Automatic Photo Popup

Derek Hoiem
Alexei A. Efros
Martial Hebert

Carnegie Mellon University
What Is Automatic Photo Popup
Introduction

• Creating 3D models from images is a complex process
  – Time-consuming
  – Special equipment
  – Large number of photographs
• Easy methods not widely accessible to general public
Related Work

• Multiple Images
  – Manual Reconstruction
    • Façade [Debevec et al. 1996], REALVIZ Image Modeler, Photobuilder [Cipolla et al. 1999], [Ziegler et al. 2003], etc.
  – Automatic Reconstruction
    • Structure from Motion (e.g. [Nister 2001], [Pollefeys et al. 2004])

• Single Image
  – Manual Reconstruction
    • Metric ([Liebowitz et al. 1999], [Criminisi et al. 1999])
    • Approximate: Tour into the Picture [Horry et al. 1997]; [Kang et al. 2001], [Oh et al. 2001], [Zhang et al. 2001], etc.
  – Automatic Reconstruction
Our Goals

• Simple, piecewise planar models
• Outdoor scenes
• Doesn’t need to work all the time (~35%)
The Problem

- Recovering 3D geometry from **single** 2D projection

- Infinite number of possible solutions!

from [Sinha and Adelson 1993]
Intuition

• From a single photograph, humans can fairly easily
  – Grasp the overall structure of a scene
  – Imagine reasonably well how the scene would appear from a different perspective

• How?
  – What is it about our world that allows us to do so?
Structure In Our World

- Gravity
  - Flat ground
  - Objects (trees, buildings) meet the ground at roughly right angles
  - Ground is below sky
- Materials correspond to 3D orientations
  - e.g. bark is usually on a vertical surface
- Structures exhibit high degree of similarity
  - e.g. grass is usually green
Solving the Problem

- Exploit structure, eliminate unlikely geometrical interpretations
- Use training data to learn correspondences between image features and geometric classes
Basic Idea

- Assign geometric labels to regions of the image
- Use the labeled regions to infer a 3D model
Overview

- Image to Superpixels
- Superpixels to Multiple Constellations
- Multiple Constellations to Superpixel Labels
- Superpixel Labels to 3D Model
Superpixels

- Without knowledge of the scene’s structure, we can only compute simple features
  - Pixel colors
  - Filter responses
- Superpixels enable us to
  - Compute more complex statistics
  - Improve the efficiency of the algorithm
  - Locate regions with the same geometric label
  - Reduce noise
Superpixels to Multiple Constellations

- Superpixels do not provide enough spatial support
- Group together superpixels that are likely to be from the same surface
  - Call each group a *constellation*
- Ideally, each constellation corresponds to one physical object in the scene
  - Cannot guarantee that a constellation describes a single object
  - Contained superpixels may not have the same label
- Generate several overlapping sets of possible constellations
  - Use this to help determine the final labels for superpixels
Forming Constellations

- Group superpixels that are likely to share a common geometric label
- We cannot be completely confident that these constellations are *homogenous*
  - Find multiple constellations for each superpixel
- Superpixel features allow us to determine whether a pair of superpixels have the same label
  - Not sufficient to determine the label for the superpixel
Forming Constellations

- Assign a superpixel randomly to each of $N_c$ constellations
- Then iteratively assign each remaining superpixel to the constellation most likely to share its label
- Want to maximize the average pairwise log-likelihoods with other superpixels in the constellation

\[
S(C) = \sum_{k}^{N_c} \frac{1}{n_k (1 - n_k)} \sum_{i,j \in C_k} \log P(y_i = y_j \mid \| \mathbf{z}_i - \mathbf{z}_j \|)
\]

Estimated prob. that two superpixels share the same label
Forming Constellations

• Vary $N_c$ from 3 to 25 to explore
  – The trade-off of variance from poor spatial support
  – The bias from risk of mixed labels in a constellation
Features for Geometric Classes

• When looking at a region \( r \), what characteristics can help us determine the label for \( r \)?
  
