NYC3DCars: A Dataset of 3D Vehicles in Geographic Context

Kevin Matzen
Cornell University

Noah Snavely
Cornell University

Abstract

Geometry and geography can play an important role in recognition tasks in computer vision. To aid in studying connections between geometry and recognition, we introduce NYC3DCars*, a rich dataset for vehicle detection in urban scenes built from Internet photos drawn from the wild, focused on densely trafficked areas of New York City. Our dataset is augmented with detailed geometric and geographic information, including full camera poses derived from structure from motion, 3D vehicle annotations, and geographic information from open resources, including road segmentations and directions of travel. NYC3DCars can be used to study new questions about using geometric information in detection tasks, and to explore applications of Internet photos in understanding cities. To demonstrate the utility of our data, we evaluate the use of the geographic information in our dataset to enhance a parts-based detection method, and suggest other avenues for future exploration.*

1. Introduction

Methods for 3D reconstruction and for object recognition have each, separately, seen significant advances in recent years. Yet there has been relatively little recent work exploring the intersection of these two areas despite evidence that explicit 3D reasoning can aid in recognition tasks [15]. We believe that new datasets that combine structure from motion (SfM)-style models with recognition methods can result in fruitful new approaches to scene understanding that leverage detailed camera and scene geometry. Towards this end, we present a new vehicle recognition dataset, NYC3DCars, comprised of challenging urban photos from the wild, and augmented with rich geometric and geographic information. Our dataset enables the study of new questions about the use of rich geometric data in recognition tasks, and for new applications in *geography-aware vision*, where image understanding is grounded in a geographic setting.

In particular, NYC3DCars consists of over two thousand annotated Internet photos from New York City, from a wide range of viewpoints, times of day, and camera models. The dataset includes (1) **camera viewpoints** for the photo collection solved for using SfM, anchored in a geographic coordinate system; (2) **detailed ground truth 3D vehicle annotations**, including 3D pose and vehicle type; and (3) **geographic data** associated with roads, sidewalks, and buildings in the surrounding scene, drawn from online resources. This data is illustrated in Figures 1, 2, and 4. Compared to existing datasets with vehicle pose information, ours has a richer variety of photos, and comes with detailed geographic data. Our dataset can serve as a benchmark for pose-sensitive vehicle detection in the wild, a problem we evaluate in Section 6. Moreover, our data, with its rich annotations and geometric information, can be used to explore a range of research questions in computer vision:

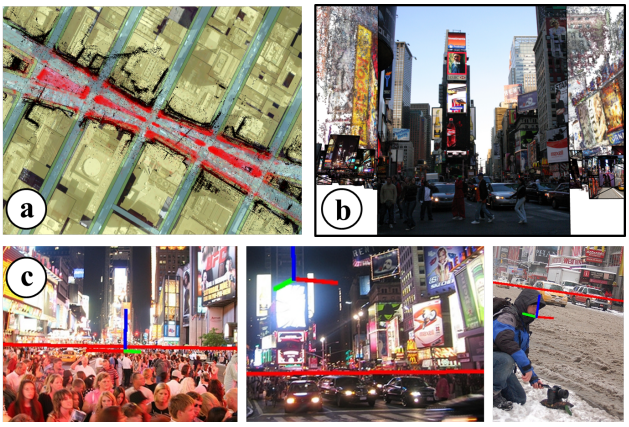

Figure 1. **NYC3DCars images and geometry**. (a) An overhead view showing the 3D SfM model overlaid on a map. Black points represent 3D scene points, and a red heat map shows the distribution of reconstructed camera positions (note that they tend to appear on sidewalks). The map is color-coded according to our geographic data. (b) A view of the SfM model from the perspective of one of the reconstructed images. The point cloud, and other camera frusta, are visible outside the frame of the central photo. (c) Several other reconstructed photos, illustrating the diversity in our data. Each image has a red horizon line, obtained from the camera extrinsics, overlaid, as well as axes specifying global east (red), north (green), and up (blue). Please zoom in for best results.

New methods. Given multiple images of the same scene from different viewpoints and times, can we improve detection methods using information estimated from SfM? SfM can immediately provide information such as horizons and

focal lengths useful in geometric reasoning, as well as additional information in the form of 3D scene structure. At the same time, open geographic data is proliferating online with sources such as OpenStreetMap. How useful is this rich 3D data for recognition problems? Our dataset provides a testbed for studying such questions.

New applications in urban scene understanding. Vision methods are largely unplugged from the real world, in that geographic information about the world is largely untapped in vision, and, conversely, vision methods estimate properties of images, but generally do not tie these back to observations about the world. Because our data is georeferenced, our dataset can be used to explore new applications of geography-aware vision, in which image observations can be related to real-world coordinates. For instance, a vehicle detected in an image can be placed at a real position on a street, and detections aggregated across many Internet photos can potentially be used to study traffic patterns or other large-scale phenomena, leveraging Internet photos as a new source of data for understanding cities.

