Geometric Context from a Single Image

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

Abstract

Many computer vision algorithms limit their performance by ignoring the underlying 3D geometric structure in the image. We show that we can estimate the coarse geometric properties of a scene by learning appearance-based models of geometric *classes, even in cluttered natural scenes. Geometric classes describe the 3D orientation of an image region with respect to the camera. We provide a multiple-hypothesis framework for robustly estimating scene structure from a single image and obtaining confidences for each geometric label. These confidences can then be used to improve the performance of many other applications. We provide a thorough quantitative evaluation of our algorithm on a set of outdoor images and demonstrate its usefulness in two applications: object detection and automatic single-view reconstruction.*

1. Introduction

How can object recognition, while seemingly effortless for humans, remain so excruciatingly difficult for computers? The reason appears to be that recognition is inherently a *global process*. From sparse, noisy, local measurements our brain manages to create a coherent visual experience. When we see a person at the street corner, the simple act of recognition is made possible not just by the pixels inside the person-shape (there are rarely enough of them!), but also by many other cues: the surface on which he is standing, the 3D perspective of the street, the orientation of the viewer, etc. In effect, our entire visual panorama acts as a global recognition *gestalt*.

In contrast, most existing computer vision systems attempt to recognize objects using local information alone. For example, currently popular object detection algorithms [26, 32, 33] assume that all relevant information about an object is contained within a small window in the image plane (objects are found by exhaustively scanning over all locations and scales). Note that typical errors made by such systems – finding faces in tree-tops or cars in keyboards – are not always the result of poor object modeling! There really *are* faces in the tree-tops when one only looks at the world through a small peephole [31]. But if our eventual goal is to approach the level of human performance, then we must look outside the box and consider the entire image as context for the global recognition task.

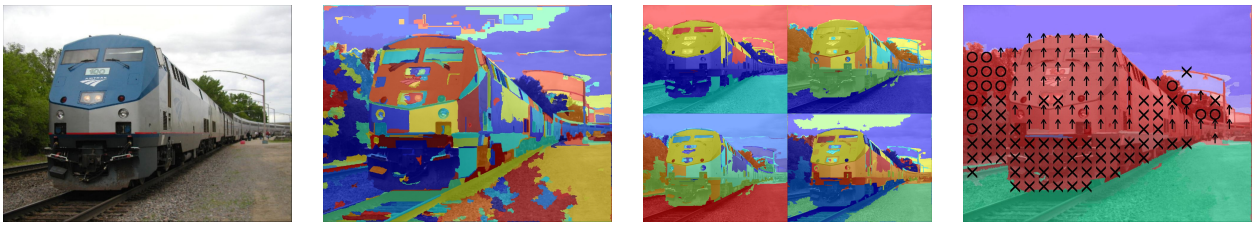
The recent work of Torralba *et al.* [29, 30] has been very influential in showing the importance of global scene context for object detection. Low-level features have also been used to get a coarse representation of the scene in [1, 25]. Other researchers have exploited local contextual information using random field frameworks [16, 2, 11] and other representations (e.g. [19]). Unfortunately, the above methods all encode contextual relationships between objects *in the image plane* and not in the 3D world where these objects actually reside. This proves a severe limitation, preventing important information – scale relationships, surface orientations, free-space reasoning, etc. – from ever being captured. Clearly, 2D context is not enough.

Figure 1: Geometric context from a single image: ground (green), sky (blue), vertical regions (red) subdivided into planar orientations (arrows) and non-planar solid ('x') and porous ('o').

Our ultimate goal is to recover a 3D "contextual frame" of an image, a sort of theater stage representation containing major surfaces and their relationships to each other. Having such a representation would then allow each object to be physically "placed" within the frame and permit reasoning between the different objects and their 3D environment.

In this paper, we take the first steps toward constructing this contextual frame by proposing a technique to estimate the coarse orientations of large surfaces in outdoor images. We focus on outdoor images because their lack of human-imposed manhattan structure creates an interesting and challenging problem. Each image pixel is classified as either being part of the ground plane, belonging to a surface that sticks up from the ground, or being part of the sky. Surfaces sticking up from the ground are then subdivided into planar surfaces facing left, right or toward the camera and non-planar surfaces, either porous (e.g. leafy vegetation or a mesh of wires) or solid (e.g. a person or tree trunk). We also present initial results in object detection and 3D reconstruction that demonstrate the usefulness of this geometric information.

