Putting Objects in Perspective

Derek Hoiem Alexei A. Efros Martial Hebert
Carnegie Mellon University, Robotics Institute
{dhoiem,efros,hebert}@cs.cmu.edu

Abstract

Image understanding requires not only individually estimating elements of the visual world but also capturing the interplay among them. In this paper, we provide a framework for placing local object detection in the context of the overall 3D scene by modeling the interdependence of objects, surface orientations, and camera viewpoint. Most object detection methods consider all scales and locations in the image as equally likely. We show that with probabilistic estimates of 3D geometry, both in terms of surfaces and world coordinates, we can put objects into perspective and model the scale and location variance in the image. Our approach reflects the cyclical nature of the problem by allowing probabilistic object hypotheses to refine geometry and vice-versa. Our framework allows painless substitution of almost any object detector and is easily extended to include other aspects of image understanding. Our results confirm the benefits of our integrated approach.

1. Introduction

Consider the street scene depicted on Figure 1. Most people will have little trouble seeing that the green box in the middle contains a car. This is despite the fact that, shown in isolation, these same pixels can just as easily be interpreted as a person's shoulder, a mouse, a stack of books, a balcony, or a million other things! Yet, when we look at the entire scene, all ambiguity is resolved – the car is unmistakably a car. How do we do this?

There is strong psychophysical evidence (e.g. [3, 30]) that context plays a crucial role in scene understanding. In our example, the car-like blob is recognized as a car because: 1) it's sitting on the road, and 2) it's the "right" size, relative to other objects in the scene (cars, buildings, pedestrians, etc). Of course, the trouble is that everything is tightly interconnected – a visual object that uses others as its context will, in turn, be used as context by these other objects. We recognize a car because it's on the road. But how do we recognize a road? – because there are cars! How does one attack this chicken-and-egg problem? What is the right framework for connecting all these pieces of the recognition puzzle in a coherent and tractable manner?

In this paper we will propose a unified approach for modeling the contextual symbiosis between three crucial elements required for scene understanding: low-level object detectors, rough 3D scene geometry, and approximate camera position/orientation. Our main insight is to model the contextual relationships between the visual elements, *not in the 2D image plane* where they have been projected by the camera, but *within the 3D world* where they actually reside. Perspective projection obscures the relationships that are present in the actual scene: a nearby car will appear much bigger than a car far away, even though in reality they are the same height. We "undo" the perspective projection and analyze the objects in the space of the 3D scene.

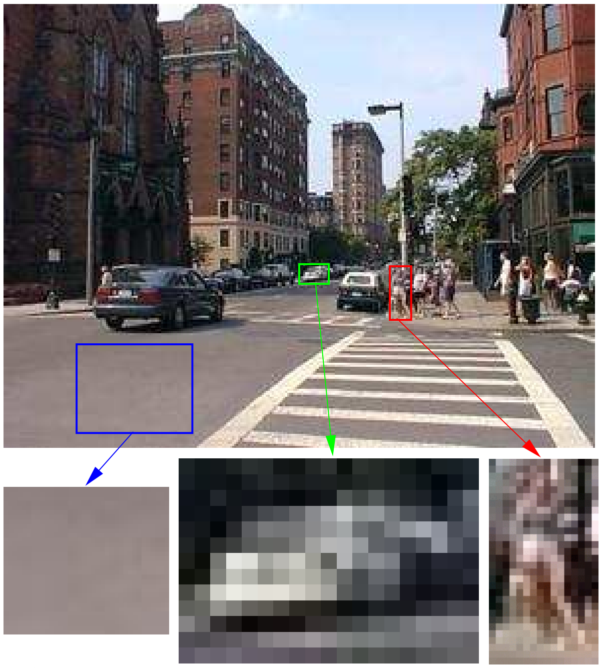


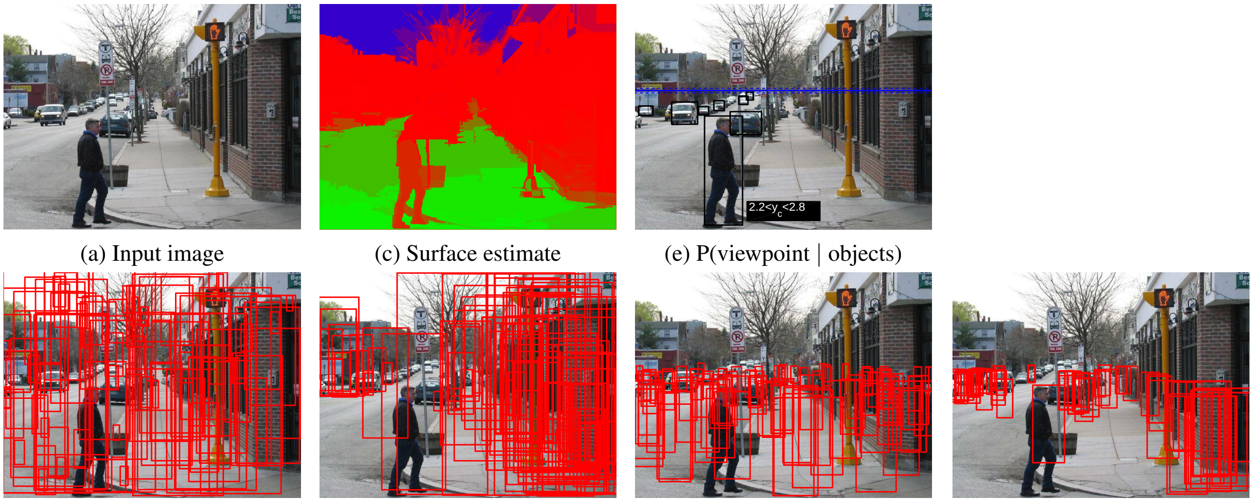
Figure 1. General object recognition cannot be solved locally, but requires the interpretation of the entire image. In the above image, it's virtually impossible to recognize the car, the person and the road in isolation, but taken together they form a coherent visual story.

1.1. Background

In its early days, computer vision had but a single grand goal: to provide a complete semantic interpretation of an input image by reasoning about the 3D scene that generated it. Indeed, by the late 1970s there were several image understanding systems being developed, including such pioneering work as Brooks' *ACRONYM* [4], Hanson and Riseman's *VISIONS* [10], Ohta and Kanade's outdoor scene understanding system [23], Barrow and Tenenbaum's intrinsic images [2], etc. For example, *VISIONS* was an extremely ambitious system that analyzed a scene on many interrelated levels including segments, 3D surfaces and volumes, objects, and scene categories. However, because of the heavy use of heuristics, none of these early systems were particularly successful, which led people to doubt the very goal of complete image understanding.

We believe that the vision pioneers were simply ahead of their time. They had no choice but to rely on heuristics because they lacked the computational resources to *learn* the relationships governing the structure of our visual world. The advancement of learning methods in the last decade brings renewed hope for a complete image understanding solution. However, the currently popular learning approaches are based on looking at small image windows at all locations and scales to find specific objects. This works wonderfully for face detection [28, 33] (since the inside of a face is much more important than the boundary) but is quite unreliable for other types of objects, such as cars and pedestrians, especially at the smaller scales.

As a result, several researchers have recently begun to consider the use of contextual information for object detection. The main focus has been on modeling direct relationships between objects and other objects [17, 21], regions [11, 18, 32] or scene categories [21, 29], all within the 2D image. Going beyond the 2D image plane, Hoiem *et al.* [12] propose a mechanism for estimating rough 3D scene geometry from a single image and use this information as additional features to improve object detection. From low-level image cues, Torralba and Sinha [31] get a sense of the viewpoint and mean scene depth, which provides a useful prior for object detection. Forsyth *et al.* [8] describe a method for geometric consistency of object hypotheses



(a) Input image (c) Surface estimate (e) P(viewpoint | objects)

(b) P(person) = uniform (d) P(person | geometry) (f) P(person | viewpoint) (g) P(person|viewpoint,geometry)

Figure 2. Watch for pedestrians! In (b,d,f,g), we show 100 boxes sampled according to the available information. Given an input image (a), a local object detector will expect to find a pedestrian at any location/scale (b). However, given an estimate of rough surface orientations (c), we can better predict where a pedestrian is likely to be (d). We can estimate the camera viewpoint (e) from a few known objects in the image. Conversely, knowing the camera viewpoint can help in predict the likely scale of a pedestrian (f). The combined evidence from surface geometry and camera viewpoint provides a powerful predictor of where a pedestrian might be (g), before we even run a pedestrian detector! Red, green, and blue channels of (c) indicate confidence in vertical, ground, and sky, respectively.

in simple scenes using hard algebraic constraints. Others have also modeled the relationship between the camera parameters and objects, requiring either a well-calibrated camera (e.g. [14]), a stationary surveillance camera (e.g. [16]), or both [9].

