the best performance on this sequence. Our method also recovers from occlusion but obtains lower average overlap ratio than TLD. One main reason is that the elevation angle of the car in this sequence is totally different from that of the instances we used for training the category-level part templates (see [1] for tracking videos on these datasets).

**Continuous Viewpoint Estimation.** The left half of Tab. 2 shows the viewpoint accuracy and the mean absolute difference in azimuth for viewpoint estimation on our new car dataset and the KITTI sequences. We compare our "Full Model" and "Category Model" with the state-of-the-art object pose estimator ALM [43]. Since ALM does not output tracks of targets, we compare the three models on the commonly tracked frames between the "Full Model" and the

**Table 2.** Viewpoint accuracy/mean absolute difference in azimuth and average overlap ratio of 3D aspect part on our new car dataset and the KITTI sequences

| | Viewpoint Estimation | | | 3D Aspect Part Localization | | |
|---|---|---|---|---|---|---|
| Video | Full Model | Category Model | ALM [43] | Full Model | Category Model | ALM [43] |
| Race1 | **0.67/18.73°** | 0.59/22.88° | 0.52/42.62° | **0.40** | 0.39 | 0.35 |
| Race2 | **0.77/10.83°** | 0.60/12.65° | 0.53/44.30° | **0.45** | 0.38 | 0.34 |
| Race3 | **0.83**/9.28° | **0.83/7.79°** | 0.64/46.08° | 0.45 | **0.48** | 0.31 |
| Race4 | 0.69/15.83° | 0.68/14.67° | **0.79/13.37°** | **0.48** | 0.47 | 0.42 |
| Race5 | 0.71/**10.75°** | **0.74**/11.78° | 0.54/57.79° | **0.44** | 0.42 | 0.28 |
| Race6 | **0.43/18.47°** | 0.40/21.34° | 0.31/37.08° | **0.35** | **0.35** | 0.29 |
| SUV1 | **0.82/7.81°** | 0.75/8.52° | 0.47/78.38° | **0.42** | 0.40 | 0.24 |
| SUV2 | **0.57/19.56°** | 0.45/56.33° | 0.39/63.41° | **0.30** | 0.23 | 0.18 |
| Sedan | 0.76/9.87° | 0.78/**9.50°** | **0.79**/20.84° | 0.44 | **0.45** | 0.43 |
| **Mean** | **0.69/13.46°** | 0.65/18.38° | 0.54/47.24° | **0.41** | 0.40 | 0.30 |
| KITTI01 | **0.95/6.54°** | 0.74/8.53° | 0.57/44.46° | **0.49** | 0.41 | 0.37 |
| KITTI02 | **1.00/5.40°** | 0.20/30.06° | 0.33/119.54° | **0.60** | 0.15 | 0.13 |
| KITTI03 | **0.42**/15.64° | **0.42/15.14°** | 0.50/15.99° | **0.33** | **0.33** | 0.24 |
| KITTI04 | 0.22/27.05° | **0.25/26.03°** | 0.17/58.42° | **0.22** | **0.22** | 0.14 |
| KITTI05 | 0.36/23.59° | 0.40/**22.17°** | **0.64**/23.65° | 0.23 | **0.25** | **0.25** |
| KITTI06 | 0.31/21.63° | 0.29/21.58° | **0.59/20.29°** | 0.21 | 0.21 | **0.23** |
| KITTI07 | **0.96/6.86°** | 0.89/7.92° | 0.70/24.50° | **0.48** | **0.48** | 0.39 |
| KITTI08 | 0.57/**15.61°** | 0.48/23.84° | **0.67**/23.26° | **0.37** | 0.29 | 0.26 |
| KITTI09 | **0.50**/21.63° | 0.42/78.67° | **0.50/17.60°** | **0.28** | 0.16 | 0.23 |
| KITTI10 | **0.81/7.99°** | 0.79/9.44° | 0.44/56.78° | **0.39** | **0.39** | 0.21 |
| KITTI11 | **0.88/9.33°** | 0.78/11.80° | 0.68/12.29° | 0.39 | 0.40 | **0.41** |
| **Mean** | **0.63/14.66°** | 0.51/23.20° | 0.53/37.89° | **0.36** | 0.30 | 0.26 |

