Combining Geometry and Local Appearance for Object Detection

Manuel Pascual García-Tubío
Automation and Control Institute
Vienna University of Technology
mp@acin.tuwien.ac.at

Horst Wildenauer
Institute for Computer Aided Automation
Vienna University of Technology
wilde@prip.tuwien.ac.at

Lech Szumilas
Industrial Research Institute for Automation and Measurements,
Al. Jerozolimskie 202, 02-486 Warsaw, Poland
lszumilas@piap.pl

Abstract

In this paper we address the problem of object detection in cluttered scenes. Local image features and their spatial configuration act as representation of object classes which are learned in a discriminative fashion. Recent contributions in the area of object detection indicate the importance of using geometrical properties for representing object classes. Prompted by this, we devised an approach tailored to control the importance of the features and their spatial alignment. We quantitatively show that modeling the spatial distribution of local features and optimising the influence of both cues significantly boosts object detection performance.

1 Introduction

Object detection can be defined as finding instances of one or more object classes in images, by analysing characteristic image features and their spatial arrangement. Typically, an object class is represented by a set of spatially distributed features [3, 5, 11, 13] (e.g. local image descriptors or edge maps associated with the object centroid) combined into a classifier that describes discriminative properties of each feature.

Recent approaches have shown remarkable results on a set of challenging databases. Their authors commonly emphasize the importance of encoding geometrical properties along with local features into an object model. For example, Opelt et al. [12] constrain the position of individual shape parts with relation to the object centroid. Shotton et al. [13] and Ferrari [6] also relate local shape based features to the object centroid and use this information to integrate the votes from individual parts. All these methods have in common that geometric information of individual parts plays an important role in the recognition process, though the use of particular features as well as the encoding and learning of geometric part-relations differs significantly.

Only recently, Stark and Schiele [14] attempted to analyze the gain in classification performance attained by the incorporation of information about geometric layout. The proposed localized Bag-of-Words approach enables for gradual addition of location information to object representations, thereby allowing for a quantification of the impact on classification accuracy. It was shown, that location information indeed boosts overall accuracy, most significantly when combined with less discriminative shape features.

Inspired by these findings, we propose a method that explicitly combines features with their geometrical relationships and allows to control the influence of both entities on the final result. In our case the similarity between the object model and the detected object instance is estimated from alignment as well as similarity of corresponding features. This solution produces a classifier that is sensitive to both and more importantly allows us to investigate the importance of features as well as geometry on the accuracy of object detection.

2 Method overview

The corpus of the method is divided in two main parts, namely the training of the object class model...
(Sec. 3) and the subsequent object detection procedure (Sec. 4). The process of the object model extraction is performed for each class separately and is divided into the following steps: 1) Image features along with their scale normalized positions are extracted from a set of training examples (Sec. 3.1). Examples of objects are obtained from rectangular regions in training images enclosed by the predefined bounding boxes. 2) During codebook generation (Sec. 3.2), features computed from training examples are quantized into clusters representing distinctive appearance and location inside the canonical bounding box. 3) A Gentle-Boost classifier is used to learn the object model, fed with the collection of positive and negative samples that result from matches between codebook and instances from the validation set (Sec. 3.3). The matching between features is tailored to optimize the balance between the feature description and the spatial information of those.

Once the object model is built, detection is performed with a sliding window approach. Classification votes are cast along the scale space and object hypothesis are aligned with the classification maxima. Multiple detections of the same object instance are eliminated by non-maxima suppression.

3 Object Representation

3.1 Feature extraction

We use SIFT [9] features extracted around symmetry based interest points [15]. In preliminary experiments we found that symmetry points are more densely distributed and repeatable than the often used DoG based interest points. We tend to utilize these well understood features and focus on improving object detection by applying geometrical properties.

3.2 Codebook

Before we compute the appearance codebook for local image patches, feature sets from positive training instances are spatially normalized to the canonical bounding box. Then, similar to [6], each bounding box is subdivided into a regular $11 \times 11$ grid of rectangular cells from which all possible $3 \times 3$ blocks of adjacent cells are formed, see Fig. 1.

For each of the resulting 121 blocks, features falling into it are clustered together using K-means. The complete codebook is obtained by concatenating all clusters along the grid. Due to the overlap of blocks, features can be shared between several clusters. For all our experiments $K$ was experimentally set to 10, resulting in a total number of 1210 codebook entries.

3.3 Classifier Training

To learn the visual object class representation, we employ the discriminative machine learning technique of boosting. To be more specific, we use Gentle-Boost [7] with weak classifiers in the form of regression stumps as proposed by [16].

The Gentle-Boost classifier has a typical form of linear combination of $M$ weak classifiers:

$$H(x, s) = \sum_{m} a_m \left( \epsilon_m > \theta_m \right) + b_m$$  \hspace{1cm} (1)

where $a_m$, $b_m$ and $\theta_m$ are the parameters of the best weak classifier, and $\epsilon_m$ is the similarity between the $m$-th codebook entry and the corresponding feature in the analyzed image or training example.

In our approach, this similarity is defined by a combination of similarity of the feature vectors and the spatial alignment of their location in the image window. More formally:

$$\epsilon_m \equiv \epsilon_{ia} = \exp \left( -\frac{d_s(i, a)^2}{2\sigma_s^2} \right) \exp \left( -\frac{d_f(i, a)^2}{2\sigma_f^2} \right)$$  \hspace{1cm} (2)

where $d_s(i, a)$ is the spatial distance between the $i$-th codebook entry and the $a$-th object instance feature (normalized to the canonical bounding box extents). $d_f(i, a)$ is the euclidean distance between the feature vectors. Parameters $\sigma_s, f$ allow to regulate the amount
of penalty that is associated with the feature misalignment and dissimilarity respectively:

- \( \sigma_f \rightarrow 0 \land \sigma_s \rightarrow 0 \) is a strict penalizer that only accepts perfectly aligned and identical features.
- \( \sigma_f \gg 1 \land \sigma_s \gg 1 \) produces almost a linear similarity function.
- \( \sigma_f \gg 1 \land \sigma_s \rightarrow 0 \) is forgiving for feature dissimilarity but requires a good spatial alignment which would agree with the notion of weak local features (both shape and appearance) and the importance of geometry in the object description.

