Unsupervised Learning of Characteristic Object Parts from Videos

Henrik Skibbe, Alexandra Teynor and Hans Burkhardt
Center for Biological Signalling Studies (bioss), Albert-Ludwigs-Universität Freiburg
Chair of Pattern Recognition and Image Processing, 79110 Freiburg, Germany
{skibbe | teynor | burkhardt}@informatik.uni-freiburg.de

Abstract

We propose a robust algorithm which learns an abstract object class model from videos in an unsupervised manner. The model consists of a small set of object parts, each of them highly characteristic for a particular object view. We obtain these parts by automatically determining those object regions that are similar in different object instances. Using such a model we are able to detect objects in arbitrary poses. We successfully validate our approach using the PASCAL Visual Object Challenge 2006 [2] image database which shows promising results. Compared to state of the art approaches we keep the performance while training is done without supervision.

1 Introduction

A lot of work has been done in order to build object categorisation and detection systems for two dimensional images. The major problem of analysing two dimensional representations of objects is the loss of information about the original three dimensional structure. The visual appearance of objects often completely differs, depending on the viewing angle. Solving this problem by explicitly modelling different object views is one possible solution [8, 10]. Similar to [8], the method proposed here automatically determines characteristic object parts from images showing objects in different orientations. The major advantage of our algorithm is the fact that training is done automatically by segmenting video data where the object to be learned is shown from differing views. In that way we comfortably obtain a large set of training samples. Our algorithm uses this data to determine a small set of characteristic object parts in an unsupervised manner, each typical and highly discriminative for a particular object view. Our experiments have shown that a small set of those parts is sufficient to cover the whole three dimensional appearance of an object class. We use histograms in order to describe their visual appearance and train one-class support vector machines for classification. In this way we are able to decide if an object is presented in a query image or not.
Our learning algorithm can be divided into two main steps: The generation of training data by segmenting video frames using an optical flow vector field on the one hand and the determination and learning of object parts that are typical and characteristic enough to represent a specific object view on the other hand. This paper is focused on the determination of characteristic object parts, but we first want to give a brief sketch of our video segmentation algorithm in section 2. In section 3 we introduce our learning algorithm and finally present our results in section 4.

2 Getting Training Data from Videos

In order to obtain an abstract object model covering the entire three dimensional appearance of an object class, training data including objects shown from differing viewing positions is necessary. In this work we use videos as appropriate data sources. Due to this fact we are able to easily generate a large set of training samples. We use a motion based segmentation algorithm which automatically and robustly segments moving objects from a static background. For this an estimation of the motion of the pixels within successive frames is obtained by computing an optical flow vector field [4]. Then the magnitude of those optical flow vectors is used to detect moving image regions. Finally a connected component labelling determines and segments the objects. An automatically segmented video sequence from our experiments is depicted in figure 1.

Figure 1: Training data is generated by automatically segmenting moving objects.

3 Learning of Characteristic Object Parts

We first want to give a formal description of what we call an object part. We consider a part as an object region, which appears similar in different object viewing angles. For example, a wheel or a door, but also the whole foreside of a car containing the number plate and the two headlights might be considered as one single object part. In our algorithm we use local features that lie within these very image regions in order to robustly describe the appearance of those parts. For the local description we use Hessian-Affine

interest points [7] in combination with SIFT gradient histogram features [6], generating the most promising results. We denote a set of interest points describing the visual appearance of an image I as

$$\mathfrak{F}_I = \{(\mathbf{x}_0, \mathbf{f}_0), \ldots, (\mathbf{x}_k, \mathbf{f}_k), \ldots, (\mathbf{x}_n, \mathbf{f}_n)\} \quad (1)$$

where each $\mathbf{x}_k$ is a interest point position and $\mathbf{f}_k$ the according feature representing the visual appearance. In this way we can describe an object's part p_i that is visible in a subimage S_i of I by using a subset $\mathfrak{F}_i \subseteq \mathfrak{F}_I$. We denote this as $p_i := \mathfrak{F}_i$, where

$$\mathfrak{F}_i := \{(\mathbf{x}, \mathbf{f}) \in \mathfrak{F}_I \wedge \mathbf{x} \text{ lies in } S_i\} \quad (2)$$