"Category Model", where we use the most confident detection with overlap ratio larger than 0.5 from ALM as its output. It is clear that "Category Model" outperforms ALM in viewpoint estimation significantly. By utilizing the temporal information from videos, our multiview tracker estimates continuous viewpoints in the particle filtering framework and smoothes the viewpoint estimation via the motion prior. ALM discretizes the viewpoint space into 24 azimuth angles (i.e., 15° interval) and it does not use the temporal information. By combining online appearance models for 3D aspect parts, our full model improves over the "Category Model" by 4%/5° and 12%/9°, and over ALM by 15%/34° and 10%/23° in terms of mean accuracy/mean absolute difference in azimuth on the two datasets respectively. Online appearance models help 2D localization of 3D aspect parts, which in turn benefits viewpoint estimation. Our full model achieves 4.6° mean absolute difference in elevation on the YouTube sequences. Fig. 4 also shows some viewpoint estimation results from our multiview tracker and ALM.

**3D Aspect Part Localization.** The right half of Tab. 2 shows the 3D aspect part localization performance in terms of PASCAL VOC overlap ratio on our new car dataset and the KITTI sequences. Compared with ALM [43], "Category Model" achieves much better mean overlap ratio. Since part locations and viewpoint are jointly optimized in our multiview tracking framework, the category-level part templates and the motion prior result in accurate viewpoint and 2D part locations. Consequently, the 2D part shapes can be estimated more accurately. By introducing online appearance learning, our full model further improves the 3D aspect part localization, where it outperforms or is on par with the "Category Model" in 7 of the 9 YouTube sequences and in 9 of the 11 KITTI

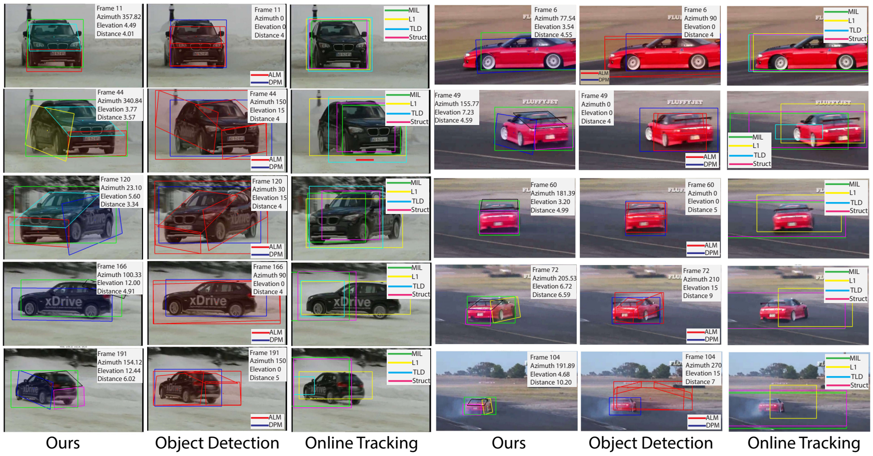
Frame 11 Azimuth 357.82 Elevation 4.49 Distance 4.01
Frame 11 Azimuth 0 Elevation 0 Distance 4
Frame 6 Azimuth 77.54 Elevation 3.54 Distance 4.55
Frame 6 Azimuth 90 Elevation 0 Distance 4
Frame 44 Azimuth 340.84 Elevation 3.77 Distance 3.57
Frame 44 Azimuth 150 Elevation 15 Distance 4
Frame 49 Azimuth 155.77 Elevation 7.23 Distance 4.59
Frame 49 Azimuth 0 Elevation 0 Distance 4
Frame 120 Azimuth 23.10 Elevation 5.60 Distance 3.34
Frame 120 Azimuth 30 Elevation 15 Distance 4
Frame 60 Azimuth 181.39 Elevation 3.20 Distance 4.99
Frame 60 Azimuth 0 Elevation 0 Distance 5
Frame 166 Azimuth 100.33 Elevation 12.00 Distance 4.91
Frame 166 Azimuth 90 Elevation 0 Distance 4
Frame 72 Azimuth 205.53 Elevation 6.72 Distance 6.59
Frame 72 Azimuth 210 Elevation 15 Distance 9
Frame 191 Azimuth 154.12 Elevation 12.44 Distance 6.02
Frame 191 Azimuth 150 Elevation 0 Distance 5
Frame 104 Azimuth 191.89 Elevation 4.68 Distance 10.20
Frame 104 Azimuth 270 Elevation 15 Distance 7
ALM
DPM
MIL
L1
TLD
Struct
Ours
Object Detection
Online Tracking
Ours
Object Detection
Online Tracking

**Fig. 4.** Tracking/Detection outputs from different methods on SUV1 and Race1. "Ours" are the tracking outputs from our multiview tracker. "Object Detection" shows the detection results from DPM [12] and ALM [43]. "Online Tracking" shows the tracking results of four state-of-the-art online tracking methods: MIL [2], L1 [3], TLD [20] and Struct [15].

