Recovering Surface Layout from an Image

Outline
- Introduction
- Features
- Superpixels & Multiple Segmentations
- Learning and Inference
- Experiments
- Conclusion

Introduction

- How do we perceive an image?

What can modern computer vision achieve:
- Face detection
- Object detection (bicycles, cars, rocks etc.)
- Find shadows
- Segment the image based on color, texture, or contours
Introduction

- Is it possible to estimate 3D layout from a single image?
- Previous works (early days CV):
  - to provide a complete semantic interpretation of an input image by reasoning about the 3D scene that generated it

- More recent works (time of this paper):
  - Use several images to estimate 3D layout of an image

*“Simple Shadow Removal”* Fredembach et al. ICPR 2006

*“Contour continuity in region-based image segmentation”* Leung et al. ECCV'98

“Blocks world” in early days of AI: [https://en.wikipedia.org/wiki/Blocks_world](https://en.wikipedia.org/wiki/Blocks_world)
Introduction

- Goals of this paper:
  - Estimate a rough surface layout of a scene
  - Using only a single image
  - "Discretize" geometric information
  - With tools of statistical learning

- Pose surface layout recovery as a recognition problem
- Label an image using several geometric classes
- Assumptions:
  - Nearly all pixels (over 97%) belong to:
    - Horizontal (support) surfaces
    - Nearly vertical surfaces
    - Sky

Example
- Main classes:
  - Support
  - Vertical
  - Sky
- Vertical Subclasses:
  - Left
  - Center
  - Right
  - Porous
  - Solid

Assumptions:
- the camera axis is roughly aligned with the ground plane

Example
- Main classes:
  - Support
  - Vertical
  - Sky
- Vertical Subclasses:
  - Left
  - Center
  - Right
  - Porous
  - Solid
Introduction

Support Surfaces:
- Are roughly parallel to the ground
- Could potentially support a solid object
- For example:
  - road surfaces
  - lawns
  - dirt paths
  - lakes
  - and table tops

Vertical Surfaces:
- Too steep to support an object
- For example:
  - walls
  - cliffs
  - the curb sides
  - people
  - trees
  - or cows

Important notes:
- Ground truth labeling is subjective
- For example:
  - Is the side of a small car planar?
  - Is a large branch solid?
- Thus causing ambiguities...
- Some quantitative errors are due to these ambiguities

Outline

Introduction
- Features
- Superpixels & Multiple Segmentations
- Learning and inference
- Experiments
- Conclusion

Features

Cues for Labeling Surfaces:
Features

Location:
- xy coordinates in image are good cues for:
  - Rough 3D geometry, for example:
    - sky is high in the image & ground is low
    - Planes facing left tend to be on the right & vice versa:
      - Because a lot of photographs are taken facing down a street or path.

Features

Color:
- good cue for materials and objects that correspond to geometric classes
  - Sky is usually blue or white
  - support segments are often green (grass) or brown (dirt)
  - RGB is used to measure “blueness” or “greenness” of a segment
  - HSV measures attributes such as hue and “grayness”

Features

Texture: same as color+ surface perspective
- a vertical plane will tend to have more vertically oriented textures
- Same for horizontal plane
- Represented through subset of the filter bank designed by Leung and Malik

Leung and Malik (2001) Representing and recognizing the visual appearance of materials using three-dimensional textons
Features

- Perspective:

- Statistics of straight lines:
  - Gives information about the vanishing points of a surface without explicitly computing them
  - Computing explicit estimate of the vanishing points can help too

- Perspective features:
  - P1. Long Lines: (number of line pixels)/\sqrt{\text{area}}
  - P2. Long Lines: percent of nearly parallel pairs of lines
  - P3. Line Intersections: histogram over 8 orientations, entropy
  - P4. Line Intersections: percent right of image center
  - P5. Line Intersections: percent above image center
  - P6. Line Intersections: percent far from image center at 8 orientations
  - P7. Line Intersections: percent very far from image center at 8 orientations
  - P8. Vanishing Points: (num line pixels with vertical VP membership)/\sqrt{\text{area}}
  - P9. Vanishing Points: (num line pixels with horizontal VP membership)/\sqrt{\text{area}}
  - P10. Vanishing Points: percent of total line pixels with vertical VP membership
  - P11. Vanishing Points: x-pos of horizontal VP—segment center (0 if none)
  - P12. Vanishing Points: y-pos of highest/lowest vertical VP wrt segment center
  - P13. Vanishing Points: segment bounds wrt horizontal VP
  - P14. Gradient: x, y center of mass of gradient magnitude wrt segment center

Outline

- Introduction
- Features
- Superpixels & Multiple Segmentations
- Learning and inference
- Experiments
- Conclusion

Superpixels & Multiple Segmentations

- Image is a 2d array of pixels (atomic unit)
- Amount of information on each pixel is limited
- Superpixels are subsets of pixels
- Superpixels as atomic units
- Relationship between Superpixels = Slightly more complex statistics can be computed
Superpixels & Multiple Segmentations

- Graph-based oversegmentation technique of Felzenszwalb and Huttenlocher
- Groups large homogeneous regions of the image together
- Dividing heterogeneous regions into many smaller superpixels

Does superpixels produce enough statistics?

