Data-Driven 3D Voxel Patterns for Object Category Recognition

Yu Xiang, Wongun Choi, Yuanqing Lin, and Silvio Savarese

Adam Allevato
April 6, 2016
CS381V: Visual Recognition, UT Austin
Outline

• Problem Statement
• Related Work
• Approach
  • Data Representation
  • Training
  • Testing
• Results
  • Examples
  • Quantitative
• Discussion
Problem Statement

How can we build a system that can classify, locate, and orient occluded 3D objects using 2D image inputs?
Related Work

How can we build a system that can classify, locate, and orient occluded 3D objects using 2D image inputs?

<table>
<thead>
<tr>
<th>Method</th>
<th>Locate in 3D</th>
<th>Orient in 3D</th>
<th>Occlusion</th>
<th>2D Input</th>
</tr>
</thead>
<tbody>
<tr>
<td>2D Detection</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>3D Pose Estimation</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Point-Cloud Based Methods</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>This Paper: 3D Voxel Patterns</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
## Related Work: 2D Detection

<table>
<thead>
<tr>
<th>Method</th>
<th>Locate</th>
<th>Orient</th>
<th>Occlusion</th>
<th>2D Input</th>
</tr>
</thead>
<tbody>
<tr>
<td>2D Detection (DPM)</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

- **Deformable Part Models**
  - Felzenswalb et al. 2010

- **Face Detector**
  - Viola and Jones 2004
Related Work: 3D Pose Estimation

<table>
<thead>
<tr>
<th>Method</th>
<th>Locate</th>
<th>Orient</th>
<th>Occlusion</th>
<th>2D Input</th>
</tr>
</thead>
<tbody>
<tr>
<td>3D Pose Estimation</td>
<td>❌</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
</tbody>
</table>

3D2PM
Pepik et al. 2012

3D Category Classification
Savarese and Fei-Fei 2007
## Related Work: Point-Cloud Based Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Locate</th>
<th>Orient</th>
<th>Occlusion</th>
<th>2D Input</th>
</tr>
</thead>
<tbody>
<tr>
<td>Point-Cloud Based Methods</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✗️</td>
</tr>
</tbody>
</table>

### Clustersed Viewpoint Feature Histogram

Aldoma 2011

### Hashed 3D Voting

Hodaň et al. 2015
Data-Driven 3D Voxel Patterns
Approach: Data Representation

- **3D Voxel Patterns (3DVPs)**
  - Capture “patterns of visibility”
  - Composed of four parts:
    - 2D object image
    - 2D segmentation mask
    - 3D voxel model
    - Metadata: pose, 3D model

- Voxels and pixels can be
  - Visible (green)
  - Occluded (red)
  - Truncated (cyan)
  - Self-occluded (blue)
Approach: Training

1. Align 2D images with 3D CAD models
2. 3D voxel exemplars
3. 3D voxel patterns
4. Training 3D voxel pattern detectors

Slide Credit: Yu Xiang
1. Align 2D Images with 3D CAD Models

Technically, this is aligning 3D images with 3D CAD models: KITTI includes 3D bounding boxes. Occlusion and illumination changes in the source data are OK because of this.

The CAD models are handpicked from the Trimble 3D Warehouse (3dwarehouse.sketchup.com)
2. Building Voxel Exemplars: Baby 3DVPs

Depth ordering

truncated
visible
occluded

2D mask labeling

3D CAD model

Voxelization

self-occluded

truncated
visible
occluded

3D voxel model

Slide Credit: Yu Xiang
2. Building Voxel Exemplars: Baby 3DVPs

A 3D voxel exemplar $E_i = (I_i, M_i, V_i)$ + metadata

Metadata includes 3D CAD model (classification) and 3D pose

57,224 voxel exemplars found in KITTI
3. Discovering 3D Voxel Patterns

- First, generate mirror-image voxel patterns to increase training set
- Similarity metric for clustering:

$$s(\mathbf{x}_1, \mathbf{x}_2) = \frac{|S|}{N^3} \sum_{i=1}^{N^3} \mathbb{1}(x_1^i = x_2^i) \cdot w(x_1^i),$$

s.t., \( \sum_{i=0}^{|S|-1} w(i) = 1, \)

- Flexibility provided by \( w(i) \), but authors use \( w(i) = \{\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}\} \).
- Gives a “flat” similarity: evaluates agreement between voxel labels
3. Discovering 3D Voxel Patterns

3DVPs are "clustered exemplars" and have the same data structure.
4. Training 3D Voxel Pattern Detectors
4. Training 3D Voxel Pattern Detectors

- Train a detector for each 3DVP using Aggregated Channel Features (ACF): Blur the image, then split into channels and downsample 4x

- Channels in $\Omega$: gradients, HoG features, and LUV channels.