In this work, we draw on several of the previous techniques: local object detection [21, 7], 3D scene geometry estimation [12], and camera viewpoint estimation. Our contribution is a statistical framework that allows *simultaneous* inference of object identities, surface orientations, and camera viewpoint using a *single image* taken from an uncalibrated camera.

1.2. Overview

To evaluate our approach, we have chosen a very challenging dataset of outdoor images [27] that contain cars and people, often partly occluded, over an extremely wide range of scales and in accidental poses (unlike, for example, the framed photographs in Corel or CalTech datasets). Our goal is to demonstrate that substantial improvement over standard low-level detectors can be obtained by reasoning about the underlying 3D scene structure.

One way to think about what we are trying to achieve is to consider the likely places in an image where an object (e.g. a pedestrian) could be found (Figure 2). Without considering the 3D structure of the scene, all image positions and scales are equally likely (Figure 2b) – this is what most object detectors assume. But if we can estimate the rough surface geometry in the scene, this information can be used to adjust the probability of finding a pedestrian at a given image location (Figure 2d). Likewise, having an estimate of the camera viewpoint (height and horizon position) supplies the likely scale of an object in the image (Figure 2f). Combining these two geometric cues together gives us a rather tight prior likelihood for the location and scale of a pedestrian, as in Figure 2g. This example is particularly interesting because this is still only a prior – we have not applied a pedestrian detector yet. Notice, as well, that the pattern of expected

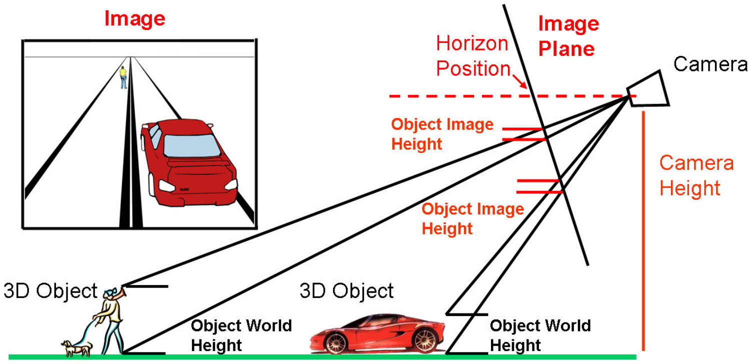


Figure 3. An object's height in the image can be determined from its height in the world and the viewpoint.

pedestrian detections is reminiscent of typical human eye-tracking experiments, where subjects are asked to search for a person in an image.

Of course, just as scene and camera geometry can influence object detection, so can the detected objects alter the geometry estimation. For example, if we know the locations/scales of some of the objects in the image, we can use this to better estimate the camera viewpoint parameters (see the 90% confidence bounds in Figure 2e). In general, our aim is to combine all these pieces of evidence into a single coherent image interpretation framework.

The rest of the paper will be devoted to exploring our two primary conjectures: 1) 3D reasoning improves object detection, even when using a single image from an uncalibrated camera, and 2) the more fully the scene is modeled (more properties, more objects), the better the estimates will be. We will first describe the mathematics of projective geometry as it relates to our problem (Section 2). We will then define the probabilistic model used for describing the relationships within the 3D scene (Section 3) and how it can be learned (Section 4). In Section 5, we present quantitative and qualitative results demonstrating the performance of our system on a difficult dataset. Finally, in Section 6, we demonstrate a new technique for estimating camera viewpoint and show that it leads to improved accuracy in object detection.

This article is an extension of our earlier work [13]. The primary new contributions are an expanded derivation and discussion of the object-viewpoint relationship (Section 2) and a new algorithm for recovering camera viewpoint based on image matching (Section 6). We also show that our integrated scene model outperforms the use of a simpler object position/size prior (Section 5).

2. Scene Projection

We assume that all objects of interest rest on the ground plane. While this assumption may seem restrictive (cannot find people on the rooftops), humans seem to make the same assumption (we fail to notice the security standing on the rooftops at political rallies unless we specifically look for them).

Under this assumption, knowing only the camera height and horizon line, we can estimate a grounded object's height in the scene from its top and bottom position in the image (see Figure 3). We will now derive this relationship using the following notation: pixel coordinates (u, v) ranging from (0,0) at the bottom-left to (1,1) at the top-right; world coordinates (x, y, z) with y being height and z being depth; camera tilt θ_x; focal length f; camera optical center (u_c, v_c); and camera height y_c. By convention, the world coordinates are defined by $z_c = 0$, $x_c = 0$, and the ground plane at $y = 0$. We assume zero roll (or that the image has been rotated to account for roll) and define the horizon position v_0 as the vanishing line of the ground plane in image coordinates. In these coordinates, camera tilt (in radians) is given by $\theta_x = 2\arctan\frac{v_c-v_0}{2f}$. We use a perspective projection model with zero skew and unit aspect ratio.

Using homogeneous coordinates, the transformation from image coordinates to scene coordinates is given by

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \frac{1}{z} \begin{bmatrix} f & 0 & u_c \\ 0 & f & v_c \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x & y_c \\ 0 & \sin\theta_x & \cos\theta_x & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}. \tag{1}$$

From this, we can see that

$$y = \frac{z\left(f\sin\theta_x - (v_c - v)\cos\theta_x\right) - f y_c}{(v_c - v)\sin\theta_x + f\cos\theta_x}. \tag{2}$$

Now suppose that we are given the top and bottom position of an upright object (v_t and v_b, respectively). Since $y = 0$ at v_b, we can solve for object depth z:

$$z = \frac{f y_c}{f\sin\theta_x - (v_c - v_b)\cos\theta_x}. \tag{3}$$

From Equations 2 and 3, we can solve for object height y:

$$y = \frac{f y_c\left(f\sin\theta_x - (v_c - v_t)\cos\theta_x\right) / \left(f\sin\theta_x - (v_c - v_b)\cos\theta_x\right) - f y_c}{(v_c - v_t)\sin\theta_x + f\cos\theta_x}. \tag{4}$$

If the camera tilt is small (e.g., if the horizon position is within the image), we can greatly simplify this equation with the following approximations: $\cos\theta_x \approx 1$, $\sin\theta_x \approx \theta_x$, and $\theta_x \approx \frac{v_c - v_0}{f}$ yielding:

$$y \approx y_c \frac{v_t - v_b}{v_0 - v_b} / \left(1 + (v_c - v_0)(v_c - v_t)/f^2\right). \tag{5}$$

In our experiments, we approximate further: $(v_c - v_0)(v_c - v_t)/f^2 \approx 0$, giving us

$$y \approx y_c \frac{v_t - v_b}{v_0 - v_b}. \tag{6}$$

How valid are these approximations? Equation 6 is exact when the camera is parallel to the ground plane (such that $\theta_x = 0$ and $v_0 = v_c$). Even when the camera is tilted, the approximation is very good for the following reasons: tilt tends to be small ($v_c - v_0 \approx 0$; $\theta_x \approx 0$); the tops of detected objects (pedestrians and cars in this paper) tend to be near the horizon position since the photograph is often taken by a person standing on the ground; and camera focal length f is usually greater than 1 for the defined coordinates ($f = 1.4$ times image height is typical). However, the approximation may be poor under the following conditions, listed roughly in order of descending importance: object is not resting on the ground; camera tilt is very large (e.g., overhead view); or image taken with a wide-angle lens (f is small). In practice, the approximation is sufficient to improve object detection (Section 5) and to accurately estimate object size (Section 6) in the LabelMe dataset.