We perform an initial study of how aspects of our data can be used to improve object detection, namely by incorporating geographic data (such as roadbed polygons and directions of travel) into a detection pipeline. We show promising results for this task, but believe the true power of our dataset will be in enabling the study of a range of geometric approaches.

In summary, our paper makes two main contributions: first, the NYC3DCars dataset itself, and the methodology for creating it, including a new online 3D annotation toolkit[1]; and second, a study of how the information in our dataset can be used within a detection framework. We close with a discussion of biases and other limitations of our dataset, as well as directions for its future applications.

2. Related Work

Detection datasets. Several datasets and benchmarks have been influential in vision in recent years. Notably, the PASCAL VOC benchmarks, with their wide variety of images, have driven work in recognition [4]. Our work incorporates much more detailed geometry than PASCAL, including 3D vehicle poses, as in related datasets [23, 19, 10]. Among these, perhaps the most closely related is the recent KITTI dataset [7]. However, KITTI is focused on the goal of autonomous driving, and so the images are all captured from the top of a vehicle with the same camera. We also provide vehicles with precise 3D pose, but for much more unconstrained imagery drawn from the Web, with the wide variety in viewpoint, illumination, image resolution, and other factors more typical of benchmarks such as VOC. Others have also presented 3D annotated vehicle datasets (e.g., [23, 19, 10]), but these generally contain carefully captured images that lack the variety of our own data. Our dataset also incorporates new types of geographic information, such as road data.

Geometry and recognition. Our dataset provides a framework for exploring new ways to combine explicit 3D reasoning with recognition methods. Our geographic data—with its roads, sidewalks, etc.—can be considered a "stage" within which one can reason about objects and their placement in the scene. This idea is related to prior methods that infer geometric properties and use them in image understanding tasks, for both outdoor [14, 15, 24] and indoor [12, 6] scenes. Our work allows for similar reasoning, but leverages much richer information derived from SfM and from geographic data sources. Hays and Efros augment images with coarse geographic data, such as elevation and population density [11], based on a rough global position. Our work is based on precise camera viewpoints, and allows for reasoning based on much more specific, pixel-level information, such as road segment polygons (see Figure 4). Hejrati and Ramanan propose a 3D detection approach [13], but the models are trained entirely with 2D annotations. Finally, other work combines multi-view reasoning with object recognition [3, 2], generally for images taken at the same time. In contrast, we build models from photos taken from widely varying times, and so individual objects will differ from photo to photo.

3. The NYC3DCars Dataset

In creating NYC3DCars, we decided to start from a vibrant urban area—Times Square in New York City—and create a dataset of photos in the wild for which we have both detailed ground truth 3D vehicle annotations, as well as geographic data describing streets and other static elements of the city. To our knowledge, ours is the first detection dataset of real-world Internet imagery along with camera poses and detailed 3D annotations. Beyond the dataset itself, a key contribution is a Web-based 3D labeling interface, available online. NYC3DCars consists of three components:

1. A set of Flickr photos of NYC and 3D structure from motion (SfM) models reconstructed from these photos and georegistered to the world. These photos span a variety of viewpoints, camera models, illuminations, and times of year. Each photograph has computed extrinsics and intrinsics in a geographic coordinate system.
2. 3D ground truth vehicles labeled in a set of photos, with each vehicle annotated with a geolocation, orientation, vehicle type, and level of occlusion.
3. Geographic data describing roads, sidewalks, medians, elevation, etc., obtained from `www.nyc.gov` and `www.openstreetmap.org`.

We now describe each of these components in turn.

[1] Dataset and tools available at `nyc3d.cs.cornell.edu`.

3.1. Input Photos and SfM Model

To begin, we downloaded 14,000 geotagged photos taken around Times Square from Flickr; these photos were taken between the years 2000 and 2008, by over 1,000 distinct photographers. Using these photos as input, we ran an SfM pipeline to reconstruct 3D camera geometry and a 3D point cloud [1]. Of the 14,000 input photos, 5,186 images were reconstructed, along with 567K 3D points. This 3D model was georegistered by downloading geotagged Google Street View photos from the same area, adding these to the SfM reconstruction [16], then using these photos as anchors to roughly align the model to the world via absolute orientations. Finally, we ran the Iterated Closest Point algorithm (ICP) between the 3D SfM point cloud and a publicly available aerial LIDAR scan of Manhattan to refine the alignment. The reconstruction is shown in Figure 1, along with several images from the dataset. This dataset can also serve as an index for georegistering new photos, through 2D-to-3D matching techniques [16].

For each photo, the SfM reconstruction (or a later registration to the SfM model) gives us its extrinsics—its position and orientation, in a geographic coordinate system—as well as intrinsics including focal length and radial distortion parameters. This information is already very useful in detection problems, e.g., in determining the horizon, or in detailed reasoning involving depth or perspective. Moreover, the fact that the data is georegistered allows us to draw on additional sources of geographic data for detection, as described below.