- Use boosting on pixels selected from all channels at once to build discriminative trees

- Realtime algorithm developed in 2014, over 200 citations

Approach: Testing

1. Apply 3DVP detectors

2D detection

2. Transfer meta-data
3. Occlusion reasoning

4. Backproject to 3D

3D localization

2D segmentation
1. Apply 3DVP Detectors

Testing images are 2D only

Slide Credit: Yu Xiang
2. Collect and Apply Metadata
2. Collect and Apply Metadata
3. Occlusion Reasoning

\[ E = \sum_i \left( \psi_{\text{detection score}} + \psi_{\text{truncation}} \right) + \sum_{ij} \psi_{\text{occlusion}} \]
3. Occlusion Reasoning

What we want:

\[
E(\hat{D}) = \sum_{i \in \hat{D}} \left( w_d(s_i - b) - w_o \frac{|m_i^o| + |p_i|}{|m_i|} \right) + w_o \frac{|m_i^t \not\in I|}{|m_i|} + \sum_{i,j \in \hat{D}, i \neq j} \left( w_o \frac{|m_{\text{far}}(i,j) \cap m_{\text{near}}(i,j)|}{|m_{\text{far}}(i,j)|} \right) - w_p \frac{\sum_{k=v,o,t} |m_i^k \cap m_j^k|}{\min(|m_i|, |m_j|)}
\]

Greedy approach to maximize E – better methods may be possible

Adam Allevato | 2016-06-04 | CS381V | UT Austin
3. Occlusion Reasoning
3. Occlusion Reasoning
4. 3D Localization

At this point we also **assign a single CAD model to each detection** based on the closest match from clustering.
Results
This approach can generalize.
Car Detection and Orientation on KITTI

<table>
<thead>
<tr>
<th>Method</th>
<th>Object Detection (AP)</th>
<th></th>
<th>Object Detection and Orientation estimation (AOS)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Easy</td>
<td>Moderate</td>
<td>Hard</td>
<td>Easy</td>
</tr>
<tr>
<td>ACF [1]</td>
<td>55.89</td>
<td>54.77</td>
<td>42.98</td>
<td>N/A</td>
</tr>
<tr>
<td>DPM [2]</td>
<td>71.19</td>
<td>62.16</td>
<td>48.43</td>
<td>67.27</td>
</tr>
<tr>
<td>DPM-VOC+VP [3]</td>
<td>74.95</td>
<td>64.71</td>
<td>48.76</td>
<td>72.28</td>
</tr>
<tr>
<td>OC-DPM [4]</td>
<td>74.94</td>
<td>65.95</td>
<td>53.86</td>
<td>73.50</td>
</tr>
<tr>
<td>SubCat [5]</td>
<td>81.94</td>
<td>66.32</td>
<td>51.10</td>
<td>80.92</td>
</tr>
<tr>
<td>AOG [6]</td>
<td>84.36</td>
<td>71.88</td>
<td>59.27</td>
<td>43.81</td>
</tr>
<tr>
<td>SubCat [7]</td>
<td>84.14</td>
<td>75.46</td>
<td>59.71</td>
<td>83.41</td>
</tr>
<tr>
<td>Regionlets [8]</td>
<td>84.75</td>
<td>76.45</td>
<td>59.70</td>
<td>N/A</td>
</tr>
<tr>
<td><strong>Ours NMS</strong></td>
<td><strong>84.81</strong></td>
<td><strong>73.02</strong></td>
<td><strong>63.22</strong></td>
<td><strong>84.31</strong></td>
</tr>
<tr>
<td><strong>Ours Occlusion</strong></td>
<td><strong>87.46</strong></td>
<td><strong>75.77</strong></td>
<td><strong>65.38</strong></td>
<td><strong>86.92</strong></td>
</tr>
</tbody>
</table>

References:


Slide Credit: Yu Xiang
Analysis

The Good
- State-of-the-art results
- Explicitly models occlusion
- Data driven approach
- Can perform segmentation

The Bad
- Complicated training pipeline
- Requires CAD models of objects
- Requires 3D data for training
- Little to no work on classification (sedans vs. vans)

Uses handcrafted features

How long does it take?
Discussion

• Will we ever not need 3D data for training?
• Does this work with deformable objects?
  • Does it need to?
• Possible extension: extend to more diverse classes
  • 227 different detectors just for “car”
  • What if we want to detect 20 different objects?
  • What kind of grouping would need to be done?
References