To simplify the notation in the remainder of the paper, we will refer to the world height, image height, and bottom position in the image of object i as y_i, h_i, and v_i, respectively. As before, we denote horizon position v_0 and camera height y_c. Using this notation (illustrated in Figure 4), we have the following relationship:

$$y \approx y_c \frac{h_i}{v_0 - v_i}. \tag{7}$$

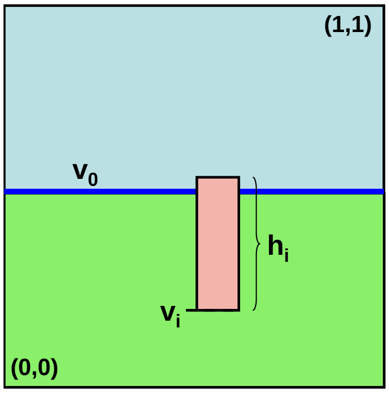


Figure 4. Illustration of horizon position v_0, object bottom position v_i, and object image height h_i. With these and camera height y_c, we can estimate object world height y_i using $y_i \approx \frac{h_i y_c}{v_o - v_i}$.

3. Modeling the Scene

We want to determine the viewpoint, object identities, and surface geometry of the scene from an image. We could estimate each independently, but our estimates will be much more accurate if we take advantage of the interactions between the scene elements. We consider the objects (e.g., cars, pedestrians, background) and geometric surfaces to each produce image evidence. The viewpoint, defined by the horizon position in the image and the camera height, directly affects the position and size of the objects in the image. In turn, the objects directly affect nearby geometric surfaces. We assume that local geometric surfaces are independent given their corresponding object identities and that the object identities are independent given the viewpoint. In Figure 5, we represent these conditional independence assumptions in a graphical model, denoting objects as $\mathbf{o}$, surface geometries as $\mathbf{g}$, object evidence as $\mathbf{e}_o$, geometry evidence as $\mathbf{e}_g$, and the viewpoint as θ.

Our model implies the following decomposition:

$$\mathrm{P}(\theta, \mathbf{o}, \mathbf{g}, \mathbf{e}_\mathrm{g}, \mathbf{e}_\mathrm{o}) = \mathrm{P}(\theta) \prod_\mathrm{i} \mathrm{P}(\mathrm{o_i}|\theta) \mathrm{P}(\mathbf{e}_\mathrm{oi}|\mathrm{o_i}) \mathrm{P}(\mathrm{g_i}|\mathrm{o_i}) \mathrm{P}(\mathbf{e}_\mathrm{gi}|\mathrm{g_i}). \tag{8}$$

We can use Bayes rule to give the likelihood of the scene conditioned on the image evidence:

$$\mathrm{P}(\theta, \mathbf{o}, \mathbf{g}|\mathbf{e}_\mathrm{g}, \mathbf{e}_\mathrm{o}) = \mathrm{P}(\theta) \prod_\mathrm{i} \mathrm{P}(\mathrm{o_i}|\theta) \frac{\mathrm{P}(\mathrm{o_i}|\mathbf{e}_\mathrm{oi})}{\mathrm{P}(\mathrm{o_i})} \frac{\mathrm{P}(\mathrm{g_i}|\mathbf{e}_\mathrm{gi})}{\mathrm{P}(\mathrm{g_i})}. \tag{9}$$

Our approach allows other researchers to easily integrate their own detectors into our framework. A classifier for a new object or an improved classifier for an existing one can be incorporated, using its probabilistic output for $P(o_i|\mathbf{e}_{oi})$. Each addition or improvement to the estimated likelihoods can then be used to improve the entire scene interpretation. This model does imply the assumption that the image evidence terms are independent, conditioned on the object or surface labels. While this assumption may be violated when objects are in close proximity, it provides modularity and facilitates inference.

3.1. Viewpoint

The viewpoint θ involves two variables: the horizon position in the image v_0 and the camera height (in meters) y_c. We consider camera height and horizon position to be independent *a priori* so that $P(\theta) = \mathrm{P}(\mathrm{v_0})\mathrm{P}(\mathrm{y_c})$. In our initial experiments (Sections 4 and 5), we model the horizon position likelihood with

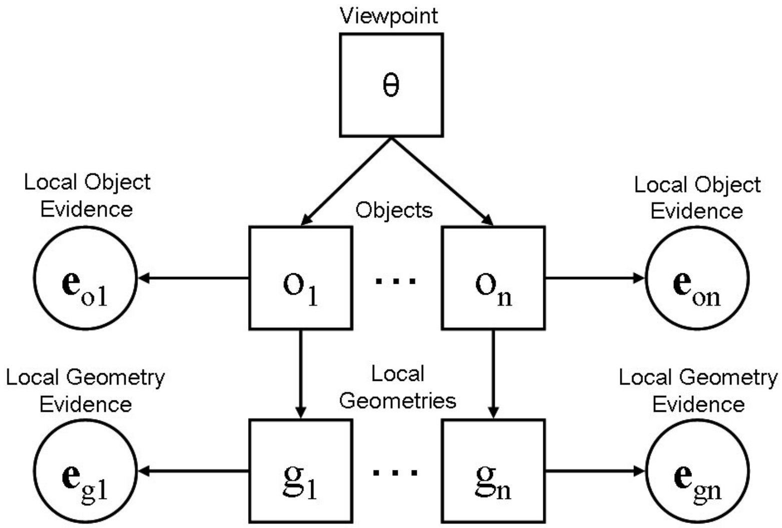


Figure 5. Graphical model of conditional independence for viewpoint θ, object identities $\mathbf{o}$, and the 3D geometry of surfaces $\mathbf{g}$ surrounding the objects. Viewpoint describes the horizon position in the image and the height of the camera in the 3D scene (in relation to the objects of interest). Each image has n object hypotheses, where n varies by image. The object hypothesis o_i involves assigning an identity (e.g., pedestrian or background) and a bounding box. The surface geometry g_i describes the 3D orientations of the i^{th} object surface and nearby surfaces in the scene.

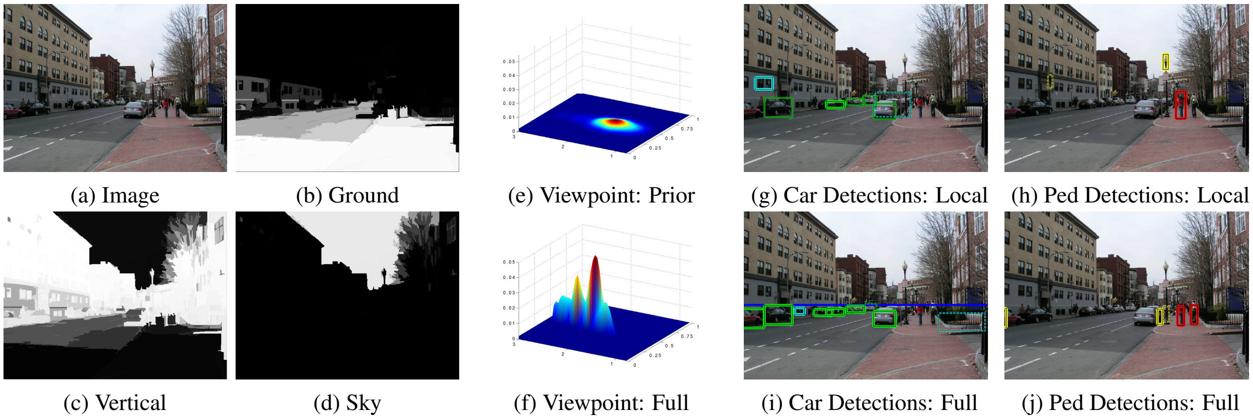

(a) Image
(b) Ground
(e) Viewpoint: Prior
(g) Car Detections: Local
(h) Ped Detections: Local
(c) Vertical
(d) Sky
(f) Viewpoint: Full
(i) Car Detections: Full
(j) Ped Detections: Full

Figure 6. We begin with geometry estimates (b,c,d), local object detection confidences (g,h), and a prior (e) on the viewpoint. Using our model, we improve our estimates of the viewpoint (f) and objects (i,j). In the viewpoint plots, the left axis is camera height (meters), and the right axis is horizon position (measured from the image bottom). The viewpoint peak likelihood increases from 0.0037 *a priori* to 0.0503 after inference. At roughly the same false positive (cars:cyan, peds:yellow) rate, the true detection (cars:green, peds:red) rate doubles when the scene is coherently modeled. Note that, though only detections above a threshold are shown, detections with lower confidence exist for both the local and full model.

a simple Gaussian prior. Similarly, for the camera height y_c, we estimate a prior distribution using kernel density estimation over the y_c values (computed based on objects of known height in the scene) in a set of training images. We will later (Section 6) show how to estimate the horizon position from image data directly, resulting in improved viewpoint estimation and object detection.