3.2. Annotated 3D Vehicles

Our goal is to provide a richly annotated set of ground truth vehicles that can be used for detection tasks, but also for recovering and evaluating the 3D position and pose of detected cars. To create such ground truth, we designed a new Web-based tool for 3D vehicle annotation.

We considered several existing annotation tools, but found that none met our specific goals. Many, such as LabelMe [22], provide for labeling of 2D regions, but not 3D objects. Little *et al*. provide an interface in which a user is asked to pose wireframe car renderings to annotate webcam images [17]. We tested this interface, but found that for our highly varied images it was difficult to adjust all of the degrees of freedom necessary to accurately place a vehicle in each photo; in particular, the three orientation angles were difficult to set. For these reasons, we created a new interface that restricts the number of free parameters in posing a car as much as possible, using the extra information from the estimated camera pose of the photo. This makes the task of annotating an image simpler and more efficient.

In addition to camera pose and intrinsics from SfM, the absolute scale of the scene is known from the georegistration process, and we assume that cars are supported by a planar ground surface. To label a photo, we set up an interface

Figure 2. **Ground truth annotation interface.** Several 3D cars are shown placed in the scene on a virtual ground plane. Please see the supplemental video for a recorded annotation session.

where a user looks "through" the photo from the correct camera viewpoint, and can slide and rotate vehicles in 3D on a rendered ground plane with the correct perspective.

Our interface is shown in Figure 2. 3D vehicles of various types can be placed into the scene, then moved and rotated until they align with the actual vehicles in the image. The user may also adjust the height of the camera above the ground plane in order to correct for mis-estimated camera heights in the SfM model. With these limited degrees of freedom, we found that most images were easy to annotate. In addition, since the camera poses are in a geographic coordinate system, each vehicle is placed at a real position in the world, and we can record its latitude, longitude, and heading.

For each photo, a user is asked to label all cars as long as he or she can confidently determine the pose of the car in 3D (even if it is partially occluded, as is often the case). The user can also indicate that either the ground plane or photo is defective (and give a reason, e.g. extreme camera blur). After this initial labeling step, several post-processing steps take place. Users label each photo as day or night, and label the occlusion level of each annotated vehicle on a scale from "fully visible" to "fully occluded." Because our 3D proxy models may not fit the annotated vehicle exactly, we also have users correct each 2D bounding box. Finally, to keep track of vehicles that users were unable to label (e.g. too far away, too occluded, not sure if actually a car), we have users click on all the objects they think are cars in the image.

Our Times Square dataset was labeled by students hired as annotators, and contains 1,287 labeled photos and 3,787 labeled vehicles with a wide variety of occlusion levels, truncation, pose, and time of day, each with a geolocation, heading, and vehicle class. Users flagged an additional 444 photographs as having a mis-estimated ground plane, 712

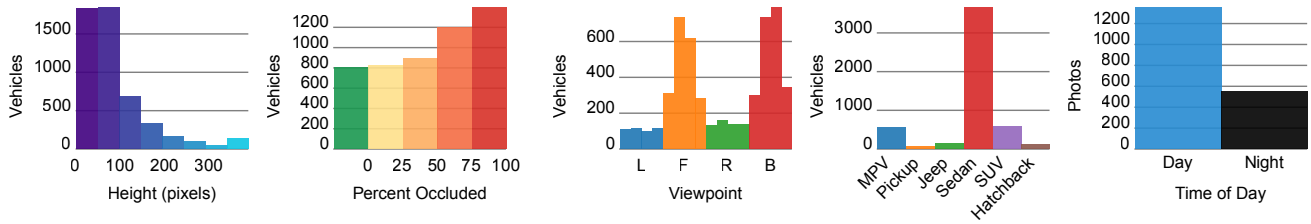

Figure 3. **NYC3DCars dataset statistics.** Several statistics over our dataset, including (left to right): histograms of (1) the height of each vehicle as measured in pixels, (2) the level of occlusion for each vehicle, (3) viewpoints (i.e. which side is visible to the camera), (4) vehicle types, and (5) the approximate time of day each photo was taken.

photographs as having no visible ground (i.e., pedestrians or other occluders cover the entire ground and obscure any potential vehicles), and 257 photographs as having some other defect such as extreme motion blur. Figure 3 highlights several key statistics of the data. To increase the generality of our dataset, we also gathered data for two other locations in NYC, Herald Square (130 images) and the area around the Apple Store on Fifth Avenue (489 images). These two locations are visually distinct from Times Square, but also offer many challenging instances of pedestrian and car occlusion. In the remainder of the paper, we focus on the Times Square dataset. For evaluating detection, we divide the images into equal-sized training and test sets.