We pose the problem of 3D geometry estimation in terms of statistical learning. Rather than trying to explicitly compute all of the required geometric parameters from the im-



(a) Input (b) Superpixels (c) Multiple Hypotheses (d) Geometric Labels

Figure 2: To obtain useful statistics for modeling geometric classes, we slowly build our structural knowledge of the image: from pixels (a), to superpixels (b), to multiple potential groupings of superpixels (c), to the final geometric labels (d).

age, we rely on other images (a training set) to furnish this information in an implicit way, through recognition. But in contrast to most recognition approaches that model semantic classes, such as cars, vegetation, roads, or buildings [21, 6, 14, 28], our goal is to model *geometric* classes that depend on the orientation of a physical object with relation to the scene. For instance, a piece of plywood lying on the ground and the same piece of plywood propped up by a board have two different geometric classes. Unlike other reconstruction techniques that require multiple images (e.g. [23]), manual labeling [4, 17], or very specific scenes [9], we want to automatically estimate the 3D geometric properties of general outdoor scenes from a single image.

The geometric context is philosophically similar to the $2\frac{1}{2}$D sketch proposed by David Marr [18]. However, we differ from it in several important ways: 1) we use statistical learning instead of relying solely on a geometric or photometric methodology (e.g. Shape-from-X methods), 2) we are interested in a rough sense of the scene geometry, not the orientation of every single surface, and 3) our geometric context is to be used *with* the original image data, not as a substitute for it.

We observed two tendencies in a sampling of 300 outdoor images that we collected using Google's image search The first is that over 97% of image pixels belong to one of three main geometric classes: the ground plane, surfaces at roughly right angles to the ground plane, and the sky. Thus, our small set of geometric classes is sufficient to provide an accurate description of the surfaces in most images. Our second observation is that, in most images, the camera axis is roughly parallel (within 15 degrees) to the ground plane. We make this rough alignment an assumption, reconciling world-centric cues (e.g. material) and view-centric cues (e.g. perspective).

Our main insight is that 3D geometric information can be obtained from a single image by learning appearance-based models of surfaces at various orientations. We present a framework that progressively builds structural knowledge of the scene by alternately using estimated scene structure to compute more complex image features and using these more complex image features to gain more structural knowledge. Additionally, we provide a thorough analysis of the impact of different design choices in our algorithm and offer evidence of the usefulness of our geometric context.

2. Obtaining Useful Geometric Cues

A patch in the image could theoretically be generated by a surface of any orientation in the world. To determine which orientation is most *likely*, we need to use all of the available cues: material, location, texture gradients, shading, vanishing points, etc. Much of this information, however, can be extracted only when something is known about the structure of the scene. For instance, knowledge about the intersection of nearly parallel lines in the image is often extremely useful for determining the 3D orientation, but only when we know that the lines belong to the same planar surface (e.g. the face of a building or the ground). Our solution is to slowly build our structural knowledge of the image: from pixels to superpixels to related groups of superpixels (see Figure 2).

Our first step is to apply the over-segmentation method of Felzenszwalb *et al.* [7] to obtain a set of "superpixels". Each superpixel is assumed to correspond to a single label (superpixels have been shown to respect segment boundaries [24]). Unlike plain pixels, superpixels provide the spatial support that allows us to compute some basic first-order statistics (e.g. color and texture). To have any hope of estimating the orientation of large-scale surfaces, however, we need to compute more complex geometric features that must be evaluated over fairly large regions in the image. How can we find such regions? One possibility is to use a standard segmentation algorithm (e.g. [27]) to partition the image into a small number of homogeneous regions. However, since the cues used in image segmentation are themselves very basic and local, there is little chance of reliably obtaining regions that correspond to entire surfaces in the scene.

2.1. Multiple Hypothesis Method

Ideally, we would evaluate all possible segmentations of an image to ensure that we find the best one. To make this tractable, we sample a small number of segmentations that are representative of the entire distribution. Since sampling from all of the possible pixel segmentations is infeasible, we reduce the combinatorial complexity of the search further by sampling sets of superpixels.

Our approach is to make multiple segmentation hypotheses based on simple cues and then use each hypothesis' increased spatial support to better evaluate its quality. Different hypotheses vary in the number of segments and make errors in different regions of the image (see Figure 2c). Our

Feature Descriptions	Num
Color	**16**
C1. RGB values: mean	3
C2. HSV values: C1 in HSV space	3
C3. Hue: histogram (5 bins) and entropy	6
C4. Saturation: histogram (3 bins) and entropy	4
Texture	**15**
T1. DOOG filters: mean abs response of 12 filters	12
T2. DOOG stats: mean of variables in T1	1
T3. DOOG stats: argmax of variables in T1	1
T4. DOOG stats: (max - median) of variables in T1	1
Location and Shape	**12**
L1. Location: normalized x and y, mean	2
L2. Location: norm. x and y, 10^{th} and 90^{th} pctl	4
L3. Location: norm. y wrt horizon, 10^{th}, 90^{th} pctl	2
L4. Shape: number of superpixels in region	1
L5. Shape: number of sides of convex hull	1
L6. Shape: $num\ pixels/area(convex\ hull)$	1
L7. Shape: whether the region is contiguous $\in \{0, 1\}$	1
3D Geometry	**35**
G1. Long Lines: total number in region	1
G2. Long Lines: % of nearly parallel pairs of lines	1
G3. Line Intsctn: hist. over 12 orientations, entropy	13
G4. Line Intsctn: % right of center	1
G5. Line Intsctn: % above center	1
G6. Line Intsctn: % far from center at 8 orientations	8
G7. Line Intsctn: % very far from center at 8 orient.	8
G8. Texture gradient: x and y "edginess" (T2) center	2

Table 1: Features computed on superpixels (C1-C2,T1-T4,L1) and on regions (all). The "Num" column gives the number of features in each set. The boosted decision tree classifier selects a discriminative subset of these features.

challenge, then, is to determine which parts of the hypotheses are likely to be correct and to accurately determine the labels for those regions.