Figure 6 displays the viewpoint prior (e) and an example of the revised likelihood (f) when object and surface geometry evidences are considered. *A priori*, the most likely camera height is 1.67m, which happens to be eye level for a typical adult male, and the most likely horizon position is 0.50. While the viewpoint prior does have high variance, it is much more informative than the uniform distribution that is implicitly assumed when scale is considered irrelevant.

3.2. Objects

An object candidate o_i consists of a type $t_i \in \{object, background\}$ (e.g. "pedestrian") and a bounding box $bbox_i = \{u_i, v_i, w_i, h_i\}$ (lower-left coordinate, width, and height, respectively). The object term of our scene model is composed as follows:

$$\mathrm{P}(\mathrm{o_i}|\mathbf{e}_{\mathrm{oi}}, \theta) = \frac{\mathrm{P}(\mathrm{o_{oi}}|\mathbf{e}_{\mathrm{o}})}{\mathrm{P}(\mathrm{o_i})}\mathrm{P}(\mathrm{o_i}|\theta) \tag{10}$$

At each position and scale (with discrete steps) in the image, our window-based object detector outputs an estimate of the class-conditional log-likelihood ratio

$$c_i = \log \frac{\mathrm{P}(\mathbf{I}_\mathrm{i}|\mathrm{t_i} = \mathrm{obj}, \mathrm{bbox_i})}{\mathrm{P}(\mathbf{I}_\mathrm{i}|\mathrm{t_i} \neq \mathrm{obj}, \mathrm{bbox_i})} \tag{11}$$

based on local image information $\mathbf{I}_i$ at the i^{th} bounding box.[1] From these ratios and a prior $\mathrm{P}(\mathrm{o_i})$, we can compute the probability of an object occurring at a particular location/scale.

$$\mathrm{P}(\mathrm{t_i} = \mathrm{obj}, \mathrm{bbox_i}|\mathbf{I}_\mathrm{i}) = \frac{1}{1 + \exp[-\mathrm{c_i} - \log \frac{\mathrm{P}(\mathrm{o_i})}{1 - \mathrm{P}(\mathrm{o_i})}]} \tag{12}$$

Typically, researchers perform non-maxima suppression, assuming that high detection responses at neighboring positions could be due to an object at either of those positions (but not both). Making the same assumption, we also apply non-maxima suppression, but we form a point distribution out of the non-maxima, rather than discarding them. An object candidate is formed out of a group of closely overlapping bounding boxes.[2] The candidate's likelihood $\mathrm{P}(\mathrm{t_i} = \mathrm{obj}|\mathbf{e}_\mathrm{o})$ is equal to the likelihood of the highest-confidence bounding box, and the likelihoods of the locations given the object identity $\mathrm{P}(\mathrm{bbox_i}|\mathrm{t_i} = \mathrm{obj}, \mathbf{e}_\mathrm{o})$ are directly proportional to $\mathrm{P}(\mathrm{t_i} = \mathrm{obj}, \mathrm{bbox_i}|\mathbf{I})$. After thresholding to remove detections with very low confidences from consideration, a typical image will contain several dozen object candidates (determining n of Figure 5), each of which has tens to hundreds of possible position/shapes.

An object's height depends on its position when given the viewpoint. Formally, $\mathrm{P}(\mathrm{o_i}|\theta) \propto \mathrm{p}(\mathrm{h_i}|\mathrm{t_i}, \mathrm{v_i}, \theta)$ (the proportionality is due to the uniformity of $\mathrm{P}(\mathrm{t_i}, \mathrm{v_i}, \mathrm{w_i}|\theta)$). From Equation 7, if y_i is normal, with parameters $\{\mu_i, \sigma_i\}$, then h_i conditioned on $\{t_i, v_i, \theta\}$ is also normal, with parameters $\frac{\mu_i(v_0 - v_i)}{y_c}$ and $\frac{\sigma_i(v_0 - v_i)}{y_c}$.

3.3. Surface Geometry

Most objects of interest can be considered as vertical surfaces supported by the ground plane. Estimates of the local surface geometry could, therefore, provide additional evidence for objects. To obtain the rough 3D surface orientations in the image, we apply the method of [12] (we use the publicly available executable), which produces confidence maps for three main classes: "ground", "vertical", and "sky", and five subclasses of "vertical": planar, facing "left", "center", and "right", and non-planar "solid" and "porous". Figure 6(b,c,d) displays the confidence maps for the three main surface labels.

[1] To simplify notation, we omit parameter terms in likelihood estimates and do not distinguish between estimated likelihoods and true likelihoods.

[2] Each detector distinguishes between one object type and background in our implementation. Separate candidates are created for each type of object.

We define g_i to have three values corresponding to whether the object surface is visible in the detection window and, if so, whether the ground is visible just below the detection window. For example, we consider a car's geometric surface to be planar or non-planar solid and a pedestrian's surface to be non-planar solid. We can compute $\mathrm{P(g_i|o_i)}$ and $\mathrm{P(g_i)}$ by counting occurrences of each value of g_i in a training set. If o_i is background, we consider $\mathrm{P(g_i|o_i)} \approx \mathrm{P(g_i)}$. We estimate $\mathrm{P(g_i|\mathbf{e}_g)}$ based on the confidence maps of the geometric surfaces. In experiments, we found that the average geometric confidence in a window is a well-calibrated probability for the geometric value.

3.4. Inference

For tree-structured graphs like our model (Figure 5), Pearl's belief propagation algorithm [25] is optimal and very efficient. We simplify inference by quantizing continuous variables v_0 and y_c into evenly-spaced bins (50 and 100 bins, respectively). Our implementation makes use of the Bayes Net Toolbox [20]. After inference, we can pose queries, such as "What is the expected height of this object?" or "What are the marginal probabilities for cars?" or "What is the most probable explanation of the scene?". In this paper, we report results based on marginal probabilities from the sum-product algorithm (this allows an ROC curve to be computed). Figure 6 shows how local detections (g,h) improve when viewpoint and surface geometry are considered (i,j).

4. Training

Viewpoint. To estimate the priors for θ, we manually labeled the horizon in 60 outdoor images from the LabelMe database [27]. In each image, we labeled cars (including vans and trucks) and pedestrians (defined as an upright person) and computed the maximum likelihood estimate of the camera height based on the labeled horizon and the height distributions of cars and people in the world. We then estimated the prior for camera height using kernel density estimation (`ksdensity` in Matlab).

Objects. Our baseline car and pedestrian detector uses a method similar to the local detector of Murphy, Torralba, and Freeman [21]. We used the same local patch template features but added six color features that encode the average L*a*b* color of the detection window and the difference between the detection window and the surrounding area. The classifier uses a logistic regression version of Adaboost [5] to boost eight-node decision tree classifiers. For cars, we trained two views (front/back: 32x24 pixels and side: 40x16 pixels), and for pedestrians, we trained one view (16x40 pixels). Each were trained using the full PASCAL dataset [1].

To verify that our baseline detector has reasonable performance, we trained a car detector on the PASCAL challenge training/validation set, and evaluated the images in test set 1 using the criteria prescribed for the official competition. For the sake of comparison in this validation experiment, we did not search for cars shorter than 10% of the image height, since most of the official entries could not detect small cars. We obtain an average precision of 0.423 which is comparable to the best scores reported by the top 3 groups: 0.613, 0.489, and 0.353.

To estimate the height distribution of cars (in the 3D world), we used Consumer Reports (www.consumerreports.org) and, for pedestrians, used data from the National Center for Health Statistics (www.cdc.gov/nchs/). For cars, we estimated a mean of 1.59m and a standard deviation of 0.21m. For adult humans, the mean height is 1.7m with a standard deviation of 0.085m. In Section 6, we show how to automatically estimate distributions of camera viewpoint and object heights using an iterative EM-like algorithm on a standard object dataset.