3.3. Geographic Data

One reason we selected NYC as the location of our dataset was to leverage high-fidelity geographic data made freely available online. Every four years (most recently in 2009), NYC releases a free, updated set of polygons spanning the city, representing roadbeds, sidewalks, building footprints, medians, and road centerlines. We incorporate this data into our dataset, and augment roadbed polygons with road orientation information (i.e., the expected direction of traffic) from OpenStreetMap. While such comprehensive data is currently available for a small number of cities, adding such geographic information to our dataset allows researchers to study how this data can be used, so as to guide its use in other locations as more data becomes available.

The geographic data used in our dataset is illustrated in Figure 4. As shown in the figure, the fact that our photos are georegistered allows us to project this data into each photo. This results in pixel-level segmentations of surface types (e.g., road, sidewalk) and 3D buildings. We explore the use of such data for vehicle detection in Section 5.

Discussion. We were initially unsure how well SfM methods could be applied to photos of Times Square, due to its dynamic nature (with moving objects, such as cars and people, and changes in the scenery itself, e.g. through electronic displays and billboards). However, as the visualization in Figure 1 suggests, the reconstructed cameras largely align with sidewalks and other areas where one would expect photos to be captured. Despite the dynamic elements, enough of the background stays the same for many (but not all) images to be registered together correctly. The images that were incorrectly registered were generally photos with only a few matches to 3D points in the scene, but occasionally contained street signs or other confusing features.

Our dataset has inherent bias in that all photos are from a common geographic area; one way this bias manifests itself is in the skewed distributions of vehicle poses and types evident in Figure 3. We discuss bias in Section 7.

4. Viewpoint-Aware Vehicle Detection

In the following sections, we propose several methods and experiments that we apply to the NYC3DCars dataset. Here we discuss how we run existing learning methods on our data, and adapt them to build a set of baseline viewpoint-aware car detectors. As baselines, we consider several detector variants, and several types of data for training (including our own data, as well as non-viewpoint-annotated data from PASCAL VOC). Broadly speaking, our detectors start by training a state-of-the-art vehicle detector in a viewpoint-aware way, then apply these to new images with a non-maxima suppression step to produce a set of candidate detections.

Viewpoint-aware detector. Our detector is based on the successful Deformable Part Model (DPM) [5]. We first divide our viewpoint-annotated training examples into 16 uniformly spaced bins based on viewpoint azimuthal angle, and train a detector for each (as in [7]). For each bin, we train a linear SVM on HOG features, using unoccluded vehicles as positives, and negative examples mined from images that do not contain cars. Following the work of Girshick *et al*. [9], we use these linear SVMs as the initial filter coefficients in a car detection DPM mixture model. In particular, each viewpoint bin becomes a component in the mixture model.

In order to train this mixture model, we use the Weak-Label Structural SVM (WL-SSVM) framework proposed in [9]. The WL-SSVM is a generalization of the Latent SSVM where a structured loss can be applied between (1) a set of labels and (2) a set of predictions when the two sets might not be the same. Let $y = (y^l, y^b, y^v)$ be an annotation with class label $y^l \in \{-1, +1\}$, 2D bounding box y^b, and viewpoint bin $y^v \in \{1, \ldots, K, \varnothing\}$ (where K is the number of viewpoint bins, and $\varnothing$ indicates that a viewpoint

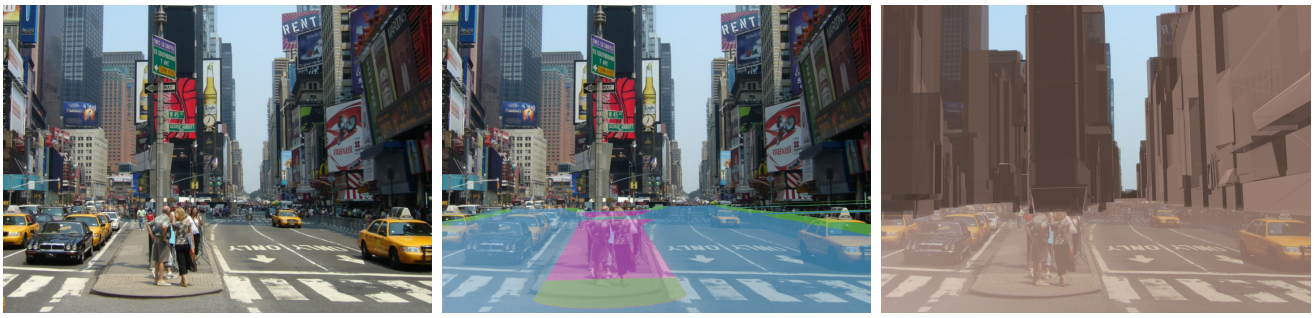

Figure 4. **3D visualization of geographic data.** By exploiting georegistration, we can obtain pixel-level labels. Left: Input image. Center: Image with vector data: roads (blue), medians (violet), and sidewalks (green). Right: Image with a 3D model of Times Square.

is unavailable, e.g. with VOC data). Similarly, let $s = (s^l, s^b, s^v)$ be a prediction with class label s^l, 2D bounding box s^b, and viewpoint s^v. We use the WL-SSVM over the more traditional latent SVM in order to penalize true positive class detections with incorrect viewpoint classification as well as to handle training examples without a viewpoint annotation (e.g., if we include VOC data for training).