2.2. Features

Table 1 lists the features used by our system. Color and texture allow the system to implicitly model the relation between material and 3D orientation. Image location also provides strong 3D geometry cues (e.g. ground is below sky). Our previous work [12] provides further rationale for these features.

Although the 3D orientation of a plane (relative to the viewer) can be completely determined by its vanishing line [10], such information cannot easily be extracted from relatively unstructured outdoor images. By computing statistics of straight lines (G1-G2) and their intersections (G3-G7) in the image, our system gains information about the vanishing points of a surface without explicitly computing them. Our system finds long, straight edges in the image using the method of [15]. The intersections of nearly parallel lines (within $\pi/8$ radians) are radially binned from the image center, according to direction (8 orientations) and distance (2 thresholds, at 1.5 and 5 times the image size). When computing G1-G7, we weight the lines by length, improving robustness to outliers. The texture gradient (G8) can also provide orientation cues, even for natural surfaces without parallel lines.

3. Learning Segmentations and Labels

We gathered a set of 300 outdoor images representative of the images that users choose to make publicly available on the Internet. These images are often highly cluttered and span a wide variety of natural, suburban, and urban scenes. Figure 4 shows twenty of these images. Each image is oversegmented, and each segment is given a ground truth label according to its geometric class. In all, about 150,000 superpixels are labeled. We use 50 of these images to train our segmentation algorithm. The remaining 250 images are used to train and evaluate the overall system using 5-fold cross-validation. We make our database publicly available for comparison[1].

3.1. Generating Segmentations

We want to obtain multiple segmentations of an image into geometrically homogeneous regions[2]. We take a learning approach to segmentation, estimating the likelihood that two superpixels belong in the same region. We generate multiple segmentations by varying the number of regions and the initialization of the algorithm.

Ideally, for a given number of regions, we would maximize the joint likelihood that all regions are homogeneous. Unfortunately, finding the optimal solution is intractable; instead, we propose a simple greedy algorithm based on pairwise affinities between superpixels. Our algorithm has four steps: 1) randomly order the superpixels; 2) assign the first n_r superpixels to different regions; 3) iteratively assign each remaining superpixel based on a learned pairwise affinity function (see below); 4) repeat step 3 several times. We want our regions to be as large as possible (to allow good feature estimation) while still being homogeneously labeled. We run this algorithm with different numbers of regions ($n_r \in \{3, 4, 5, 7, 9, 11, 15, 20, 25\}$ in our implementation).

Training. We sample pairs of same-label and different-label superpixels (2,500 each) from our training set. We then estimate the likelihood that two superpixels have the same label based on the absolute differences of their feature values: $P(y_i = y_j || \mathbf{x}_i - \mathbf{x}_j |)$. We use the logistic regression form of Adaboost [3] with weak learners based on naive density estimates:

$$f_m(\mathbf{x}_1, \mathbf{x}_2) = \sum_i^{n_f} log \frac{P(y_1 = y_2, |x_{1i} - x_{2i}|)}{P(y_1 \neq y_2, |x_{1i} - x_{2i}|)} \qquad (1)$$

where n_f is the number of features. Each likelihood function in the weak learner is obtained using kernel density estimation [5] over the m^{th} weighted distribution.

We assign a superpixel to the region (see step 3 above) with the maximum average pairwise log likelihood between the superpixels in the region and the superpixel being added.

[1] Project page: http://www.cs.cmu.edu/∼dhoiem/projects/context/

[2] A region is "homogeneous" if each of its superpixel has the same label. The regions need not be contiguous.

In an experiment comparing our segmentations with ground truth, using our simple grouping method, 40% of the regions were homogeneously labeled[3], 89% of the superpixels were in at least one homogeneous region for the main classes, and 61% of the vertical superpixels were in at least one homogeneous region for the subclasses. A superpixel that is never in a homogeneous region can still be correctly labeled, if the label that best describes the region is the superpixel's label.