  – Color
  
  – Texture
  
  – Location and Shape
  
  – Perspective
Features: Color

- Color helps identify material
  - Sky is usually blue or white
  - Ground is usually brown or green

- RGB
  - blueness or greenness

- HSV
  - perceptual attributes (hue and “darkness”)
Features: Texture

• Additional information about the material
  – Distinguish sky from water, green leaves from grass

• Derivative of oriented Gaussian filters provides texture cues

• Textons also provide texture cues
Features: Location

- Location provides cues for distinguishing between ground, vertical structures and sky.
Features: Perspective

- Perspective helps determine 3D orientation of surfaces
  - Use region boundaries to compute vanishing points
Many Weak Geometric Cues

- Material
- Image Location
- 3D Geometry

<table>
<thead>
<tr>
<th>Feature Descriptions</th>
<th>Num</th>
<th>Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Color</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C1. RGB values: mean</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>C2. HSV values: conversion from mean RGB values</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>C3. Hue: histogram (5 bins) and entropy</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>C4. Saturation: histogram (3 bins) and entropy</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Texture</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T1. DOOG Filters: mean abs response</td>
<td>12</td>
<td>3</td>
</tr>
<tr>
<td>T2. DOOG Filters: mean of variables in T1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>T3. DOOG Filters: id of max of variables in T1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>T4. DOOG Filters: (max - median) of variables in T1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>T5. Textons: mean abs response</td>
<td>12</td>
<td>7</td>
</tr>
<tr>
<td>T6. Textons: max of variables in T5</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>T7. Textons: (max - median) of variables in T5</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Location and Shape</td>
<td></td>
<td></td>
</tr>
<tr>
<td>L1. Location: normalized x and y, mean</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>L2. Location: norm. x and y, 10th and 90th percentile</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>L3. Location: norm. y wrt horizon, 10th and 90th pctl</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>L4. Shape: number of superpixels in constellation</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>L5. Shape: number of sides of convex hull</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>L6. Shape: num pixels/area/(convex hull)</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>L7. Shape: whether the constellation region is contiguous</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3D Geometry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>G1. Long Lines: total number in constellation region</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>G2. Long Lines: % of nearly parallel pairs of lines</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>G3. Line Intersection: hist. over 12 orientations, entropy</td>
<td>13</td>
<td>11</td>
</tr>
<tr>
<td>G4. Line Intersection: % right of center</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>G5. Line Intersection: % above center</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>G6. Line Intersection: % far from center at 8 orientations</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>G7. Line Intersection: % very far from center at 8 orienta</td>
<td>8</td>
<td>5</td>
</tr>
<tr>
<td>G8. Texture gradient: x and y “edginess” (T2) center</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>
Multiple Constellations to Superpixel Labels

• The greatest challenge is to determine the geometric label of an image region
  – Machine learning approach
  – Model the appearance of geometric labels with training images

• For each constellation, we estimate
  – The likelihood of each of the three possible geometric labels
  – The confidence that all contained superpixels have the same label

• Superpixel’s label is inferred from the likelihoods of the constellations that contain that superpixel
Multiple Constellations to Superpixel Labels
Labeling the Image

- Feature descriptors are computed from
  - Superpixels
  - Constellations
- Geometric classification of regions
- Training
  - Training Data
  - Superpixel Same-Label Likelihoods
  - Constellation Label and Homogeneity Likelihoods
Geometric Classification

• Goal: obtain the most likely label for each superpixel in the image
• Label likelihood – confidence in each geometric label
• Homogeneity likelihood – do all superpixels in a constellation share the same label?
**Geometric Classification**

- Estimate the likelihood of a superpixel label with

\[
P(y_i = t | \mathbf{x}) = \sum_{k: s_i \in C_k} P(y_k = t | \mathbf{x}_k, C_k) P(C_k | \mathbf{x}_k)
\]

- \(s_i\) – the \(i^{th}\) superpixel
- \(C_k\) – the \(k^{th}\) constellation
- \(y_i\) – the label of \(s_i\)
- \(y_k\) – the label of \(C_k\)
- \(x_k\) – the feature vector for \(C_k\)
Training