The WL-SSVM requires two loss functions to be defined, $L_{\text{margin}}(y, s)$ which is used to push bad prediction scores down and $L_{\text{output}}(y, s)$ which is used to push good prediction scores up. (We provide a review of [9] with more details in the supplemental material.) We define our viewpoint-aware structured loss function, $L_{\text{view},l,\tau}(y, s)$, as

$$L_{l,\tau}(y,s) = \begin{cases} l & y^l = -1 \wedge s^l = +1 \\ 0 & y^l = -1 \wedge s^l = -1 \\ l & y^l = +1 \wedge \text{overlap}(y^b, s^b) < \tau \\ 0 & y^l = +1 \wedge \text{overlap}(y^b, s^b) \geq \tau \end{cases} \tag{1}$$

$$L_{\text{view},l,\tau}(y,s) = \begin{cases} \frac{1}{2}\left[l + L_{l,\tau}(y,s)\right] & y^v \neq s^v \wedge y^v \neq \varnothing \\ \frac{1}{2} L_{l,\tau}(y,s) & y^v = s^v \wedge y^v \neq \varnothing \\ L_{l,\tau}(y,s) & y^v = \varnothing \end{cases} \tag{2}$$

where $\text{overlap}(y^b, s^b) = \frac{\|y^b \cap s^b\|}{\|y^b \cup s^b\|}$. We use $L_{\text{margin}}(y, s) = L_{\text{view},1,0.5}(y, s)$ and $L_{\text{output}}(y, s) = L_{\text{view},\infty,0.7}(y, s)$. This formulation is similar in spirit to a model proposed by [20], but differs in that we do not encode the precise bounding box overlap in the loss function.

An alternative approach to training this viewpoint model is to use a latent SVM (LSVM) as formulated in [5]. In order to accommodate the viewpoint annotations, it is desirable to only choose compatible latent variable assignments in the "relabel positive examples" step of the coordinate descent approach. However, unlike the WL-SSVM formulation, this fails to apply a loss to true positive car detections with incorrect viewpoint classifications.

Non-maxima Suppression (NMS). We run the trained detector as a sliding window and threshold. We now take these (possibly overlapping) detections, and create a final set of detections via NMS. As in [18], we greedily select the top scoring detection not yet selected or removed, then remove all other detections whose overlap with the selected detection is greater than a threshold (we use a threshold of 0.3 on the ratio of bounding box intersection area over union area).

5. Geographic Context Rescoring

Given that our dataset contains geographic context, such as road boundaries and sidewalks, a natural question is how useful this kind of geographic information is for recognition tasks (e.g., in reducing false positive detections, or in improving detected 3D vehicle pose estimates). We now describe a method for utilizing this data as a "stage" on which we can reason about positions and orientations of objects.

Let (ϕ, λ) denote a (latitude, longitude) position on the Earth's surface. We reason about both the world and detected vehicles in this coordinate system. We define $w_e(\phi, \lambda)$ as the terrain elevation at (ϕ, λ); W_{r_i} as a single roadbed polygon with traffic-direction vectors $\vec{w_{d_i}}(\phi, \lambda)$ defined at each point in the polygon (we model road intersections as overlapping polygons, each with its own direction of travel); and $W_r = \cup_i W_{r_i}$ as the set of all roadbed surfaces.

For each 2D detection produced by our baseline detector, we reason about its plausibility given the provided geographic data. To do so, we convert the 2D detection into a set of hypothesis 3D car poses, each placed inside the physical scene so as to fit the 2D bounding box. We parameterize a 3D vehicle hypothesis as $\vec{v} = (v_\phi, v_\lambda, v_\theta, v_e)$ where (v_ϕ, v_λ) is the 2D ground position of the vehicle's centroid, v_θ is the vehicle heading, and v_e is the elevation of the bottom of the vehicle. Let $\vec{v}_f$ be the 2D vehicle footprint on the ground. We assume that the car is rotated only about the scene up vector, but do not assume the car is strictly resting on the ground, in order to account for 2D localization error.

To generate a set of 3D hypotheses from a 2D detection, we begin with a small database of example 3D vehicle CAD models of different types (e.g., sedans, SUVs, etc.), to account for different possible shapes and sizes of the detected vehicle. For each example 3D model, we place it in the scene

so that it matches (1) the viewpoint predicted by the detector, and (2) the 2D bounding box of the detection. The result is a set of 3D hypotheses, $V = \{\vec{v}_k\}$. More details about this placement step are provided in the supplemental material.