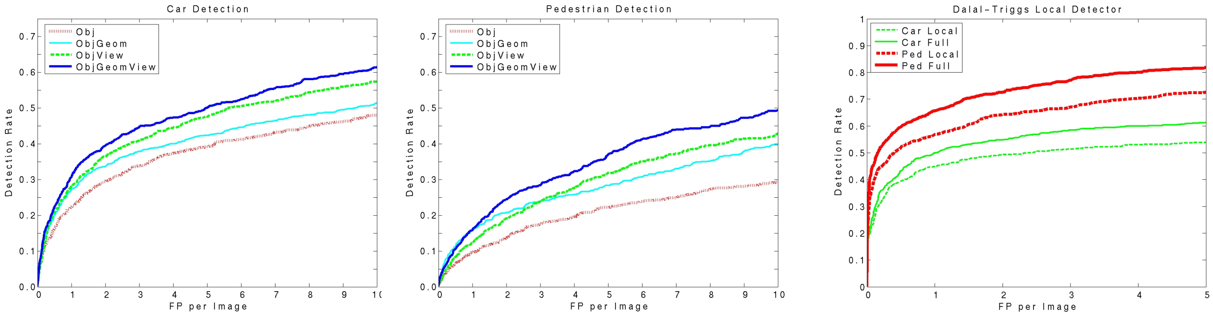


Figure 7. Considering viewpoint and surface geometry improves results over purely local object detection. The left two plots show object detection results using only local object evidence (Obj), object and geometry evidence (ObjGeom), objects related through the viewpoint (ObjView), and the full model (ObjViewGeom). On the right, we plot results using the Dalal-Triggs local detector [7].

	Cars			Pedestrians		
	1FP	5FP	10FP	1FP	5FP	10FP
+Geom	6.6%	5.6%	7.0%	7.5%	8.5%	17%
+View	8.2%	16%	22%	3.2%	14%	23%
+GeomView	**12%**	**22%**	**35%**	**7.2%**	**23 %**	**40%**

Table 1. Modeling viewpoint and surface geometry aids object detection. Shown are percentage reductions in the missed detection rate while fixing the number of false positives per image.

Surface Geometry. $\mathrm{P(g_i|o_i)}$ was found by counting the occurrences of the values of g_i for both people and cars in the 60 training images from LabelMe. We set $\mathrm{P(g_i)}$ to be uniform, because we found experimentally that learned values for $\mathrm{P(g_i)}$ resulted in the system over-relying on geometry. This over-reliance may be due to our labeled images (general outdoor) being drawn from a different distribution than our test set (streets of Boston) or to the lack of a modeled direct dependence between surface geometries.

5. Evaluation

Our test set consists of 422 random outdoor images from the LabelMe dataset [27]. The busy city streets, sidewalks, parking lots, and roads provide realistic environments for testing car and pedestrian detectors, and the wide variety of object pose and size and the frequency of occlusions make detection extremely challenging. In the dataset, 60 images have no cars or pedestrians, 44 have only pedestrians, 94 have only cars, and 224 have both cars and pedestrians. In total, the images contain 923 cars and 720 pedestrians.

We detect cars with heights as small as 14 pixels and pedestrians as small as 36 pixels tall. To get detection confidences for each window, we reverse the process described in Section 3.2. We then determine the bounding boxes of objects in the standard way, by thresholding the confidences and performing non-maxima suppression.

Our goal in these experiments is to show that, by modeling the interactions among several aspects of the scene and inferring their likelihoods together, we can do much better than if we estimate each one individually.

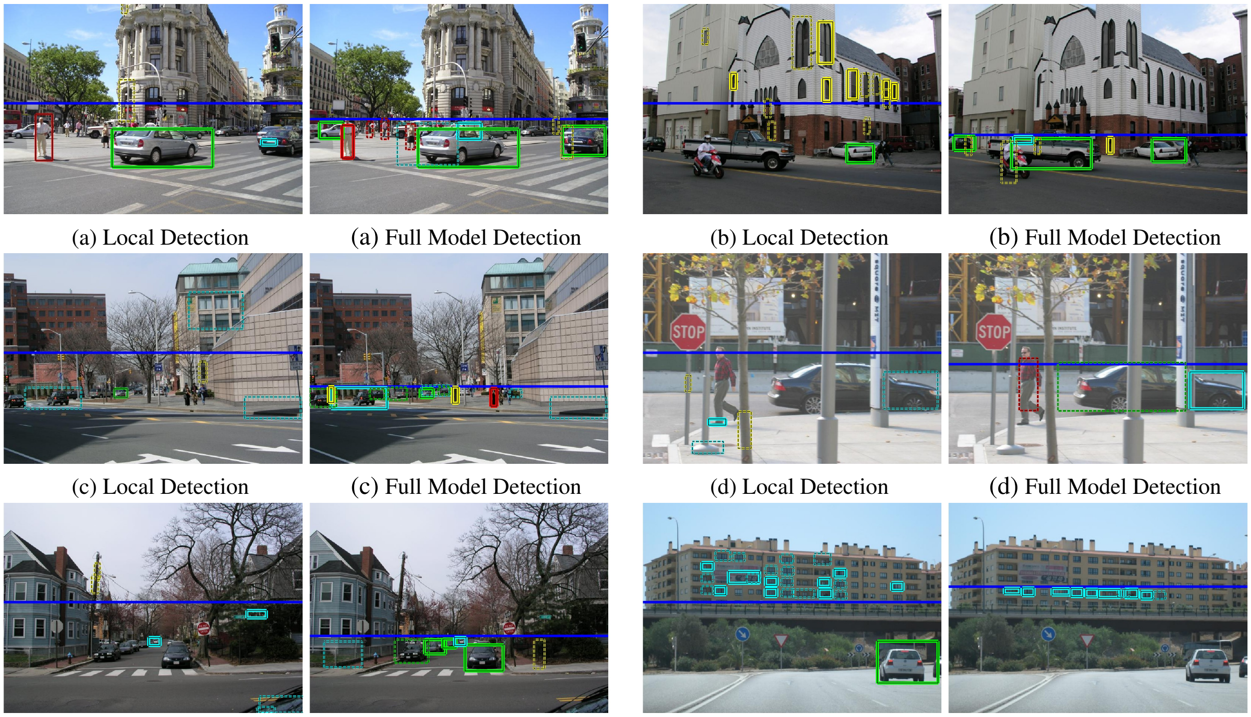
Object Detection Results. Figure 7 plots the ROC curves for car and pedestrian detection on our test set when different subsets of the model are considered. Figure 8 displays and discusses several examples. To provide an estimate of how much other detectors may improve under our framework, we report the percent

	Mean	Median
Prior	10.0%	8.5%
+Obj	7.5%	4.5%
+ObjGeom	7.0%	3.8%

Table 2. Object and geometry evidence improve horizon estimation. Mean/median absolute error (as percentage of image height) are shown for horizon estimates.

	Horizon	Cars (FP)		Ped (FP)	
Car	7.3%	5.6	7.4	—	—
Ped	5.0%	—	—	12.4	13.7
Car+Ped	3.8%	5.0	6.6	11.0	10.7

Table 3. Horizon estimation and object detection are more accurate when more object models are known. Results shown are using the full model in three cases: detecting only cars, only pedestrians, and both. The horizon column shows the median absolute error. For object detection we include the number of false positives per image at the 50% detection rate computed over all images (first number) and the subset of images that contain both cars and people (second number).



