Training

1. For each training image:
   (a) Compute superpixels (Sec 4.1)
   (b) Compute superpixel features (Table 1)
2. Estimate pairwise-likelihood function (Eq 3)
3. For each training image:
   (a) Form multiple sets of constellations for varying $N_c$ (Sec 4.2)
   (b) Label each constellation according to superpixel ground truth
   (c) Compute constellation features (Table 1)
4. Estimate constellation label and homogeneity likelihood functions (Sec 4.3)

$$f_m(\mathbf{z}_1, \mathbf{z}_2) = \sum_i^{n_f} \log \frac{P_m(y_1 = y_2, |z_{1i} - z_{2i}|)}{P_m(y_1 \neq y_2, |z_{1i} - z_{2i}|)}$$
Training

1. For each training image:
   (a) Compute superpixels (Sec 4.1)
   (b) Compute superpixel features (Table 1)
2. Estimate pairwise-likelihood function (Eq 3)
3. For each training image:
   (a) Form multiple sets of constellations for varying $N_c$ (Sec 4.2)
   (b) Label each constellation according to superpixel ground truth
   (c) Compute constellation features (Table 1)
4. Estimate constellation label and homogeneity likelihood functions
   (Sec 4.3)
Algorithm Overview

- Segment & Classify
  - Find “Superpixels”
  - Classify “Superpixels” into groups (“Constellations”)
  - Classify “Constellations” into “Labels”

- Transform
  - Create 3D model
  - Map texture

3D Model

- “Cutting & Folding”
3D Model

Partition the pixels labeled as vertical into connected regions
For each connected vertical region:
1. Find ground-vertical boundary (x,y) locations p
2. Iteratively find best-fitting line segments until no segment contains
   more than \( m_p \) points:
   (a) Find best line \( L \) in \( p \) using Hough transform
   (b) Find largest set of points \( p_j \in p \) within distance \( d_j \) of \( L \) with no
       gap between consecutive points larger than \( g \)
   (c) Remove \( p_j \) from \( p \)
3. Form set of polyline from line segments
   (a) Remove smaller of completely overlapping (in x-axis) segments
   (b) Sort segments by median of points on segment along x-axis
   (c) Join consecutive intersecting segments into polyline if the inter-
       section occurs between segment medians
   (d) Remove smaller of any overlapping polylines
Fold along polylines
Cut upward from polylines endpoint. at ground-sky and vertical-sky boundaries
Project planes into 3D coordinates and texture map

Camera Parameters

- Where is the camera located (extrinsic parameters)?
  - Estimate horizon position

- What are the properties of the camera (intrinsic parameters)?
  - Set to constants (skew, affine ratio, FOV, camera height)
Horizon Position Estimation

- Find nearly parallel lines
- Compute their intersections (find vanishing points)
- Estimate horizon position by minimizing $L^{1/2}$-distance from all intersections

$$[\sqrt{x_1} + \sqrt{x_2}]^2$$
3D Model

- Ground projected into 3D using horizon estimate
- “Polylines” determine where ground/vertical boundaries lie
- Height of vertical sections determined from image and parameters
- Vertical sections then “popped up” to appear 3D

Results

- 800x600 image: 1.5 min
- 30% accuracy for outdoor scenes
Results – Errors

- Labeling error
- Polyline fitting error
- Modeling assumptions
- Occlusions
- Poor horizon estimation

Successes
Successes – Video

Further work

- Obvious failures:
  - Crowds, trees, tilted surfaces, multiple ground planes, occlusions
- Increase number of labels
  - Move to more complex (indoor) scenes
- Improve training/segmentation
  - Reduce labeling errors
Further work

- Estimate orientation of vertical regions
  - ICCV Follow-up

Geometric Context

<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>

Table 2: Confusion matrix for the main geometric classes.

<table>
<thead>
<tr>
<th>Vertical Subclass</th>
<th>Left</th>
<th>Center</th>
<th>Right</th>
<th>Porous</th>
<th>Solid</th>
</tr>
</thead>
<tbody>
<tr>
<td>Left</td>
<td>0.15</td>
<td>0.46</td>
<td>0.04</td>
<td>0.15</td>
<td>0.21</td>
</tr>
<tr>
<td>Center</td>
<td>0.02</td>
<td>0.55</td>
<td>0.06</td>
<td>0.19</td>
<td>0.18</td>
</tr>
<tr>
<td>Right</td>
<td>0.03</td>
<td>0.38</td>
<td>0.21</td>
<td>0.17</td>
<td>0.21</td>
</tr>
<tr>
<td>Porous</td>
<td>0.01</td>
<td>0.14</td>
<td>0.02</td>
<td>0.76</td>
<td>0.08</td>
</tr>
<tr>
<td>Solid</td>
<td>0.02</td>
<td>0.20</td>
<td>0.03</td>
<td>0.26</td>
<td>0.50</td>
</tr>
</tbody>
</table>

Table 3: Confusion matrix for the vertical structure subclasses.
Geometric Context

- Could also prove helpful in object recognition

And finally

- the end
- questions?
- answers?