For each 3D hypothesis $\vec{v} \in V$, we rescore the detection using three geographic cues: an **elevation** score, an **orientation** score, and a **road coverage** score. The elevation score favors detections that are close to the ground, the orientation score encourages detections that have a plausible orientation given where it is on the road network (e.g., going the correct direction down a one-way street), and the road coverage score encourages detections that lie on the road. This idea is related to the scene-based reasoning of Hoiem, *et al*. [15], but using known geographic data.

The elevation score S_E is defined in terms the height of the 3D car's wheels above the ground:

$$S_E(\vec{v}) = \exp\left[-\frac{(v_e - w_e(v_\phi, v_\lambda))^2}{2\sigma_e^2}\right] \tag{3}$$

We use $\sigma_e = 0.5$ in our work.

Next, using roadbed polygons from our geographic data, we compute the percentage of the car's footprint $\vec{v}_f$ that intersects the roadbed. This is our road coverage score, S_C.

$$S_C(\vec{v}) = \frac{\|\vec{v}_f \cap W_r\|}{\|\vec{v}_f\|} \tag{4}$$

Finally, we find the roadbed polygons that the car's footprint intersects, along with their associated directions of travel. (The car might overlap with multiple road polygons if it is in an intersection.) Among these possible local directions of travel, we find the one that most closely agrees with the car's predicted orientation and penalize deviations from this orientation. Because our detector produces discretized orientations, we penalize based on whether the road-predicted orientation falls within the same bin as the vehicle:

$$\Delta_d = \min_{d_i} \arccos(\vec{v_d} \cdot \vec{w_{d_i}}(v_\phi, v_\lambda)) \tag{5}$$

$$S_O(\vec{v}) = \begin{cases} 1.0 & \Delta_d < \frac{w}{2} \\ 0.5 & \frac{w}{2} \leq \Delta_d < \frac{3w}{2} \\ 0.0 & \frac{3w}{2} \leq \Delta_d \end{cases} \tag{6}$$

where w is the viewpoint bin width (22.5 degrees in the case of our 16 bin detector). This is the orientation score.

In order to calibrate the DPM detector output to be compatible with the rest of our pipeline, we turn raw detection scores into scores between 0 and 1 using Platt calibration [21] with VOC2007 *test* as the validation set. The final detection score, $S(\vec{v})$ is defined as

$$S(\vec{v}) = S_E(\vec{v}) S_C(\vec{v}) S_O(\vec{v}) S_V(\vec{v}) \tag{7}$$

where the visual score, $S_V(\vec{v})$, is the posterior of the DPM detection. These rescored detections are then fed into the NMS procedure to produce a set of output detections.

6. Experiments and Results

In our experiments we use the widely employed VOC overlap criterion to evaluate detections. A detection is considered a true positive if $\frac{\|s^b \cap gt^b\|}{\|s^b \cup gt^b\|} \geq 0.5$ where s^b is the predicted bounding box as in Section 4 and gt^b is the ground truth 2D bounding box. Only the first overlapping bounding box is considered a true positive. We also evaluate accuracy in recognizing viewpoint; we use *orientation-similarity* (OS), presented in [7] as a way to evaluate orientation estimation. OS can be thought of as a precision weighted by a normalized cosine similarity between detection and annotation poses, and is therefore bounded above by precision.

6.1. Baseline Detectors

To get a sense for how NYC3DCars compares to established datasets in terms of difficulty, we begin by evaluating a set of DPM training methods applied to different combinations of datasets, including NYC3DCars *train-TimesSquare*, KITTI *train*, and VOC2007 *train+val*. We extended **voc-release5** [8] software package for all experiments. We compare three models: (a) a 6-component, unsupervised mixture model with left-right mirroring, (b) a 16-component viewpoint-aware detector using the LSVM with $L_{\text{output}}(y, s)$ to restrict the set of valid latent variable assignments, and (c) a 16-component viewpoint-aware detector using the WL-SSVM with both L_{margin} and L_{output}. For (a), we provide all positive examples during initialization, but for the viewpoint-aware DPMs ((b) and (c)) we provide only viewpoint-annotated examples during initialization and then introduce other examples, if applicable, after the mixture model has been built. The same negative training set is used for all models, VOC2007 *train+val* car-free images, so as to focus on comparing positive training sets and models.

Table 1 presents average precision and average orientation similarity. The top row shows the training data used, and results are shown for testing on both VOC2007 and our test data. We note a significant decrease in average precision (AP) as we move from the unsupervised DPM to the viewpoint-aware models. This is consistent with previous work, where the unsupervised DPM performs better at raw detection [12]. We also see little to no improvement between the LSVM and WL-SSVM when tested on VOC2007. However, a slight increase in AP and a significant increase in average orientation-similarity (AOS) is seen when using the WL-SSVM over the LSVM for NYC3DCars. When trained on NYC3DCars alone, the WL-SSVM offers no benefit over the LSVM. We also trained the NYC3DCars methods using NYC3DCars *train-TimesSquare* car-free images for the negative set and found that for the unsupervised DPM method AP remained roughly the same, whereas the LSVM and WL-SSVM AP both decreased to **0.377** when tested on VOC2007 and both increased to **0.460** when tested on NYC3DCars.