(a) Local Detection (a) Full Model Detection (b) Local Detection (b) Full Model Detection

(c) Local Detection (c) Full Model Detection (d) Local Detection (d) Full Model Detection

(e) Local Detection (e) Full Model Detection (f) Local Detection (f) Full Model Detection

Figure 8. We show car and pedestrian results from our baseline local detector (from [21]) and after inference using our model. The blue line shows the horizon estimate (always 0.5 initially). The boxes show detection estimates (green=true car, cyan=false car, red=true ped, yellow=false ped), with the solid lines being high confidence detections (0.5 FP/Image) and the dotted lines being lower confidence detections (2 FP/Image). In most cases, the horizon line is correctly recovered, and the object detection improves considerably. In particular, boxes that make no sense from a geometric standpoint (e.g. wrong scale (d), above horizon (b), in the middle of the ground (e)) usually are removed and objects not initially detected are found. Of course, improvement is not guaranteed. Pedestrians are often hallucinated (c,e) in places where they could be (but are not). In (f), a bad geometry estimate and repeated false detections along the building windows causes the horizon estimate to become worse and the car to be missed.

reduction in false negatives for varying false positive rates in Table 1. When the viewpoint and surface geometry are considered, about 20% of cars and pedestrians missed by the baseline are detected for the same false positive rate! The improvement due to considering the viewpoint is especially amazing, since the viewpoint uses no direct image evidence. Also note that, while individual use of surface geometry estimates and the viewpoint provides improvement, using both together improves results further.

Horizon Estimation Results. By performing inference over our model, the object and geometry evidence can also be used to improve the horizon estimates. We manually labeled the horizon in 100 of our images that contained both types of objects. Table 2 gives the mean and median absolute error over these images. Our prior of 0.50 results in a median error of 0.085% of the image height, but when objects and surface geometry are considered, the median error reduces to 0.038%. Notice how the geometry evidence provides a substantial improvement in horizon estimation, even though it is separated from the viewpoint by two variables in our model.

More is Better. Intuitively, the more types of objects that we can identify, the better our horizon estimates will be, leading to improved object detection. We verify this experimentally, performing the inference with only car detection, only pedestrian detection, and both. Table 3 gives the accuracy for horizon estimation and object detection when only cars are detected, when only pedestrians are detected, and when both are detected. As predicted, detecting two objects provides better horizon estimation and object detection than detecting one.

Dalal-Triggs Detector. To support our claim that any local object detector can be easily improved by plugging it into our framework, we performed experiments using the Dalal-Triggs detector [7] after converting the SVM outputs to probabilities using the method of [26]. We used code, data, and parameters provided by the authors, training an 80x24 car detector and 32x96 and 16x48 (for big and small) pedestrian detectors. The Dalal-Triggs local detector is currently among the most accurate for pedestrians, but its accuracy (Figure 7) improves considerably with our framework, from 57% to 66% detections at 1 false positive (FP) per image. Similarly, the car detection rate improves from 45% to 50% at 1 FP per image.

Integration is Important. In our model, surface and object information is integrated through a single camera viewpoint. What if we instead simply marginalized out the viewpoint prior over each object separately, essentially providing only a position/size prior for the objects? To find out, we re-ran our experiments using the Dalal-Triggs detectors after making this change. Our fully integrated model outperforms the weaker marginalized model by 5% (cars) and 8% (pedestrians) detection rate at 1 FP per image.

6. Viewpoint by Example

One of the benefits of our proposed system is the ability to recover camera viewpoint from a single image using the detections of known objects in the scene. But what if the image does not include any easily detectable known objects? This makes the problem extremely difficult. Solutions based on edges and perspective geometry, such as methods by Kosecka and Zhang [15] and Coughlan and Yuille [6], show good results for uncluttered man-made scenes with lots of parallel lines (the so-called Manhattan worlds), but fail for less structured environments. Inspired by work in pre-attentive human vision, Oliva and Torralba [24] convincingly argue that simple spatial-frequency statistics (the “gist”) of the image may be sufficient to recover a general sense of the space of the scene. They show some impressive results on estimating rough

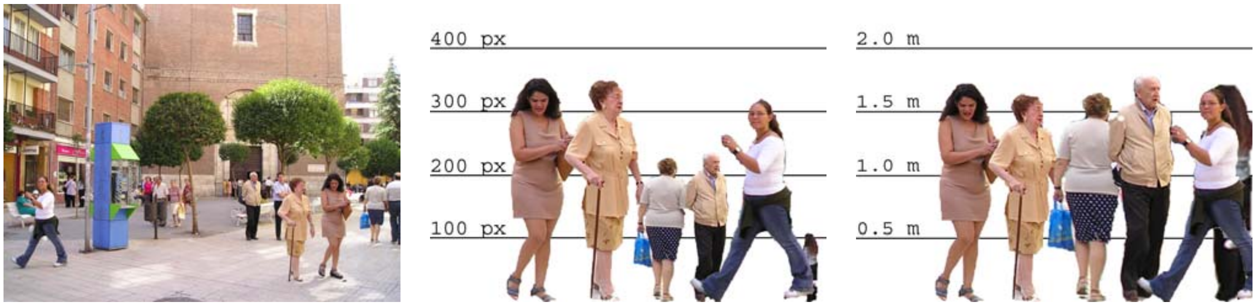


Figure 9. Automatic object height estimation. Objects taken from a typical image in LabelMe dataset (left) are first shown in their original pixel size (center), and after being resized according to their automatically estimated 3D heights (right).

"scene envelope" properties such as *openness*, *smoothness* and *depth* by matching to a small number of manually labeled training images. Torralba and Sinha [31] suggest that gist statistics could also provide a rough estimate of the horizon position. But to obtain more precise estimates, we require (1) a large amount of training data and (2) a way of accurately labeling this data.

Here we propose to solve both of these issues by applying our object-viewpoint model to automatically recover camera viewpoint parameters of images in a standard object recognition database. The resulting large dataset of image/viewpoint pairs then allows us to use the gist descriptor in a simple example-based approach to compute viewpoint estimates for a novel image. Moreover, we can use this informed viewpoint estimate in place of our simple Gaussian prior (Section 3.1) to improve the object detection results of our overall system.

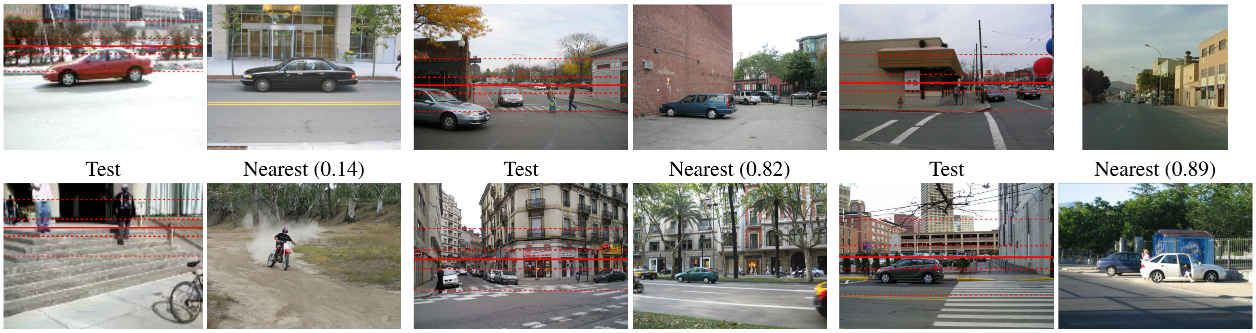
6.1. Discovery of Viewpoint and Object Size

Example-based techniques require many training samples to attain good performance. Manual labeling of camera viewpoint is tedious and error-prone, so we automatically recover the camera viewpoint and 3D object sizes in the LabelMe database [27]. Our method, described in [19], iterates between estimating the camera viewpoint for images that contain objects of known size distributions and estimating the size distributions of objects that are contained in images with known viewpoints. After initially providing only a guess of the mean and standard deviation height of people, we infer the camera viewpoint of over 5,000 images and heights of 13,000 object instances in roughly fifty object classes. Figure 9 shows an example of our automatic height estimation for people in an image.

Based on the inferred object heights, we re-estimate the 3D height distributions of cars (mean of 1.51m, standard deviation of 0.191m) and people (mean of 1.70m and standard deviation of 0.103m). We consider the camera viewpoint estimates to be reliable for the 2,660 images (excluding images in our test set) that contain at least two objects with known height distributions. Using the publicly available code, we compute the gist statistics (8x8 blocks at 3 scales with 8,4,4 orientations, giving 1280 variables per gist vector) over these images, providing a training set for viewpoint estimation.