3.2. Geometric Labeling

We compute the features for each region (Table 1) and estimate the probability that all superpixels have the same label (homogeneity likelihood) and, given that, the confidence in each geometric label (label likelihood). After forming multiple segmentation hypotheses, each superpixel will be a member of several regions, one for each hypothesis. We determine the superpixel label confidences by averaging the label likelihoods of the regions that contain it, weighted by the homogeneity likelihoods:

$$C(y_i = v|\mathbf{x}) = \sum_{j}^{n_h} P(y_j = v|\mathbf{x}, \mathbf{h}_{ji}) P(\mathbf{h}_{ji}|\mathbf{x}) \qquad (2)$$

where C is the label confidence, y_i is the superpixel label, v is a possible label value, $\mathbf{x}$ is the image data, n_h is the number of hypotheses, $\mathbf{h}_{ji}$ defines the region that contains the i^{th} superpixel for the j^{th} hypothesis, and y_j is the region label.[4] The sum of the label likelihoods for a particular region and the sum of the homogeneity likelihoods for all regions containing a particular superpixel are normalized to sum to one. The main geometric labels and vertical subclass labels are estimated independently (subclass labels are assigned to the entire image but are applied only to vertical regions).

Training. We first create several segmentation hypotheses for each training image using the learned pairwise likelihoods. We then label each region with one of the main geometric classes or "mixed" when the region contains multiple classes and label vertical regions as one of the subclasses or "mixed". Each label likelihood function is then learned in a one-vs.-rest fashion, and the homogeneity likelihood function is learned by classifying "mixed" vs. homogeneously labeled. Both the label and the homogeneity likelihood functions are estimated using the logistic regression version of Adaboost [3] with weak learners based on eight-node decision trees [8]. Decision trees make good weak learners, since they provide automatic feature selection and limited modeling of the joint statistics of features. Since correct classification of large regions is more important than of small regions, the weighted distribution is initialized to be proportional to the percentage of image area spanned.

[3]To account for small manual labeling errors, we allow up to 5% of the a region's pixels to be different than the most common label.

[4]If one were to assume that there is a single "best" hypothesis, Equation 2 has the interpretation of marginalizing over a set of possible hypotheses.

Geometric Class			
	Ground	Vertical	Sky
Ground	0.78	0.22	0.00
Vertical	0.09	0.89	0.02
Sky	0.00	0.10	0.90

Table 2: Confusion matrix for the main geometric classes.

Vertical Subclass					
	Left	Center	Right	Porous	Solid
Left	0.15	0.46	0.04	0.15	0.21
Center	0.02	0.55	0.06	0.19	0.18
Right	0.03	0.38	0.21	0.17	0.21
Porous	0.01	0.14	0.02	0.76	0.08
Solid	0.02	0.20	0.03	0.26	0.50

Table 3: Confusion matrix for the vertical structure subclasses.

4. Results

We test our system on 250 images using 5-fold cross-validation. We note that the cross-validation was *not* used to select any classification parameters. Accuracy is measured by the percentage of image pixels that have the correct label, averaged over the test images. See our web site for the 250 input images, the ground truth labels, and our results.

4.1. Geometric Classification

Figure 4 shows the labeling results of our system on a sample of the images. Tables 2 and 3 give the confusion matrices of the main geometric classes (ground plane, vertical things, sky) and the vertical subclasses (left-facing plane, front-facing plane, right-facing plane, porous non-planar, solid non-planar). The overall accuracy of the classification is 86% and 52% for the main geometric classes and vertical subclasses, respectively (see Table 4 for baseline comparisons with simpler methods). The processing time for a 640x480 image is about 30 seconds using a 2.13GHz Athalon processor and unoptimized MATLAB code.

As the results demonstrate, vertical structure subclasses are much more difficult to determine than the main geometric classes. This is mostly due to ambiguity in assigning ground truth labels, the larger number of classes, and a reduction of useful cues (e.g. material and location are not very helpful for determining the subclass). Our labeling results (Figures 4 and 5), however, show that many of the system's misclassifications are still reasonable.

4.2. Importance of Structure Estimation

Earlier, we presented a multiple hypothesis method for robustly estimating the structure of the underlying scene before determining the geometric class labels. To verify that this intermediate structure estimation is worthwhile, we tested the accuracy of the system when classifying based on only class priors (CPrior), only pixel locations (Loc), only color and texture at the pixel level (Pixel), all features at the

Intermediate Structure Estimation						
	CPrior	Loc	Pixel	SPixel	OneH	MultiH
Main	49%	66%	80%	83%	83%	86%
Sub	34%	36%	43%	45%	44%	52%

Table 4: Classification accuracy for different levels of intermediate structure estimation. "Main" is the classification among the three main classes. "Sub" is the subclassification of the vertical structures class. The column labels are defined in Section 4.2.

Importance of Different Feature Types				
	Color	Texture	Loc/Shape	Geometry
Main	6%	2%	16%	2%
Sub	6%	2%	8%	7%

Table 5: The drop in overall accuracy caused by individually *removing* each type of feature.

superpixel level (SPixel), a single ($n_r = 9$) segmentation hypothesis (OneH), and using our full multiple-hypothesis framework (MultiH). Our results (Table 4) show that each increase in the complexity of the algorithm offers a significant gain in classification accuracy.