The classifier training is performed on a set of positive (objects) and negative (background) examples extracted from the training data sets. Each example provides information about similarity and spatial alignment between codebook entries and the corresponding object features. Negative examples are randomly extracted from images not showing any instance of the specific class as well as from background of positive images. These additional negative examples are sampled from image regions with 30% overlap with the object bounding box (cf. [13]).

Due to the limited number of training images containing positive examples we artificially create additional examples by displacing, and stretching the bounding box window within small, predefined ranges to account for possible transformations not represented by the training data set [8]. Overall, we sampled positive and negative examples in a ratio of 1:10. Depending on the training data set the learner typically converged after selecting around 100 weak classifiers.

4 Object Detection

The presented method performs multi-scale object detection based on the sliding window approach [17, 4]. The classification scores \( H(x, s) \) are computed across the scale-space and stored in a discrete 4D map that covers image locations and detection window sizes representing anisotropic scale. The number of scales to be searched is estimated from the training data set and represents frequently occurring combinations of window size and ratio.

Next we give a detailed description of the detection process computing classification score \( H(x, s) \) for different locations and sizes of the detection window: 1) Establish codebook feature correspondences – each of the codebook entries is associated with the most similar feature inside the detection window using (2), that measures spatial alignment and feature similarity. Note that some of the image features inside the detection window may be left unassigned. 2) Compute classification score \( H(x, s) \) (1). Then, object hypotheses are associated with the local classification maxima (in the scale-space domain) that are higher than the predefined confidence threshold \( \Theta \). 3) Non-maxima suppression – the corresponding overlap between remaining maxima is computed. If bounding boxes associated with two proximal maxima exhibit more than 50% symmetric area overlap (as in [1]), the one with smaller confidence is removed.

5 Evaluation

Our approach was tested on five databases widely used to assess the performance of object detectors, namely the Weizmann Horses [2], the Caltech Faces, Aeroplanes and Motorbikes [5], and the TU-Graz Bicycles Database [10]. For each dataset, we select a relatively small number of images (< 25%) for codebook extraction and the remainder of the training set to serve for learning the boosted classifier. The full training set does not exceed 50% of the whole database in each case.

Figure 2 shows the performance of our algorithm utilizing precision-recall curves (PR). Table 1 gives corresponding area under PR curve (PR-AUC) values in comparison to Shotton et al. [13]. Shotton operates on boundary-based features extracted from edges, which were tuned for detection of objects best characterized by shape. Despite using local appearance features, our method is able to compete and in some cases outperform [13], especially on the challenging databases containing horses and bicycles. Note that Shotton separated bicycles into side, frontal, and back views; we follow this and report only the side view results here.

Figure 3 illustrates the dependency of PR-AUC for the bicycles database on parameters \( \sigma_s \) and \( \sigma_f \), which regulate the influence of spatial alignment and similarity between features (respectively) in the codebook and the detected object instance as discussed in Section 3.3. This graph clearly shows that once the influence of fea-

<table>
<thead>
<tr>
<th>Database</th>
<th>Training</th>
<th>Test</th>
<th>AUC</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horses</td>
<td>100</td>
<td>228</td>
<td>0.92</td>
<td>0.89</td>
</tr>
<tr>
<td>Faces</td>
<td>100</td>
<td>217</td>
<td>0.95</td>
<td>0.98</td>
</tr>
<tr>
<td>Aeroplanes</td>
<td>100</td>
<td>400</td>
<td>0.81</td>
<td>0.93</td>
</tr>
<tr>
<td>Motorbikes</td>
<td>100</td>
<td>400</td>
<td>0.97</td>
<td>1.0</td>
</tr>
<tr>
<td>Bikes (side)</td>
<td>72</td>
<td>72</td>
<td>0.72</td>
<td>0.69</td>
</tr>
</tbody>
</table>

Table 1. PR-AUC values for the performance of our method.
Figure 2. Precision-Recall curves for evaluated databases.

Feature similarity is made dominant over spatial alignment ($\sigma_s \gg \sigma_f$) the detection performance deteriorates. The same is evident for the opposite case ($\sigma_f \gg \sigma_s$) although results are better, suggesting that geometrical properties play a more important role than the raw features similarity.

Figure 3. PR-AUC as a function of $\sigma_f$ and $\sigma_s$ for the bicycles database. Star indicates a combination of $\sigma_f$ and $\sigma_s$ producing best score. Letters A and B represent extreme cases of $\sigma_s \gg \sigma_f$ and $\sigma_f \gg \sigma_s$.

6 Conclusions

We have presented an object detection approach that combines the spatial configuration of local image features with their inherent information. The method is tailored to carry out the evaluation that quantitatively shows the influence of geometry on the classification accuracy. We show that, despite the simplicity of the approach, excellent results are obtained by properly balancing geometry and feature similarity. Our method compares favourably to the boundary-based approach of [13] which is another indication that not the type of features but their spatial distribution plays a crucial role in the ability to detect complex objects.

Further improvements may be achieved through the use of multiple object models per object class that could better capture intra-class variability - an issue which we plan to investigate further in future research.

References