Geometric Context from a Single Image

Derek Hoiem Alexei A. Efros Martial Hebert

Carnegie Mellon University

February 26, 2009
Presented by Luis Guimbarda

Outline

Motivation

- The goal is to recover a 3D "contextual frame" from a single image.

- Global scene context is also important for object detection.[1][2]

---

[1] Antonio Torralba. Contextual priming for object detection. *Int. J. Comput. Vision*, 53(2):169–191, July 2003

[2] A. Torralba, K. P. Murphy, and W. T. Freeman. Contextual models for object detection using boosted random fields. In *Advances in Neural Information Processing Systems 17 (NIPS)*, pages 1401–1408, 2005

Approach

- 3D geometry estimation is treated as a statistical learning problem.
- The system models geometric classes that depend on the orientation of a physical scene.
    - For example, plywood lying on the ground and the same plywood propped by a board are in different geometric classes.
- The geometric structure is built progressively.

Observations on the training/testing data

- Over 97% of pixels belonged to one of three geometric classes:
  - the ground plane
  - surfaces roughly perpendicular to the ground
  - sky
- The camera axis was roughly parallel to the ground plane in most of the images.

Observations on the training/testing data

[3]

[3] from Derek Hoiem's presentation "Automatic Photo Popup", http://www.cs.uiuc.edu/homes/dhoiem/presentations/index.html

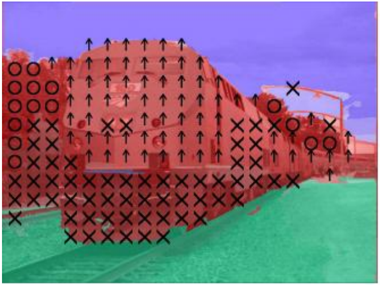
Overview of the Algorithm
Raw image

- Every patch of an image is induced by a surface with some orientation in the real world.
- All available cues are necessary to determine the most likely orientations.

- Each superpixel is assumed to belong to a single geometric class.
- To estimate the orientation of large-scale surfaces, it's necessary to compute more complex geometric features over large regions of the image.

Overview of the Algorithm
Multiple Hypotheses

- A small number of segmentations from all possible superpixel segmentations are sampled.
- The likelihood of each superpixel label is determined.



- There are 3 main geometric labels:
  - ground
  - vertical
  - sky
- And 5 subclasses of vertical:
  - left ($\leftarrow$)
  - center ($\uparrow$)
  - right ($\rightarrow$)
  - porous ($\bigcirc$)
  - solid ($\times$)

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

C1 captures the red, green and blue values, as expected

C2 represents the hue and "grayness" of a pixel

T1-4 Derivative of oriented Gaussian filters

Training Data

- 300 publicly available images from the Internet
- Images are often cluttered and span several environments.
- Each image is over-segmented, and each segment is labeled according to its geometric class.
- 50 images are used to train the segmentation algorithm.
- 250 image are used to train and test the system using 5-fold cross validation.

Generating Multiple Segmentations

- An image is to be segmented into n_r geometrically homogeneous (and not necessarily contiguous) regions.
- The superpixels are shuffled.
- The first n_r superpixels are assigned to different regions.
- Each of the remaining superpixels are iteratively assigned based on a learned pairwise affinity function.
- The algorithm was run with nine different values for n_r, ranging from 3 to 25.

Training the Pairwise Affinity Function

- Pairs of superpixels were sampled.
  - 2500 same-label pairs
  - 2500 different-label pairs
- The probability that two superpixels share a label given the absolute difference of their feature vectors is derived:

$$P\left(y_i = y_j \middle| \left|\mathbf{x}_i - \mathbf{x}_j\right|\right)$$

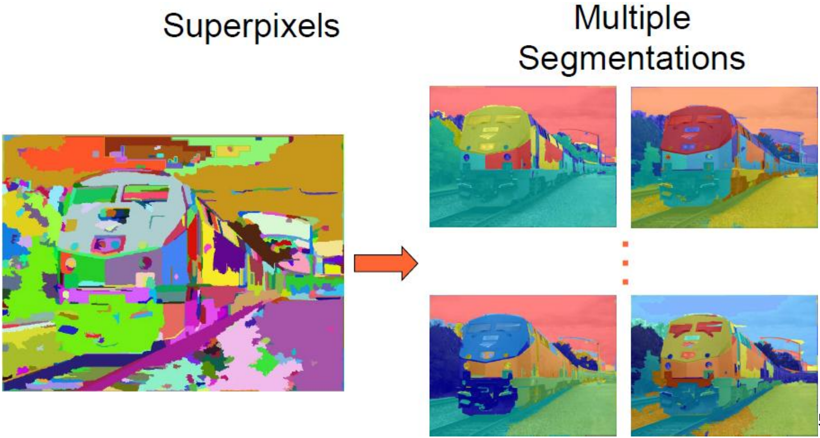
Training the Pairwise Affinity Function

- The pairwise likelihood function is estimated using the logistic regression form of Adaboost[4].
- Each weak learner f_m is based on the naive density estimates of the absolute feature differences:

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

[4] A. Criminisi, I. Reid, and A. Zisserman. Single view metrology. *International Journal of Computer Vision*, V40(2):123–148, November 2000

Training the Pairwise Affinity Function



[5]

[5] from Derek Hoiem's presentation "Automatic Photo Popup", http://www.cs.uiuc.edu/homes/dhoiem/presentations/index.html

Geometric Labeling

- Each superpixel will belong to several regions, one per hypothesis.
- The confidence of the superpixel label is the average label likelihood of the regions containing it, weighted by the homogeneity likelihoods:

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

Training the Label and Homogeneity Likelihood Functions

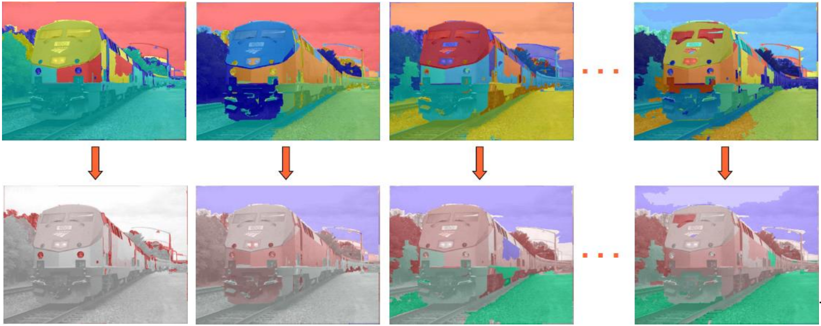
- Several segmented Hypotheses are generated as described above.
- Each region is labeled with one of the main geometric classes or "mixed".
- Each region that is "vertical" is labeled with one of the vertical subclasses or "mixed".

Training the Label and Homogeneity Likelihood Functions

- The label likelihood function is learned as one-versus-many.
- The homogeneity likelihood function is learned as mixed-versus-homogeneously labeled.
- Both functions are learned using the logistic regression form of Adaboost with weak learners based on eight-node decision trees[6].

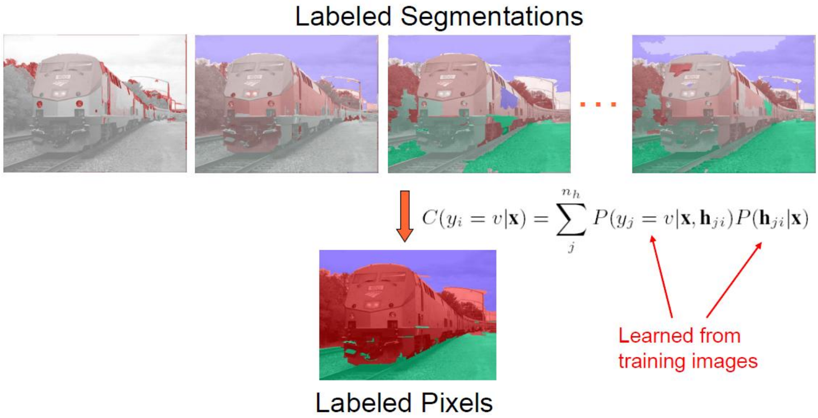
[6] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting, 1998

Training the Label and Homogeneity Likelihood Functions


[7]

[7] from Derek Hoiem's presentation "Automatic Photo Popup", http://www.cs.uiuc.edu/homes/dhoiem/presentations/index.html

Training the Label and Homogeneity Likelihood Functions



[8]

---

[8] from Derek Hoiem's presentation "Automatic Photo Popup", http://www.cs.uiuc.edu/homes/dhoiem/presentations/index.html

Geometric Classification

- The overall accuracy for main geometric classes was 86%.
- The overall accuracy for vertical subclasses was 52%.
- The difficulty of classifying vertical subclasses is mostly due to ambiguity of ground truth labeling.