	KITTI			VOC2007+KITTI			NYC3DCars			VOC2007+KITTI+NYC3DCars		
	[5]	VP LSVM	VP WL-SSVM	[5]	VP LSVM	VP WL-SSVM	[5]	VP LSVM	VP WL-SSVM	[5]	VP LSVM	VP WL-SSVM
VOC2007	0.420	0.404	0.406	0.532	0.474	0.481	0.426	0.381	0.381	0.516	0.482	0.487
NYC3DCars	0.395	0.371/0.302	0.378/0.338	0.456	0.406/0.324	0.413/0.372	0.511	0.450/0.416	0.450/0.417	0.513	0.448/0.389	0.455/0.426

Table 1. Average precision (AP) and average orientation similarity (AOS) for several training methods and positive training sets (top) evaluated on two test sets (left). For methods or test sets that do not provide viewpoint predictions, only AP is shown. Otherwise, both AP and AOS are shown (as AP/AOS).

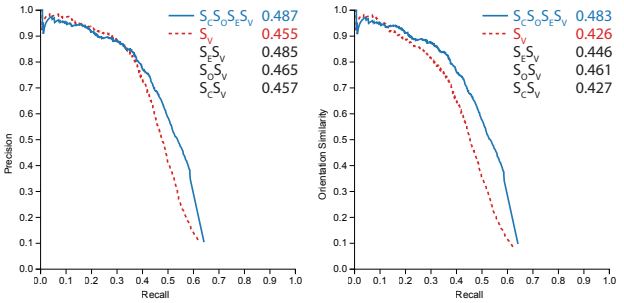

Figure 5. Precision-Recall and Orientation Similarity-Recall plots for geographic context-rescored detections. S_V is the visual score, S_C is the coverage score, S_O is the orientation score, and S_E is the elevation score. Two curves are shown: one for a method that uses only visual features S_V, and one that uses all available features ($S_C S_O S_E S_V$). Average precision and average orientation similarity scores for each method are shown in the upper right corner of each plot.

We also note that the best-performing methods when testing on our data also require our data for training, which accords with the varying biases across datasets. The best performing pose-sensitive method on our data achieves an AP/AOS of **0.455/0.426**. Note that training on KITTI alone performed relatively poorly, and even adding VOC2007 training data improved results above training on KITTI alone. We believe this is due to NYC3DCars having greater variety than KITTI, as our data is drawn from the Internet.

6.2. Geography-Aware Detection

We compare detection results before and after the introduction of specific types of geographic context. As our baseline, we selected our top performing viewpoint-aware detector from Section 4, the VOC2007+KITTI+NYC3DCars-trained WL-SSVM DPM.

Figure 5 shows precision-recall and orientation similarity-recall curves for several combinations of geographic context scores. By incorporating geographic context into our system, we are able to raise AP from **0.455** to **0.487** and AOS from **0.426** to **0.483**. In terms of AP, we found no significant improvement by including the coverage score. However, both the elevation and orientation scores showed improvement with the elevation score providing the greatest contribution. In terms of orientation estimation, not surprisingly, the orientation score proves to be a useful prior, but the elevation score also helps to improve AOS (as it improves AP).

We also tried using the horizon estimate from SfM to cull detections, which did offer improvement over the baseline, but achieved only **0.470** AP and **0.438** AOS, a smaller increase than with the geographic information. We also used the geometric context framework of [15] and replaced their detector with our own. While this sort of system has offered improvement in the past, we had difficulty getting it to provide competitive output for our modern detector. Often, low scoring false positives in plausible locations were significantly boosted making it a challenge to configure it to perform as well as the baseline.

In terms of 3D vehicle position estimation, we found that before geographic context rescoring we had a mean absolute ground plane translation error of **3.932 meters** and a mean absolute elevation error of **0.452 meters** for a set of detetions with high precision (0.9). After geographic context rescoring, this mean absolute error was reduced to **3.392 meters** and **0.201 meters** respectively.

7. Discussion and conclusion

We conclude with a discussion of limitations, as well as ideas for additional research directions our data can enable.

Data complexity. One potential concern with our dataset is that it involves a more involved construction—involving SfM, georegistration, etc.—than traditional datasets. While it this construction process is more complex, it relies on reconstruction algorithms that are starting to mature. Similarly, the geographic data we use is becoming more widespread all the time. Moreover, our approach allows for study of how useful this additional data is in vision, as a guide for future datasets and methods.

Bias. Our dataset covers a limited geographic area, and is biased towards vehicles that commonly appear in NYC (especially taxis, and sedans in general), and viewpoints that are accessible to photographers in this area (see Figure 3). Another, more subtle form of bias is that it is restricted to images that can be registered using SfM. For some applications, this bias is a limitation—for instance, a detector built from our data might not work well in a completely different application. However, these biases are also a strength if the goal is to build a region-specific detector, e.g., for applying detection in future Internet photos of Times Square.