We also tested the accuracy of the classifier when the intermediate scene structure is determined by partitioning the superpixels according the ground truth labels. This experiment gives us an intuition of how well our system would perform if our grouping and hypothesis evaluation algorithms were perfect. Under this ideal partitioning, the classifier accuracy is 95% for the main geometric classes and 66% for the vertical subclasses[5]. Thus, large gains are possible by improving our simple grouping algorithm, but much work remains in defining better features and a better classifier.

4.3. Importance of Cues

Our system uses a wide variety of statistics involving location and shape, color, texture, and 3D geometric information. We analyzed the usefulness of each type of information by removing all features of a given type from the feature set and re-training and testing the system. Table 5 displays the results, demonstrating that information about each type of feature is important but non-critical. These results show that location has a strong role in the system's performance, but our experiments in structure estimation show that location needs to be supplemented with other cues. Color, texture, and location features affect both the segmentation and the labeling. Geometric features affect only labeling. Figure 6 qualitatively demonstrates the importance of using all available cues.

5. Applications

We have shown that we are able to extract geometric information from images. We now demonstrate the usefulness of this information in two areas: object detection and automatic single-view reconstruction.

[5]Qualitatively, the subclass labels contain very few errors. Ambiguities such as when "left" becomes "center" and when "planar" becomes "non-planar" inflate the error estimate.

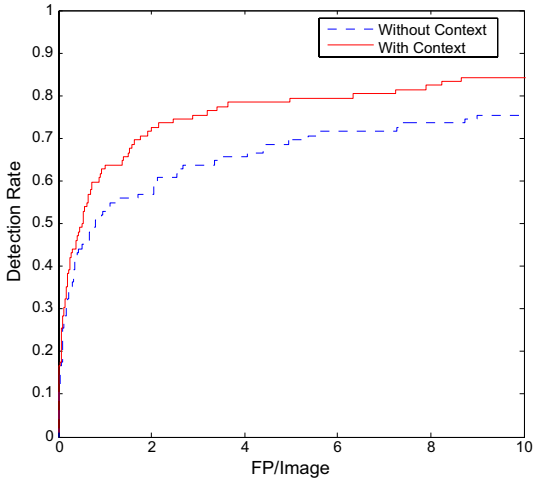


Figure 3: ROC for Car Detection. Detectors were trained and tested using identical data, except that the detector "With Context" used an additional 40 context features computed from the confidence values outputted by our system.

5.1. Object Detection

Our goal in this experiment is to demonstrate that our contextual information improves performance in an *existing* object detection system, even when naively applied. We train and test a multiple-orientation car detector using the PASCAL [22] training and validation sets with the grayscale images removed. We use a local detector from Murphy *et al.* [20] that employs GentleBoost to form a classifier based on fragment templates. We train two versions of the system, one using 500 local features (templates) and one that adds 40 new contextual features from the geometric context. The contextual features are the average confidence for the object window region (center), average confidences for the windows above and below the object, and the above-center and below-center differences for each of the three main geometric classes and five subclasses. Our results (Figure 3) show that the geometric contextual information improves detection performance considerably. When training, four out of the first five features selected by the boosting algorithm were contextual. The most powerful (first selected) feature indicates that cars are usually less ground-like than the region immediately below them. Figure 7 shows two specific examples of improvement.

Our representation of geometric context in this experiment is quite simple. In future work, we plan to use our geometric information to construct a 3D contextual frame, allowing powerful reasoning about objects in the image. We believe that providing such capabilities to computer vision algorithms could result in substantially better systems.

5.2. Automatic Single-View Reconstruction

Our main geometric class labels and a horizon estimate are sufficient to reconstruct coarse scaled 3D models of many outdoor scenes. By fitting the ground-vertical intersection in the image, we are able to "pop up" the vertical surfaces from the ground. Figure 8 shows the Merton College im-

age from [17] and two novel views from a texture-mapped 3D model automatically generated by our system. The details on how to construct these models and additional results are presented in our companion graphics paper [12]. Object segmentation and estimation of the intrinsic and extrinsic camera parameters would make automatic construction of metric 3D models possible for many scenes. Besides the obvious graphics applications, we believe that such models would provide extremely valuable information to other computer vision applications.

6. Conclusion

We have taken important steps toward being able to analyze objects in the image within the context of the 3D world. Our results show that such context can be estimated and usefully applied, even in outdoor images that lack human-imposed structure. Our contextual models could be improved by including additional geometric cues (e.g. symmetry [13]), estimating camera parameters, or improving the classification techniques. Additionally, much research remains in finding the best ways to apply this context to improve other computer vision applications.

Acknowledgments. We thank Jianbo Shi and Rahul Sukthankar for helpful discussions and suggestions. We also thank David Liebowitz for his Merton College image.