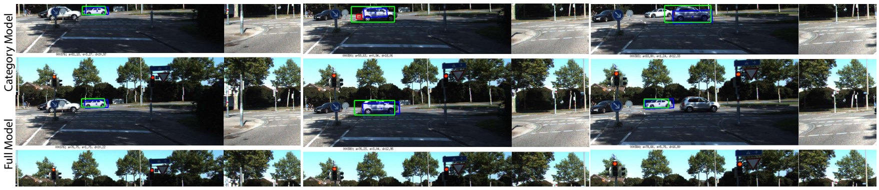

**Fig. 5.** The tracking results on KITTI03. "Category Model" fails to track the target and switches to the occluder, while our full model is able to recover from occlusion and track the correct target.

sequences. In Fig. 4, we can see that the 3D aspect parts from our tracker are more accurate than those obtained by ALM.

## 5 Conclusion

We proposed a novel multiview rigid object tracking framework to handle the topological appearance change of objects caused by viewpoint transitions. Our multiview tracker is able to predict the aspect change of the target, and track the continuous pose and the 3D aspect parts of the target. We conducted experiments on a new challenging car dataset and a set of KITTI sequences with large viewpoint variations, as well as on a standard sequence for car tracking. We demonstrated that our method is effective in tracking continuous 3D pose and aspect part locations, and it is able to handle the changes in viewpoint robustly.

**Acknowledgments.** We acknowledge the support of DARPA UPSIDE grant A13-0895-S002 and NSF CAREER grant N.1054127.

## References

1. http://cvgl.stanford.edu/projects/multiview_tracking
2. Babenko, B., Yang, M.H., Belongie, S.: Robust object tracking with online multiple instance learning. TPAMI 33(8), 1619–1632 (2011)
3. Bao, C., Wu, Y., Ling, H., Ji, H.: Real time robust l1 tracker using accelerated proximal gradient approach. In: CVPR (2012)
4. Breitenstein, M.D., Reichlin, F., Leibe, B., Koller-Meier, E., Van Gool, L.: Online multiperson tracking-by-detection from a single, uncalibrated camera. TPAMI 33(9), 1820–1833 (2011)
5. Butt, A.A., Collins, R.T.: Multi-target tracking by lagrangian relaxation to min-cost network flow. In: CVPR (2013)
6. Choi, C., Christensen, H.I.: Real-time 3D model-based tracking using edge and keypoint features for robotic manipulation. In: ICRA, pp. 4048–4055 (2010)
7. Choi, W., Pantofaru, C., Savarese, S.: A general framework for tracking multiple people from a moving camera. TPAMI (2012)
8. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR (2005)
9. Dickinson, S.J., Pentland, A.P., Rosenfeld, A.: From volumes to views: An approach to 3-d object recognition. CVGIP: Image Understanding 55(2), 130–154 (1992)
10. Drummond, T., Cipolla, R.: Real-time visual tracking of complex structures. TPAMI 24(7), 932–946 (2002)
11. Feldman, A., Hybinette, M., Balch, T.: The multi-iterative closest point tracker: An online algorithm for tracking multiple interacting targets. Journal of Field Robotics (2012)
12. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. TPAMI (2010)
13. Fidler, S., Dickinson, S., Urtasun, R.: 3D object detection and viewpoint estimation with a deformable 3D cuboid model. In: NIPS (2012)
14. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: CVPR (2012)
15. Hare, S., Saffari, A., Torr, P.H.: Struck: Structured output tracking with kernels. In: ICCV (2011)
16. Held, D., Levinson, J., Thrun, S.: Precision tracking with sparse 3D and dense color 2D data. In: ICRA (2013)
17. Hofmann, M., Wolf, D., Rigoll, G.: Hypergraphs for joint multi-view reconstruction and multi-object tracking. In: CVPR (2012)
18. Huang, Q.X., Adams, B., Wand, M.: Bayesian surface reconstruction via iterative scan alignment to an optimized prototype. In: Eurographics Symposium on Geometry Processing (2007)
19. Kaestner, R., Maye, J., Pilat, Y., Siegwart, R.: Generative object detection and tracking in 3D range data. In: ICRA (2012)
20. Kalal, Z., Mikolajczyk, K., Matas, J.: Tracking-learning-detection. TPAMI 34(7), 1409–1422 (2012)
21. Khan, S.M., Shah, M.: Tracking multiple occluding people by localizing on multiple scene planes. TPAMI 31(3), 505–519 (2009)