Other limitations. The cameras recovered by our SfM procedure are not perfect, and have some noise as well as occasional gross failures. One future direction is to leverage detection methods in a feedback loop to improve geometry, related to prior methods but with multiple photos taken over time [2]. We also plan to study how errors in camera pose

Figure 6. **Vehicle-vehicle occlusions.** Left: User annotations. Right: Visibility masks. Inset: Occlusion map for one annotation (red: unoccluded; white: occluded.)

affect detection results. Currently, our dataset only contains cars; in the future, we plan to also annotate pedestrians and service vehicles (e.g., firetrucks). Finally, our annotation tool assumes that objects are supported by a ground plane. This assumption could be relaxed (so that it could handle hilly areas) by incorporating open source terrain models.

Future directions. We present initial experiments that leverage our data for pose-sensitive vehicle detection, with promising results. However, we believe that our dataset enables rich areas for future study; we describe a few ideas here. First, one could use our data, along with multi-view stereo methods, to build a detailed model of the background, and use this in a detection method to predict background appearance. Second, because our dataset is inherently 3D, it allows for reasoning about 3D relationships of objects. For instance, it could drive new non-maxima suppression methods that reason about object overlap in 3D (world-space) rather than 2D (image-space), or methods that explicitly reason about occlusions between 3D objects placed in the scene. Our data also can be used to created detailed vehicle-vehicle occlusion maps as shown in Figure 6. These maps could be used to learn common patterns for occlusion, for instance by extending mixture models to model occluded parts [9]. Finally, a longer-term application of our method is in using Internet photos as a source of data for traffic prediction and other problems in urban understanding. For instance, one could imagine tracking taxis through NYC by applying our methods continuously to Internet photos uploaded over time.

Acknowledgements. This work was supported in part by the National Science Foundation under grant NSF-0931686, and by grants from the Intel Science and Technology Center for Visual Computing and Amazon Web Services in Education.

References

[1] S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, and R. Szeliski. Building Rome in a day. In *ICCV*, 2009.

[2] S. Bao, M. Bagra, Y.-W. Chao, and S. Savarese. Semantic structure from motion with points, regions, and objects. In *CVPR*, 2012.

[3] N. Cornelis, B. Leibe, K. Cornelis, and L. J. V. Gool. 3D urban scene modeling integrating recognition and reconstruction. *IJCV*, 78(2-3):121–141, 2008.

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

[5] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *PAMI*, 32, 2010.

[6] D. F. Fouhey, V. Delaitre, A. Gupta, A. A. Efros, I. Laptev, and J. Sivic. People watching: Human actions as a cue for single-view geometry. In *ECCV*, 2012.

[7] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *CVPR*, 2012.

[8] R. B. Girshick, P. F. Felzenszwalb, and D. McAllester. Discriminatively trained deformable part models, release 5. http://people.cs.uchicago.edu/˜rbg/latent-release5/.

[9] R. B. Girshick, P. F. Felzenszwalb, and D. A. McAllester. Object detection with grammar models. In *NIPS*, 2011.

[10] D. Glasner, M. Galun, S. Alpert, R. Basri, and G. Shakhnarovich. Viewpoint-aware object detection and pose estimation. In *ICCV*, 2011.

[11] J. Hays and A. Efros. IM2GPS: Estimating geographic information from a single image. In *CVPR*, 2008.

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

[13] M. Hejrati and D. Ramanan. Analyzing 3d objects in cluttered images. In *NIPS*, 2012.

[14] D. Hoiem, A. Efros, and M. Hebert. Geometric context from a single image. In *ICCV*, 2005.

[15] D. Hoiem, A. Efros, and M. Hebert. Putting objects in perspective. In *CVPR*, 2006.

[16] Y. Li, N. Snavely, D. Huttenlocher, and P. Fua. Worldwide pose estimation using 3D point clouds. In *ECCV*, 2012.

[17] J. Little, A. Abrams, and R. Pless. Tools for richer crowd source image annotations. In *WACV*, 2012.

[18] T. Malisiewicz, A. Gupta, and A. Efros. Ensemble of exemplar-SVMs for object detection and beyond. In *ICCV*, 2011.

[19] M. Ozuysal, V. Lepetit, and P. Fua. Pose estimation for category specific multiview object localization. In *CVPR*, 2009.

[20] B. Pepik, M. Stark, P. Gehler, and B. Schiele. Teaching 3D geometry to deformable part models. In *CVPR*, 2012.

[21] J. C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*. MIT Press, 1999.

[22] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. LabelMe: A database and web-based tool for image annotation. *IJCV*, 77(1-3), 2008.

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

[24] M. Sun, S. Y.-Z. Bao, and S. Savarese. Object detection using geometrical context feedback. *IJCV*, 100(2), 2012.