- No! Large segments are needed sometimes
- Statistics over parts of different sizes can help
- Solution: Multiple Segmentations
- Generate segmentations of varying sizes
- Compute statistics over them
Outline

- Introduction
- Features
- Superpixels & Multiple Segmentations
- Learning and inference
- Experiments
- Conclusion

Learning and inference

Training Overview
1. For each training image:
   (a) Compute superpixels
   (b) Compute superpixel cases (Table 1)
2. Train same-label classifier (Section 6)
3. For each training image:
   (a) Produce multiple segmentations for varying $w$ (Section 5)
   (b) Label each segment according to ground truth
   (c) Compute cases in each segment (Table 1)
4. Train segment label classifier and homogeneity classifier (Section 6)

Learning and inference

- Labeling:
  - Estimation of segments labels
  - $P(y_i = k | I)$ likelihood of $i$th SP having label $k$
  - Estimation of segments homogeneity
  - A segment is homogenous if all $i$ SP have the same label
  - In practice, if 95% of GT SP are labeled the same
  - Else segment is labeled “mixed”
  - $P(s_j | I)$ likelihood of $j$th segment being homogenous

Learning and inference

- If a segment is homogenous then we can estimate:
  - $P(y_i = k | I)$ likelihood of segments label
  - $P(s_j = k | I)$ is estimated from $P(\xi_j | I)$ by marginalization:
    \[ P(y_i = k | I) = \sum_{\xi_j} P(\xi_j | I) P(y_i = k | \xi_j, I) \]
  - Marginalize over the unique sampled segments $s_j$ that contain the superpixel

Learning and inference

- If a segment is homogenous then we can estimate:
  - $P(y_i = k | I)$ likelihood of segments label
  - $P(s_j = k | I)$ is estimated from $P(\xi_j | I)$ by marginalization:
    \[ P(y_i = k | I) = \sum_{\xi_j} P(\xi_j | I) P(y_i = k | \xi_j, I) \]
  - Marginalize over the unique sampled segments $s_j$ that contain the superpixel
Learning and inference

- How are the classifiers trained?
  - Boosted decision trees using logistic regression version of Adaboost

Learning and inference

Learning and inference

Learning and inference

Learning and inference

Learning and inference

Outline

- Introduction
- Features
- Superpixels & Multiple Segmentations
- Learning and inference
- Experiments
- Conclusion
Experiments

- Average accuracy for increasing levels of spatial support

<table>
<thead>
<tr>
<th>Method</th>
<th>Mean</th>
<th>Std</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single superpixel clusters</td>
<td>82.1</td>
<td>46.5</td>
</tr>
<tr>
<td>Multiple superpixels</td>
<td>86.2</td>
<td>33.6</td>
</tr>
<tr>
<td>Ground truth segmentation</td>
<td>88.1</td>
<td>60.5</td>
</tr>
</tbody>
</table>

Table 4: Average accuracy percent of correctly labeled images across methods using varying levels of spatial support.

Accuracy increases with more data.

Peaks accuracy is at 100 segments, with larger numbers of segments degrading the detection accuracy.

Increasing the number of superpixels further than 8 produces no significant change in accuracy.

Complexity tradeoff between map density and the homogeneity of boundary segments.

Adaboost is robust to overfitting complexity trade-off between the power of the classifier and its tendency to overfit.

Peaks accuracy is at 100 segments, with larger numbers of segments degrading the detection accuracy.

Peaks accuracy is at 100 segments, with larger numbers of segments degrading the detection accuracy.

Peaks accuracy is at 100 segments, with larger numbers of segments degrading the detection accuracy.

Outline

- Introduction
- Features
- Superpixels & Multiple Segmentations
- Learning and inference
- Experiments
- Conclusion

Conclusion

- Novel way to estimate 3d layout from a single image
- Using different kind of cues to classify geometry
- Treating continues domain as a discrete recognition problem
Conclusion

Questions?