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.

Importance of Structure Estimation

- Accuracy increases with the complexity of the intermediate structure estimation.

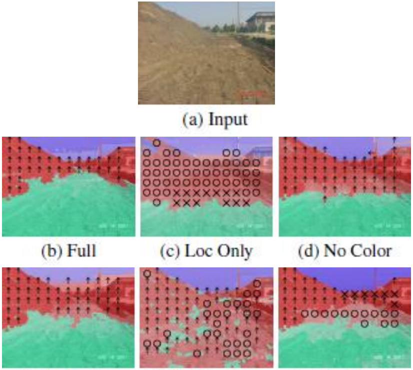
Intermediate Structure Estimation						
	CPrior	Loc	Pixel	SPixel	OneH	MultiH
Main	49%	66%	80%	83%	83%	86%
Sub	34%	36%	43%	45%	44%	52%

CPrior only class priors were used
Loc only pixel positions were used
Pixel only pixel-level colors and textures were used
SPixel all features are used at superpixel-level
OneH only used a single 9-segmented hypothesis
MultiH used the full multi-hypothesis framework

Importance of Cues

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

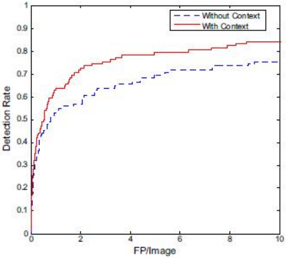
- Location features have the strongest effect on the system's accuracy.
- Location features aren't sufficient for classification.



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

Object Detection

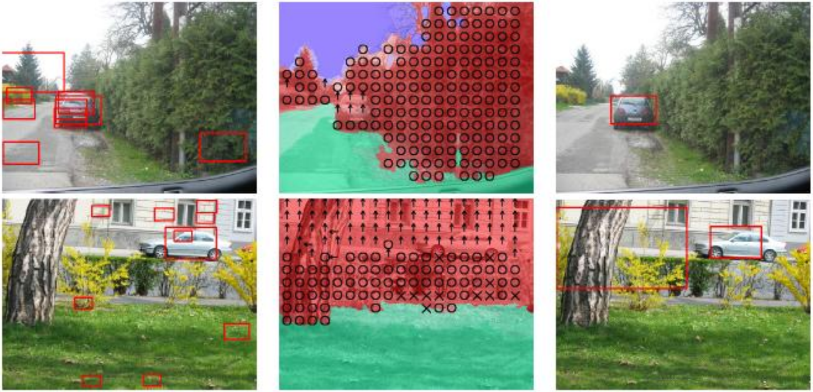
- Using a local detector[9] that uses GentleBoost to form a classifier based on fragment templates to detect multiple-oriented cars on the PASCAL[10] training set, sans grayscale images.
- One version of the system only used 500 local features, while the other added 40 contextual features form the geometric context.



[9]Kevin P. Murphy, Antonio B. Torralba, and William T. Freeman. Graphical model for recognizing scenes and objects. In Sebastian Thrun, Lawrence K. Saul, and Bernhard Schlkopf, editors, *NIPS*. MIT Press, 2003

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

Object Detection



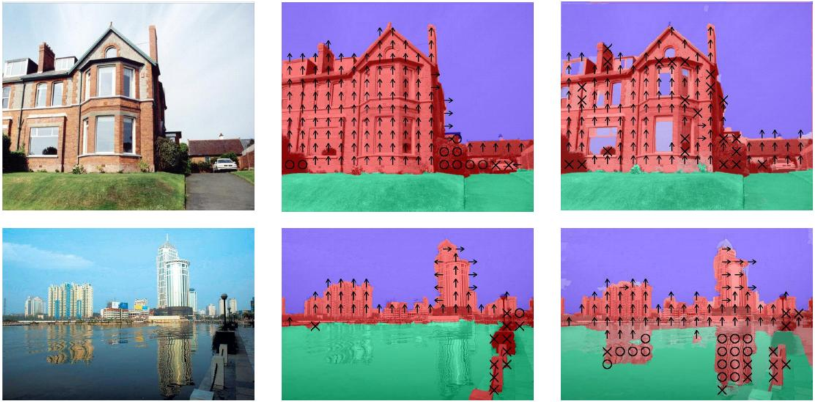
(a) Local Features Only (b) Geometric Labels (c) With Context

Automatic Single-View Reconstruction

- The automatically generated 3D model is comparable to the manually specified model[11].