6.2. Recovering the Viewpoint

In early experiments on our training set, we found that an image and its nearest neighbor (Euclidean distance in gist) tend to have similar horizon positions (correlation coefficient of 0.54). Using cross-validation, we evaluated nearest neighbor classifiers with various distance metrics, the nearest neighbor regression method of [22], and generalized regression neural networks (`newgrnn` in Matlab). The last of these (GRNN)



Test Nearest (0.14) Test Nearest (0.82) Test Nearest (0.89)

Test Nearest (0.95) Test Nearest (1.02) Test Nearest (1.33)

Figure 10. Horizon estimation. We show horizon estimation results (solid line) with 50% and 90% confidence bounds (dashed lines) for several test images with the gist nearest neighbor and distance $\tilde{d}$.

	Prior	P+ObjGeom	Gist	G+ObjGeom
Mean Error	10.0%	4.3%	5.7%	3.8%

Table 4. We show the mean error (as a percentage of image height) in horizon position estimation using a Gaussian prior, after considering surface geometry and objects (using the Dalal-Triggs detectors), our initial gist-based estimates, and after the full inference.

provides the lowest cross-validation error, after tuning the spread $\alpha = 0.22$ on our training set. The horizon estimate from the GRNN is given by

$$\tilde{v}_0(\mathbf{x}) = \frac{\sum_i v_{0i} w_i}{\sum_i w_i} \text{ with } w_i = \exp\left[\frac{-\|\mathbf{x} - \mathbf{x}_i\|^2}{2\alpha^2}\right] \tag{13}$$

where $\mathbf{x}$ is the gist statistics and v_{0i} is the horizon position for the i^{th} training sample.

Empirically, the true horizon position v_0 given the regression estimate $\tilde{v}_0$ has a Laplace probability density

$$\mathrm{p}(\mathrm{v}_0|\tilde{\mathrm{v}}_0) = \frac{1}{2\mathrm{s}_\mathrm{v}} \exp\left[\frac{-|\mathrm{v}_0 - \tilde{\mathrm{v}}_0|}{\mathrm{s}_\mathrm{v}}\right] \tag{14}$$

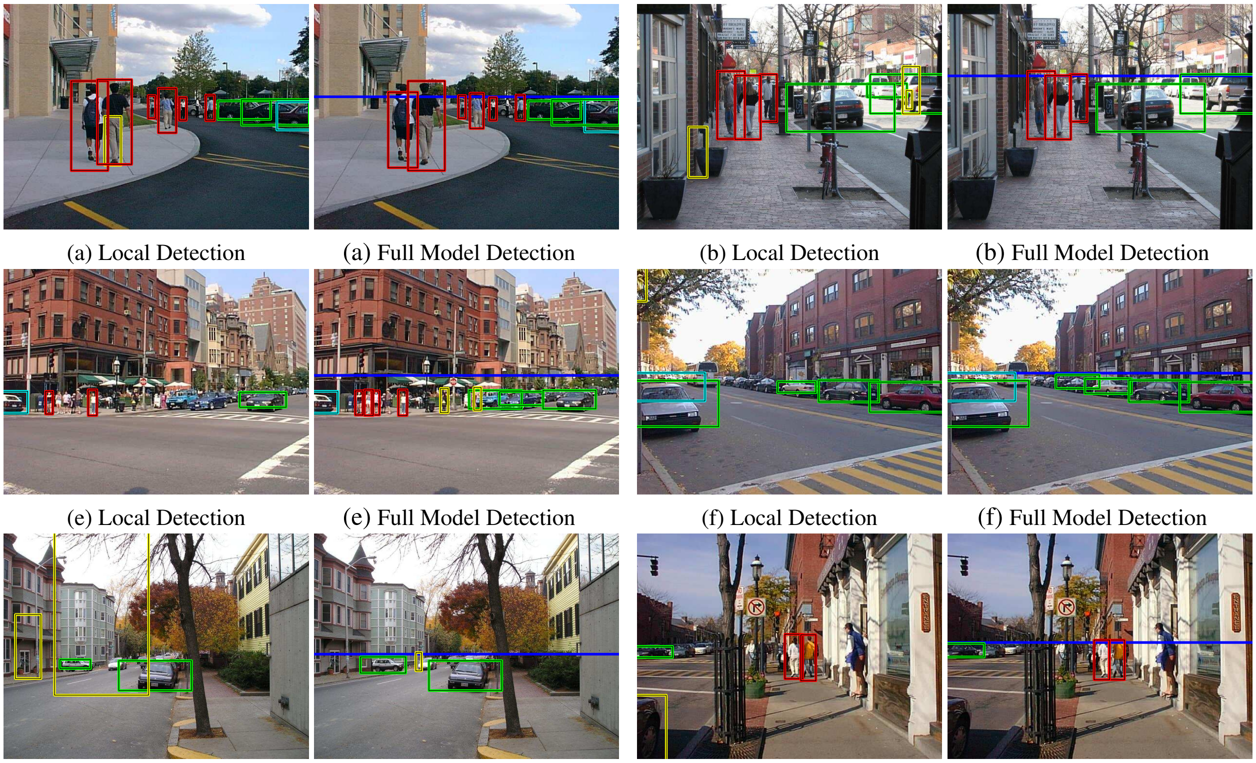
where s_v is the scale parameter and is equal to the expected error. We found a correlation (coefficient 0.27) between error and Euclidean distance to the nearest neighbor in gist statistics. Therefore, we can provide better estimates of confidence by considering the nearest neighbor distance. We fit $s_v = 0.022 + 0.060\tilde{d}$ by maximum likelihood estimation over our training set, where $\tilde{d}$ is the nearest neighbor distance.

We were not able to improve camera height estimates significantly over our prior estimate, probably because a small change in camera height has little impact on the global image statistics. We, therefore, simply re-estimate the camera height prior as in Section 4 using our larger training set.

6.3. Evaluation

In Table 4, we show that our example-based method for estimating the horizon position far outperforms the prior estimate (image center) and improves further when objects and surface geometry are considered. In Figure 10, we show several examples of test image, the nearest neighbor in our training set, and the estimated horizon. Our gist-based horizon estimates provide improvement in object detection as well, with detection rates increasing from 50% to 52% for cars and from 66% to 68% for pedestrians, at 1 false positive per image using the Dalal-Triggs object detectors. We show several examples of improved detection in Figure 11.

In summary, we can accurately estimate the horizon position from the gist statistics, providing: 1) a better final estimate of the horizon after considering objects and surface geometry; and 2) improved object detection.



(a) Local Detection (a) Full Model Detection (b) Local Detection (b) Full Model Detection

(e) Local Detection (e) Full Model Detection (f) Local Detection (f) Full Model Detection

(g) Local Detection (g) Full Model Detection (h) Local Detection (h) Full Model Detection

Figure 11. We show local object detections (left) of Dalal-Triggs (green=true car, cyan=false car, red=true ped, yellow=false ped) and the final detections (right) and horizon estimates (blue line) after considering surface geometry and camera viewpoint (initially estimated using our example-based method). Our method provides large improvement (+7%/11% for peds/cars at 1 FP/image) over a very good local detector. Many of the remaining recorded "false positives" re due to objects that are heavily occluded (a,e) or very difficult to see (e) (i.e., missed by the ground truth labeler). In (h), a missed person on the right exemplifies the need for more robust assumptions (e.g., a person *usually* rests on the ground plane) or explanation-based reasoning (e.g., the person only looks so tall because he is standing on a step).

These experiments nicely reinforce the key idea of this paper: with appropriate integration, improvement in one task benefits the others.

7. Discussion

In this paper, we have provided a "skeleton" model of a scene – a tree structure of camera viewpoint, objects, and surface geometry. We demonstrate our system's understanding of the 3D scene in Figure 12.