References

[1] B. Bose and W. E. L. Grimson, "Improving object classification in far-field video," in *Proc. CVPR*, 2004.

[2] P. Carbonetto, N. de Freitas, and K. Barnard, "A statistical model for general contextual object recognition," in *Proc. ECCV*, 2004.

[3] M. Collins, R. Schapire, and Y. Singer, "Logistic regression, adaboost and bregman distances," *Machine Learning*, vol. 48, no. 1-3, 2002.

[4] A. Criminisi, I. Reid, and A. Zisserman, "Single view metrology," *IJCV*, vol. 40, no. 2, 2000.

[5] R. Duda, P. Hart, and D. Stork, *Pattern Classification*. Wiley-Interscience Publication, 2000.

[6] M. R. Everingham, B. T. Thomas, and T. Troscianko, "Head-mounted mobility aid for low vision using scene classification techniques," *Int. J. of Virt. Reality*, vol. 3, no. 4, 1999.

[7] P. Felzenszwalb and D. Huttenlocher, "Efficient graph-based image segmentation," *IJCV*, vol. 59, no. 2, 2004.

[8] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: a statistical view of boosting," *Annals of Statistics*, vol. 28, no. 2, 2000.

[9] F. Han and S.-C. Zhu, "Bayesian reconstruction of 3d shapes and scenes from a single image," in *Int. Work. on Higher-Level Know. in 3D Modeling and Motion Anal.*, 2003.

[10] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2004.

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

[12] D. Hoiem, A. A. Efros, and M. Hebert, "Automatic photo pop-up," in *ACM SIGGRAPH 2005*.

[13] W. Hong, A. Y. Yang, K. Huang, and Y. Ma, "On symmetry and multiple-view geometry: Structure, pose, and calibration from a single image," *IJCV*, vol. 60, no. 3, 2004.

[14] S. Konishi and A. Yuille, "Statistical cues for domain specific image segmentation with performance analysis." in *Proc. CVPR*, 2000.

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

[16] S. Kumar and M. Hebert, "Discriminative random fields: A discriminative framework for contextual interaction in classification," in *Proc. ICCV*. IEEE Comp. Society, 2003.

[17] D. Liebowitz, A. Criminisi, and A. Zisserman, "Creating architectural models from images," in *Proc. EuroGraphics*, vol. 18, 1999.

[18] D. Marr, *Vision*. San Francisco: Freeman, 1982.

[19] K. Mikolajczyk, C. Schmid, and A. Zisserman, "Human detection based on a probabilistic assembly of robust part detectors," in *Proc. ECCV*. Springer-Verlag, May 2004.

[20] K. Murphy, A. Torralba, and W. T. Freeman, "Graphical model for recognizing scenes and objects," in *Proc. NIPS*, 2003.

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

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

[23] M. Pollefeys, R. Koch, and L. J. V. Gool, "Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters," in *Proc. ICCV*, 1998.

[24] X. Ren and J. Malik, "Learning a classification model for segmentation," in *Proc. ICCV*, 2003.

[25] U. Rutishauser, D. Walther, C. Koch, and P. Perona, "Is bottom-up attention useful for object recognition," in *Proc. CVPR*, 2004.

[26] H. Schneiderman, "Learning a restricted bayesian network for object detection," in *Proc. CVPR*, 2004.

[27] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. PAMI*, vol. 22, no. 8, August 2000.

[28] A. Singhal, J. Luo, and W. Zhu, "Probabilistic spatial context models for scene content understanding." in *Proc. CVPR*, 2003.

[29] A. Torralba, "Contextual priming for object detection," *IJCV*, vol. 53, no. 2, 2003.

[30] A. Torralba, K. P. Murphy, and W. T. Freeman, "Contextual models for object detection using boosted random fields," in *Proc. NIPS*, 2004.

[31] A. Torralba and P. Sinha, "Detecting faces in impoverished images," Tech. Rep., 2001.

[32] P. Viola and M. J. Jones, "Robust real-time face detection," *IJCV*, vol. 57, no. 2, 2004.

[33] P. Viola, M. J. Jones, and D. Snow, "Detecting pedestrians using patterns of motion and appearance," in *Proc. ICCV*, 2003.