Input Labels Novel View Novel View

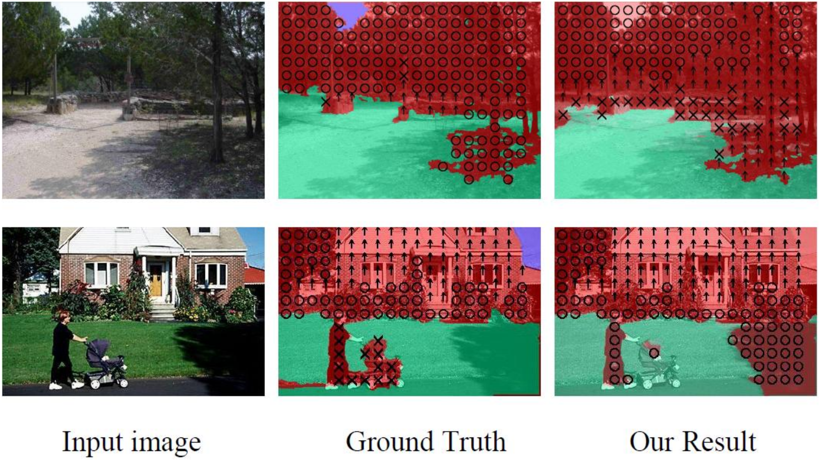
[11] D. Liebowitz, A. Criminisi, and A. Zisserman. Creating architectural models from images. *Computer Graphics Forum*, pages 39–50, September 1999



Input image Ground Truth Our Result [12]

[12] from Derek Hoiem's presentation "Automatic Photo Popup", http://www.cs.uiuc.edu/homes/dhoiem/presentations/index.html

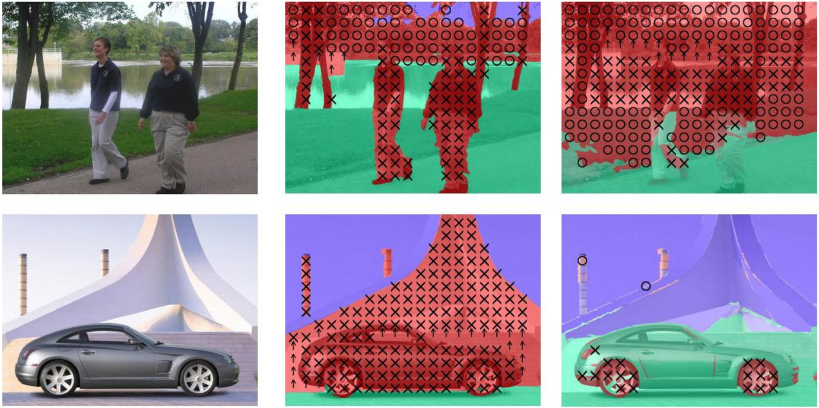
Failures
Shadow Failures



[13]

[13] from Derek Hoiem's presentation "Automatic Photo Popup", http://www.cs.uiuc.edu/homes/dhoiem/presentations/index.html

Failures
Catastrophic Failures



Input image   Ground Truth   Our Result [14]

[14] from Derek Hoiem's presentation "Automatic Photo Popup", http://www.cs.uiuc.edu/homes/dhoiem/presentations/index.html

[1] A. Criminisi, I. Reid, and A. Zisserman. Single view metrology. *International Journal of Computer Vision*, V40(2):123–148, November 2000.

[2] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting, 1998.

[3] D. Liebowitz, A. Criminisi, and A. Zisserman. Creating architectural models from images. *Computer Graphics Forum*, pages 39–50, September 1999.

[4] Kevin P. Murphy, Antonio B. Torralba, and William T. Freeman. Graphical model for recognizing scenes and objects. In Sebastian Thrun, Lawrence K. Saul, and Bernhard Schlkopf, editors, *NIPS*. MIT Press, 2003.

[5] A. Torralba, K. P. Murphy, and W. T. Freeman. Contextual models for object detection using boosted random fields. In *Advances in Neural Information Processing Systems 17 (NIPS)*, pages 1401–1408, 2005.

[6] Antonio Torralba. Contextual priming for object detection. *Int. J. Comput. Vision*, 53(2):169–191, July 2003.