Our model makes several assumptions and approximations: all objects rest on the same ground plane; objects are perpendicular to the ground; camera tilt is small to moderate; camera roll is zero or image is rectified; camera intrinsic parameters are typical (zero skew, unit aspect ratio, typical focal length); and object and surface evidence are conditionally independent given the labels. Of these, the first is the most limiting and could be relaxed simply by using a mixture model in which objects are likely to rest on the same plane but *could* be anywhere with non-zero probability. A repeated feature in the image, such as the building windows in Figure 8(f), can cause object detection responses to be correlated and an incorrect scene interpretation to result. Modeling these correlations (for example, if two object patches are very similar, consider their evidence jointly, not as conditionally independent) could improve results. The other

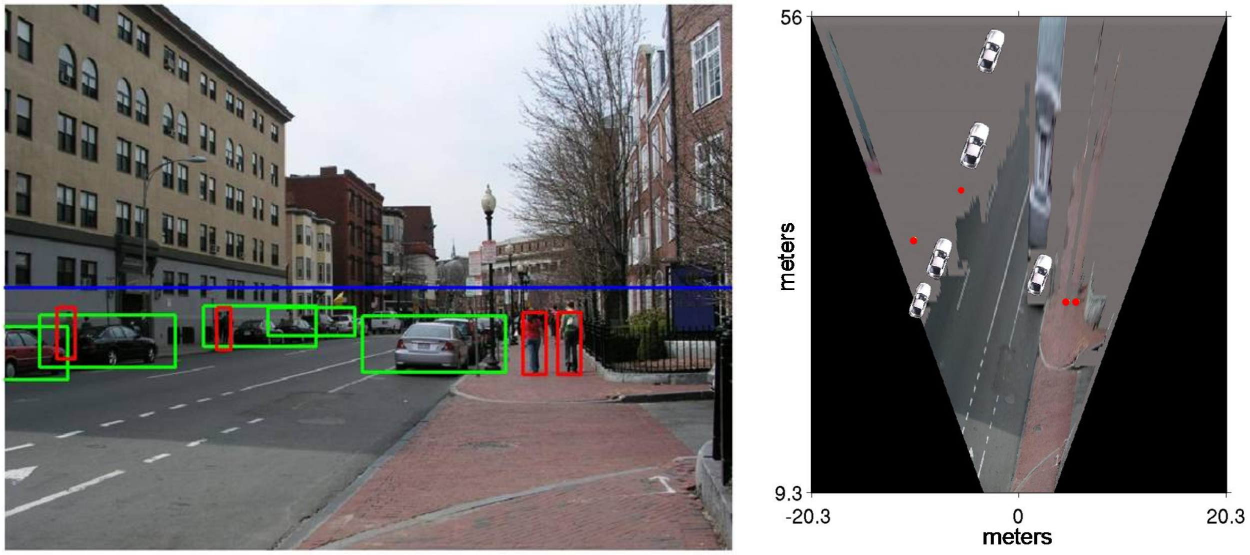


Figure 12. We project the estimated ground surface into an overhead view, using the estimated camera viewpoint, and plot scaled icons of objects (red dots for pedestrians) at their detected (using Dalal-Triggs) ground positions. Car orientation is estimated by a robust line fit, assuming that cars mostly face down the same line. To plot in metric scale, we assume a typical focal length.

approximations cause graceful degradation when violated. Our model makes no assumptions about the scene (e.g., forest vs. urban vs. indoor), but the surface classifier used in our experiments was trained on outdoor images only.

Our model-based approach has two main advantages over the more direct "bag of features/black box" classification method: 1) subtle relationships (such as that object sizes relate through the viewpoint) can be easily represented; and 2) additions and extensions to the model are easy (the direct method requires complete retraining whenever anything changes). To add a new object to our model, one needs only to train a detector for that object and supply the distribution of the object's height in the 3D scene. Our framework could also be extended by modeling other scene properties, such as scene category. By modeling the direct relationships of objects and geometry (which can be done in 3D, since perspective is already part of our framework) further improvement is possible.

As more types of objects can be identified and more aspects of the scene can be estimated, we hope that our framework will eventually grow into a vision system that would fulfill the ambitions of the early computer vision researchers – a system capable of complete image understanding.

Acknowledgements. We thank Bill Freeman for useful suggestions about the inference, Navneet Dalal for providing code and data, Moshe Mahler for his illustration in Figure 2, Takeo Kanade for his car-road illustrative example, Kevin Murphy for creating the Bayes Net Toolbox, and Antonio Torralba for making the gist code available. This research was funded in part by NSF CAREER award IIS-0546547 and a Microsoft Research Fellowship. We also thank the reviewers for their kindly admonitions to expand derivation and discussion.

References

[1] The PASCAL object recognition database collection. Website, 2005. http://www.pascal-network.org/challenges/VOC/.

[2] H. Barrow and J. Tenenbaum. Recovering intrinsic scene characteristics from images. In *Comp. Vision Systems*, 1978.

[3] I. Biederman. On the semantics of a glance at a scene. In M. Kubovy and J. R. Pomerantz, editors, *Perceptual Organization*, chapter 8. Lawrence Erlbaum, 1981.

[4] R. Brooks, R. Greiner, and T. Binford. Model-based three-dimensional interpretation of two-dimensional images. In *IJCAI*, 1979.

[5] M. Collins, R. Schapire, and Y. Singer. Logistic regression, Adaboost and bregman distances. *Machine Learning*, 48(1-3), 2002.

[6] J. Coughlan and A. Yuille. Manhattan world: orientation and outlier detection by bayesian inference. *Neural Computation*, 15(5), 2003.

[7] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[8] D. A. Forsyth, J. L. Mundy, A. Zisserman, and C. A. Rothwell. Using global consistency to recognise Euclidean objects with an uncalibrated camera. In *CVPR*, 1994.

[9] M. Greienhagen, V. Ramesh, D. Comaniciu, and H. Niemann. Statistical modeling and performance characterization of a real-time dual camera surveillance system. In *CVPR*, 2000.

[10] A. Hanson and E. Riseman. VISIONS: A computer system for interpreting scenes. In *Computer Vision Systems*, 1978.

[11] X. He, R. S. Zemel, and M. Á. Carreira-Perpiñán. Multiscale conditional random fields for image labeling. In *CVPR*, 2004.

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

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

[14] S. G. Jeong, C. S. Kim, D. Y. Lee, S. K. Ha, D. H. Lee, M. H. Lee, and H. Hashimoto. Real-time lane detection for autonomous vehicle. In *ISIE*, 2001.

[15] J. Kosecka and W. Zhang. Video compass. In *ECCV*. Springer-Verlag, 2002.

[16] N. Krahnstoever and P. R. S. Mendona. Bayesian autocalibration for surveillance. In *ICCV*, 2005.

[17] S. Kumar and M. Hebert. Discriminative random fields: A discriminative framework for contextual interaction in classification. In *ICCV*, 2003.

[18] S. Kumar and M. Hebert. A hierarchical field framework for unified context-based classification. In *ICCV*, 2005.

[19] J.-F. Lalonde, D. Hoiem, A. A. Efros, C. Rother, J. Winn, and A. Criminisi. Photo clip art. In *ACM SIGGRAPH 2007*.

[20] K. Murphy. The bayes net toolbox for matlab. In *Computing Science and Statistics*, volume 33. 2001.

[21] K. Murphy, A. Torralba, and W. T. Freeman. Graphical model for recognizing scenes and objects. In *NIPS*. 2003.

[22] A. Navot, L. Shpigelman, N. Tishby, and E. Vaadia. Nearest neighbor based feature selection for regression and its application to neural activity. In *NIPS*. 2006.

[23] Y. Ohta. *Knowledge-Based Interpretation Of Outdoor Natural Color Scenes*. Pitman, 1985.

[24] A. Oliva and A. Torralba. Building the gist of a scene: The role of global image features in recognition.

[25] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., 1988.

[26] J. C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*, 2000.

[27] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. LabelMe: a database and web-based tool for image annotation. Technical report, MIT, 2005.

[28] H. Schneiderman. Learning a restricted bayesian network for object detection. In *CVPR*, 2004.

[29] E. Sudderth, A. Torralba, W. T. Freeman, and A. Wilsky. Learning hierarchical models of scenes, objects, and parts. In *ICCV*, 2005.

[30] A. Torralba. *Contextual Influences on Saliency*, pages 586–593. Academic Press / Elsevier, 2005.

[31] A. Torralba and P. Sinha. Statistical context priming for object detection. In *ICCV*, 2001.

[32] Z. Tu, X. Chen, A. L. Yuille, and S. C. Zhu. Image parsing: Unifying segmentation, detection, and recognition. *IJCV*, 63(2):113–140, 2005.

[33] P. Viola and M. J. Jones. Robust real-time face detection. *IJCV*, 57(2), 2004.