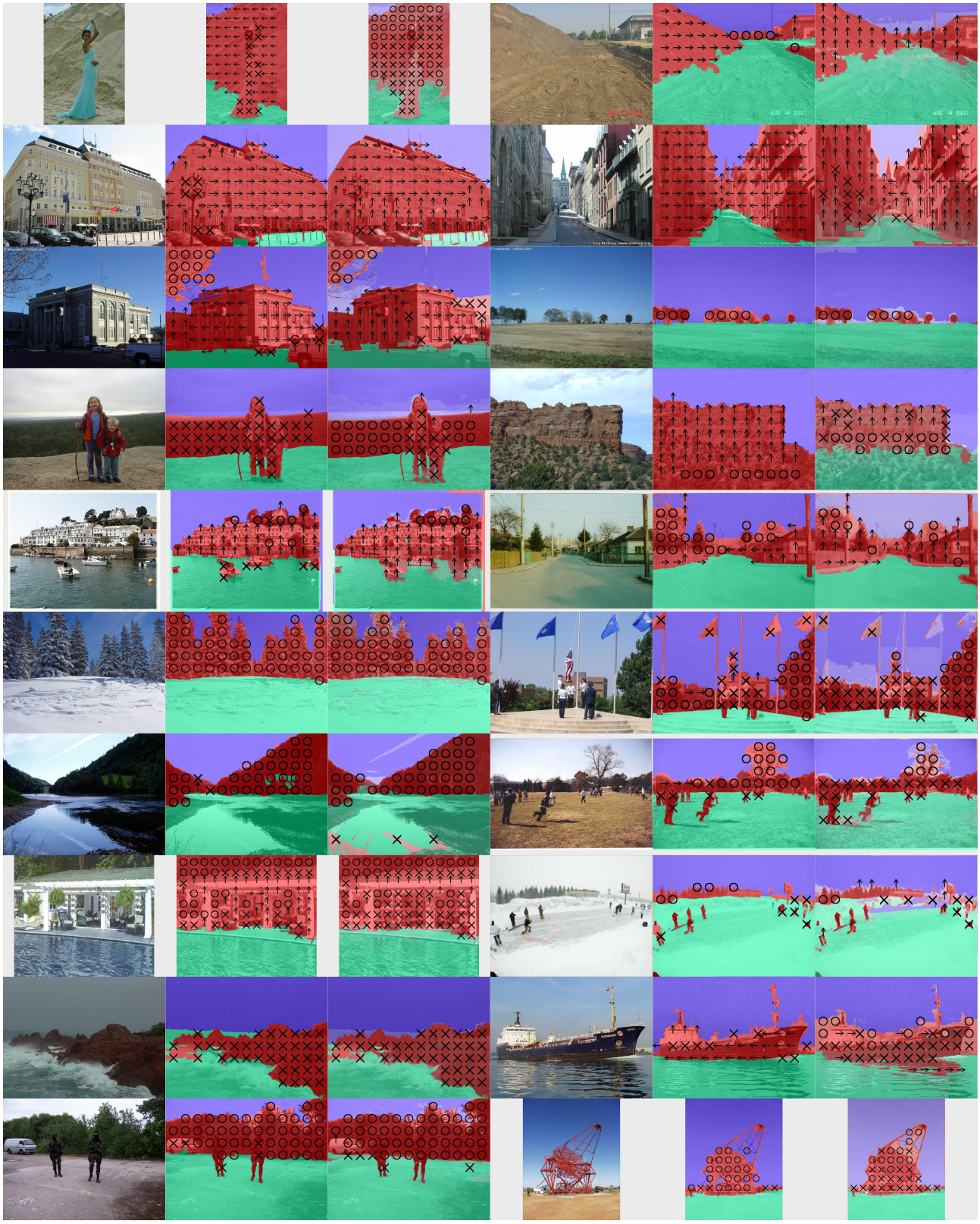


Figure 4: Results on images representative of our data set. Two columns of {original, ground truth, test result}. Colors indicate the main class label (green=ground, red=vertical, blue=sky), and the brightness of the color indicates the confidence for the assigned test labels. Markings on the vertical regions indicate the assigned subclass (arrows indicate planar orientations, "X"=non-planar solid, "O"=non-planar porous). Our system is able to estimate the geometric labels in a diverse set of outdoor scenes (notice how different orientations of the same material are correctly labeled in the top row). **This figure is best viewed in color.**