22. Khan, Z., Balch, T., Dellaert, F.: Mcmc-based particle filtering for tracking a variable number of interacting targets. TPAMI 27(11), 1805–1819 (2005)
23. Leal-Taixé, L., Pons-Moll, G., Rosenhahn, B.: Branch-and-price global optimization for multi-view multi-target tracking. In: CVPR (2012)
24. Lepetit, V., Fua, P.: Monocular model-based 3d tracking of rigid objects: A survey. Foundations and Trends in Computer Graphics and Vision 1(1), 1–89 (2005)
25. Liebelt, J., Schmid, C., Schertler, K.: Viewpoint-independent object class detection using 3D feature maps. In: CVPR (2008)
26. Lim, J.J., Pirsiavash, H., Torralba, A.: Parsing ikea objects: Fine pose estimation. In: ICCV (2013)
27. Lowe, D.G.: Three-dimensional object recognition from single two-dimensional images. Artificial Intelligence 31(3), 355–395 (1987)
28. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: Proceedings of Imaging Understanding Workshop (1981)
29. Oron, S., Bar-Hillel, A., Avidan, S.: Extended lucas kanade tracking. In: ECCV (2014)
30. Pauwels, K., Rubio, L., Diaz, J., Ros, E.: Real-time model-based rigid object pose estimation and tracking combining dense and sparse visual cues. In: CVPR, pp. 2347–2354 (2013)
31. Pepik, B., Stark, M., Gehler, P., Schiele, B.: Teaching 3D geometry to deformable part models. In: CVPR (2012)
32. Petrovskaya, A., Thrun, S.: Model based vehicle tracking for autonomous driving in urban environments. In: RSS (2008)
33. Pirsiavash, H., Ramanan, D., Fowlkes, C.C.: Globally-optimal greedy algorithms for tracking a variable number of objects. In: CVPR (2011)
34. Prisacariu, V.A., Reid, I.D.: Pwp3D: Real-time segmentation and tracking of 3D objects. IJCV 98(3), 335–354 (2012)
35. Roller, D., Daniilidis, K., Nagel, H.H.: Model-based object tracking in monocular image sequences of road traffic scenes. IJCV 10(3), 257–281 (1993)
36. Savarese, S., Fei-Fei, L.: 3D generic object categorization, localization and pose estimation. In: ICCV (2007)
37. Su, H., Sun, M., Fei-Fei, L., Savarese, S.: Learning a dense multi-view representation for detection, viewpoint classification and synthesis of object categories. In: ICCV (2009)
38. Supancic III, J.S., Ramanan, D.: Self-paced learning for long-term tracking. In: CVPR (2013)
39. Thomas, A., Ferrari, V., Leibe, B., Tuytelaars, T., Schiele, B., Van Gool, L.: Towards multi-view object class detection. In: CVPR (2006)
40. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: CVPR (2001)
41. Wu, Y., Lim, J., Yang, M.H.: Online object tracking: A benchmark. In: CVPR (2013)
42. Xiang, Y., Mottaghi, R., Savarese, S.: Beyond pascal: A benchmark for 3D object detection in the wild. In: WACV (2014)
43. Xiang, Y., Savarese, S.: Estimating the aspect layout of object categories. In: CVPR (2012)
44. Yang, B., Nevatia, R.: An online learned crf model for multi-target tracking. In: CVPR (2012)
45. Yao, R., Shi, Q., Shen, C., Zhang, Y., van den Hengel, A.: Part-based visual tracking with online latent structural learning. In: CVPR (2013)