- Data: 82 images of outdoor scenes
  - Natural, urban and suburban scenes
Superpixel Same-Label Likelihoods

- Sample 2500 superpixel pairs
  - Same-label and different-label
- Use kernel density estimation [Duda et al. 2000] and AdaBoost [Collins et al. 2002] to estimate the likelihood function

\[ P(y_i = y_j \mid \| \mathbf{z}_i - \mathbf{z}_j \|) \]
Learn From Training Images

Homogeneity Likelihood

\[ P(C_k \mid x_k) \]

Label Likelihood

\[ P(y_i = y_j \mid \| \mathbf{z}_i - \mathbf{z}_j \|) \]

• Prepare training images
  – Create multiple segmentations of training images
  – Get segment labels from ground truth
    • Ground, vertical, sky or “mixed”

• Density estimation by decision trees
  – Improved accuracy with logistic regression form of AdaBoost [Collins et al. 2002]
Superpixel Labels to 3D Model

• Once we have the geometric labels for each superpixel, we can estimate the position of each object
  – Estimate ground plane
  – Fit the boundary of the bottom of the vertical regions with the ground
  – Estimate horizon position from geometric features and ground labels
• Vertically labeled regions become planar billboards
• Remove regions labeled sky from model
• Texture-map image onto model
Vanishing Points of Surfaces

• Compute intersections for long lines in a constellation
  – Video Compass [Kosecka and Zhang 2002]
Horizon Estimation

- Find the position that minimizes the $L_{1/2}$ distance from all of the intersection points in the image
  - Robustness to outliers
  - Provides a reasonable estimate of the horizon in man-made images
Creating the 3D Model

- Cutting and folding
- Camera parameters
Cutting And Folding

• Construct a simple 3D model
  – Single ground plane
  – Simple, piecewise planar objects for vertically labeled regions

• Partition vertical regions into a set of objects

• Determine where each object intersects the ground plane
  – There are difficulties, be conservative
Partitioning Vertically Labeled Regions

- Divide vertically labeled pixels into loosely connected regions using the connected components algorithm.
- For each region, fit a set of lines to the boundary using the Hough transform [Duda et al. 1972].
- Form disjoint line segments into a set of polylines.
Cutting and Folding

- Fit ground-vertical boundary
  - Iterative Hough transform
Cutting and Folding

- Form polylines from boundary segments
  - Join segments that intersect at slight angles
  - Remove small overlapping polylines
- Estimate horizon position from perspective cues
Cutting and Folding

- ``Fold'' along polylines and at corners
- ``Cut'' at ends of polylines and along vertical-sky boundary
Cutting and Folding

- Construct 3D model
- Texture map
Results

Input Image

Cut and Fold

Automatic Photo Pop-up
Results

Input Image

Automatic Photo Pop-up
Results

Input Images

Automatic Photo Pop-up
Results

Input Image

Automatic Photo Pop-up
Comparison with Manual Method

Input Image

Automatic Photo Pop-up (30 sec)!

[Liebowitz et al. 1999]
Failures

Labeling Errors
Failures

Foreground Objects
The Music Video
Just the first step…

• Richer set of geometric classes (ICCV 05 paper)
• Segment foreground objects
• Cleaner segmentation
• More accurate labeling
Conclusion

• First system to automatically recover 3D scene from single image!

• Learn statistics of our world from training images
Quantitative Results (ICCV 05)

Average: 86%

<table>
<thead>
<tr>
<th>Geometric Class</th>
<th>Ground</th>
<th>Vertical</th>
<th>Sky</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ground</td>
<td>0.78</td>
<td>0.22</td>
<td>0.00</td>
</tr>
<tr>
<td>Vertical</td>
<td>0.09</td>
<td>0.89</td>
<td>0.02</td>
</tr>
<tr>
<td>Sky</td>
<td>0.00</td>
<td>0.10</td>
<td>0.90</td>
</tr>
</tbody>
</table>
Questions?