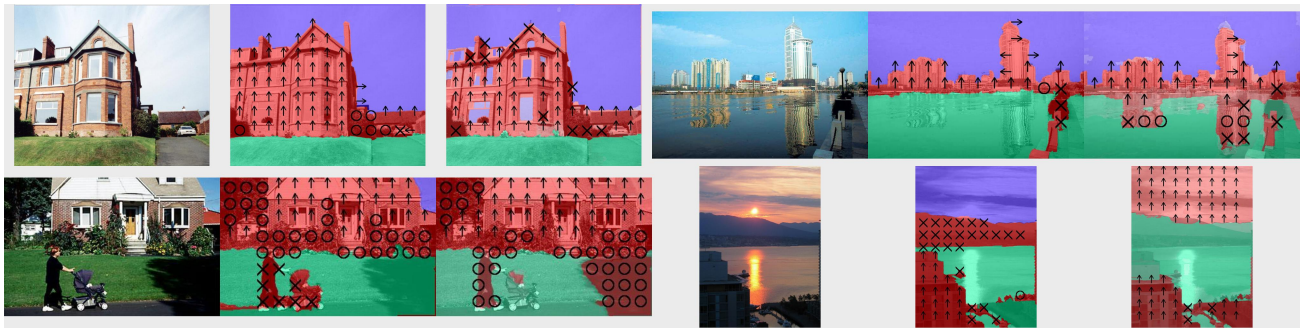
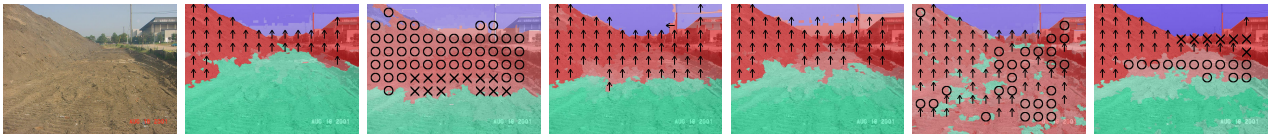
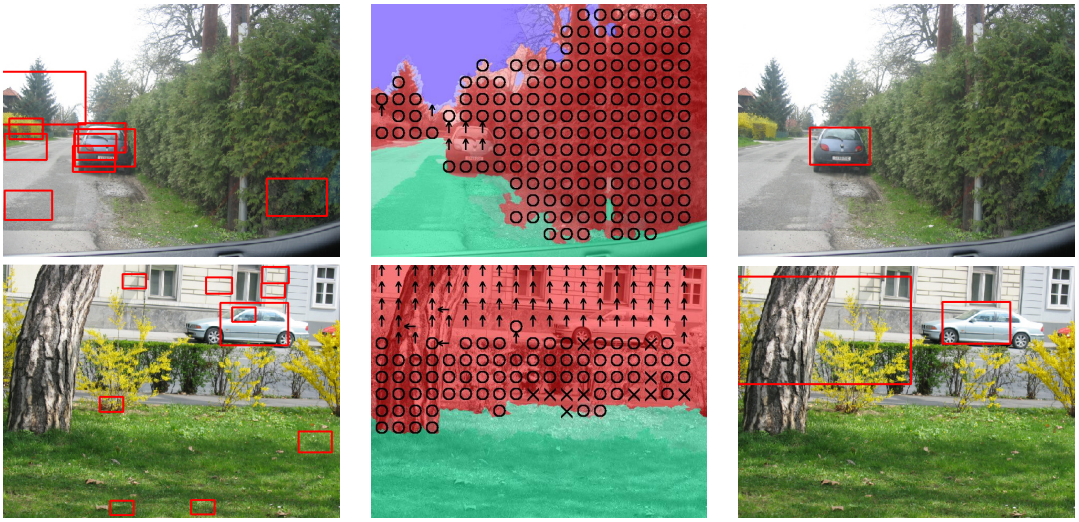


Figure 5: Failure examples. Two columns of {original, ground truth, test result}. Failures can be caused by reflections (top row) or shadows (bottom-left). At the bottom-right, we show one of the most dramatic failures of our system.



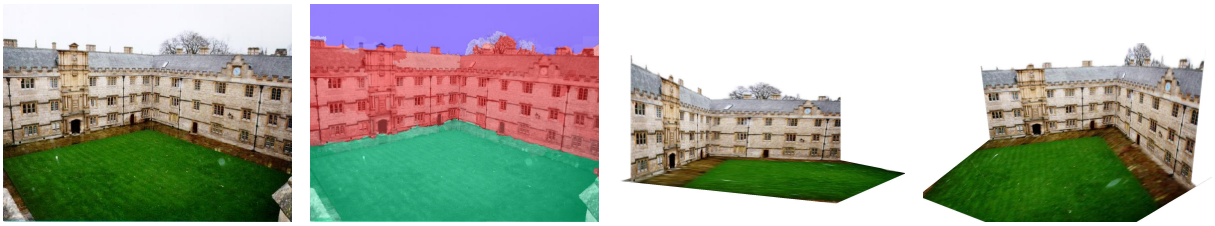
(a) Input (b) Full (c) Loc Only (d) No Color (e) No Texture (f) No Loc/Shp (g) No Geom

Figure 6: In difficult cases, every cue is important. When any set of features (d-g) is removed, more errors are made than when all features are used (b). Although removing location features (f) cripples the classifier in this case, location alone is not sufficient (c).



(a) Local Features Only (b) Geometric Labels (c) With Context

Figure 7: Improvement in Murphy *et al.*'s detector [20] with our geometric context. By adding a small set of context features derived from the geometric labels to a set of local features, we reduce false positives while achieving the same detection rate. For a 75% detection rate, more than two-thirds of the false positives are eliminated. The detector settings (e.g. non-maximal suppression) were tuned for the original detector.



Input Labels Novel View Novel View

Figure 8: Original image used by Liebowitz *et al.* [17] and two novel views from the scaled 3D model generated by our system. Since the roof in our model is not slanted, the model generated by Liebowitz, *et al.* is slightly more accurate, but their model is manually specified, while ours is created fully automatically [12]!