Implicit Shape Model

A presentation by,
Anush Mohan

References:
Leibe et al., Combined object categorization and segmentation with an implicit shape model. ECCV’04 Workshop on Statistical Learning in Computer Vision.
Introduction

- Goal: Object categorization in real world scenes

- Problem definition:
  - Training: Given a collection of images of an object category, learn an implicit shape model using local appearance patches
  - Testing: Given an input image, segment objects based on category and label the objects
Motivation

- Integrates recognition and segmentation into a joint probabilistic framework.
- Flexible: makes efficient use of available training data
- Can handle multiple objects in a scene.
Generalized Hough transform

- What if want to detect arbitrary shapes defined by boundary points and a reference point?

At each boundary point, compute displacement vector: \( \mathbf{r} = \mathbf{a} - \mathbf{p}_i \).

For a given model shape: store these vectors in a table indexed by gradient orientation \( \theta \).

[Dana H. Ballard, Generalizing the Hough Transform to Detect Arbitrary Shapes, 1980]
Example

Circle model

Query

\[ P_1 \rightarrow \theta = 0 \rightarrow R = [rx, ry] = [1, 0] \rightarrow C_1 = P_1 + R \]
\[ P_2 \rightarrow \theta = 45 \rightarrow R = [rx, ry] = [0.7, 0.7] \rightarrow C_2 = P_2 + R \]
\[ P_k \rightarrow \theta = -180 \rightarrow R = [rx, ry] = [-1, 0] \rightarrow C_k = P_k + R \]

---

<table>
<thead>
<tr>
<th>(\theta)</th>
<th>(rx)</th>
<th>(ry)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>45</td>
<td>0.7</td>
<td>0.7</td>
</tr>
<tr>
<td>90</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>135</td>
<td>-0.7</td>
<td>0.7</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>270</td>
<td>0.7</td>
<td>-0.7</td>
</tr>
</tbody>
</table>
Implicit Shape Model: Approach

- Visual codebook is used to index votes for object position

training image
Implicit Shape Model: Approach

- Visual codebook is used to index votes for object position
Training

- Generate a codebook of local appearance patches
Training

• Generate a codebook of local appearance patches
  ▫ Extract 25x25 patches around Harris corners
Training

• Generate a codebook of local appearance patches
  ▫ Extract 25x25 patches around Harris corners
  ▫ Cluster similar patches together
Training

• Generate a codebook of local appearance patches
  ▫ Extract 25x25 patches around Harris corners
  ▫ Cluster similar patches together
  ▫ For each cluster, extract its center and store it in the codebook
Training

- Generate a codebook of local appearance patches
- Map the patch around each interest point to closest codebook entry
Training

• Generate a codebook of local appearance patches
• Map the patch around each interest point to closest codebook entry
• For each codebook entry, store all positions it was found, relative to object center
Testing

- Given test image, extract patches, match to codebook entry
Testing

• Given test image, extract patches, match to codebook entry
  ▪ each observation $e$ that is extracted at location $l$, it can be matched with a set of entries $\{ I \}$ in the codebook with probability $p(I_i | e, \ell)$
Testing

- Given test image, extract patches, match to codebook entry
- Cast votes for possible positions of object center
  - each codebook entry casts votes for the center of the object \( p(o_n, x | e, I_i, \ell) \).
Testing

- \( p(I_i | e, \ell) \)
  - each observation \( e \) that is extracted at location \( l \), it can be matched with a set of entries \( \{ I \} \) in the codebook with probability

- \( p(o_n, x | e, I_i, \ell) \).
  - each codebook entry casts votes for the center of the object

- Hence, the vote for object \( o_n \) and position \( x \) is:

\[
p(o_n, x | e, \ell) = \sum_i p(o_n, x | e, I_i, \ell) p(I_i | e, \ell).
\]
Testing

- Hence, the vote for object $o_n$ and position $x$ is:

$$p(o_n, x | e, \ell) = \sum_i p(o_n, x | e, I_i, \ell) p(I_i | e, \ell).$$

Independent of $e$, due to known interpretation $I$

$$= \sum_i p(x | o_n, I_i, \ell) p(o_n | I_i, \ell) p(I_i | e).$$

Independent of location $l$
Testing

• Hence, the vote for object $o_n$ and position $x$ is:

$$p(o_n, x | e, \ell) = \sum_i p(o_n, x | e, I_i, \ell) p(I_i | e, \ell).$$

  Independent of $e$, due to known interpretation $I$
  Independent of location $l$

$$= \sum_i p(x | o_n, I_i, \ell) p(o_n | I_i, \ell) p(I_i | e).$$

  Probabilistic Hough Vote
  Quality of match between $e$ and $I$

Confidence that codeword $I_i$ is indeed a foreground patch
Testing

• Hence, the vote for object $o_n$ and position $x$ is:

$$p(o_n, x | e, \ell) = \sum_i p(o_n, x | e, I_i, \ell) p(I_i | e, \ell).$$

Independent of $e$, due to known interpretation $I$

Independent of location $l$

$$= \sum_i p(x | o_n, I_i, \ell) p(o_n | I_i, \ell) p(I_i | e).$$

Implemented as a Uniform distribution $= 1/|I|$
Testing

• Hence, the vote for object \( o_n \) and position \( x \) is:

\[
p(o_n, x \mid e, \ell) = \sum_i p(o_n, x \mid e, I_i, \ell) p(I_i \mid e, \ell).
\]

Independent of \( e \), due to known interpretation \( I \)

\[
= \sum_i p(x \mid o_n, I_i, \ell) p(o_n \mid I_i, \ell) p(I_i \mid e).
\]

Independent of location \( l \)

\[
\text{score}(o_n, x) = \sum_k \sum_{x_j \in W(x)} p(o_n, x_j \mid e_k, \ell_k).
\]
Segmentation

• Given the object and center location assumption, determine if a pixel belongs to object or background:

\[
p(p = \text{figure} | o_n, x) = \sum_{p \in (e, \ell)} \sum_I p(p = \text{fig.} | o_n, x, e, I, \ell) p(e, I, \ell | o_n, x)
= \sum_{p \in (e, \ell)} \sum_I p(p = \text{fig.} | o_n, x, I, \ell) \frac{p(o_n, x | I, \ell)p(I | e)p(e, \ell)}{p(o_n, x)}
\]

• Define a likelihood ratio:

\[
L = \frac{p(p = \text{figure} | o_n, x)}{p(p = \text{ground} | o_n, x)}
\]
Experimental Results
Performance

**Fig. 4.** Comparison of our results on the UIUC car database with others reported in the literature.
Challenges

- If enough support is present from parts of multiple objects, multiple hypotheses are generated.
  - Pro: Handle occlusions.
  - Con: False hypotheses.
- Solution: A minimal description length (MDL) criterion, inspired by [Leonardis et al, 1995]
  - Define a pixel by its grayscale value and membership to a scene object (presence of object and error in representation)
  - Best encoding minimizes total description length of the image
Improvement using MDL
Thank you

Questions?