Examples of such parts are depicted in figure 3. We further say that two parts p_i and p_j represent the same part and therefore belong to the same equivalence class P, if and only if there exists a transformation $\mathbf{H}_{ij}$ that affinely maps interest points describing part p_i to interest points with similar appearance describing part p_j:

$$p_i \sim p_j \Leftrightarrow |\mathfrak{F}_i| = |\mathfrak{F}_j| \wedge \quad (3)$$
$$\exists \mathbf{H}_{ij} \forall (\mathbf{x}, \mathbf{f}) \in \mathfrak{F}_i \exists (\mathbf{x}', \mathbf{f}') \in \mathfrak{F}_j : \mathbf{x} = \mathbf{H}_{ij}^{-1}\mathbf{x}' \wedge \mathbf{f} = \mathbf{f}'$$

Whenever the affine transformation between parts p_i and p_j is known, the *foreshortening factor* k_{ij} can be used as a heuristic that determines the part slanted to a lesser extent [8]. The *foreshortening factor* is given by

$$k_{ij} = \left(\lambda_1^{ij}\lambda_2^{ij} - 1\right) \quad (4)$$

where λ_1^{ij} and λ_2^{ij} are the two singular values of the affine transformation $\mathbf{H}_{ij}$. λ_1^{ij} and λ_2^{ij} represent the scaling values in two orthogonal directions. If k_{ij} is greater than 0, p_i is treated as a less slanted image of a part than p_j. Otherwise p_j is treated as the less slanted representation.

3.1 Determining Object Parts

In the following, we introduce the algorithm we use in order to obtain candidates for characteristic object parts from segmented video sequences. The algorithm determines object parts that are shown in their most frontal view, thus we are able to easily compare parts from different object instances. In this way we are able to determine characteristic parts that are typical for certain views of the general object class, which is described later in section 3.2. Using these parts we finally detect an object in arbitrary positions.
The algorithm obtaining candidates for characteristic object parts in their most frontal view can be divided into three steps:

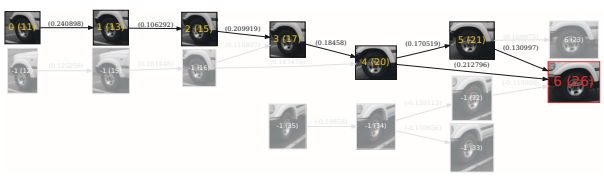

Figure 2: This example showing a wheel was generated automatically. The longest path leads to the part which is shown in its most frontal view. Edges are denoted with their corresponding foreshortening factors.

1. **Searching for corresponding object parts in video frames.** Considering two consecutive video frames, a moving object is typically shown in slightly differing positions. In order to locate object part candidates we search for image regions that can be matched in successive frames. Therefore we first compute a set of interest points and local features for each frame. Then we use the RANSAC algorithm which is suited to affinely match small sets of interest points that are visually similar between consecutive frames. A detailed description of the algorithm can be found in [8, 3, 9]. What we obtain is a set of corresponding image regions. Each correspondence represents the same object part, but in different views. The regions are described by sets of interest points $\mathfrak{F}_i$. Two sets of interest points $\mathfrak{F}_i$ and $\mathfrak{F}_j$ of corresponding image regions combined with their affine transformations $\mathbf{H}_{ij}$ form two equivalent parts $p_i \sim p_j$.
2. **Searching for further equivalent representations of previously obtained object parts.** We again use the RANSAC algorithm in order to recursively obtain further representations of previously obtained parts in nearby image frames. Moreover, we use the whole set of interest points that lie in the image region of parts we obtained previously in order to recursively match further corresponding regions of similar interest points in video frames. We obtain sets of equivalent parts, each part shown from a different view. Some representations of characteristic part candidates, each representing a different view of the lower part of a car side, are illustrated in top of figure 4. The parts are connected by known affine transformations.

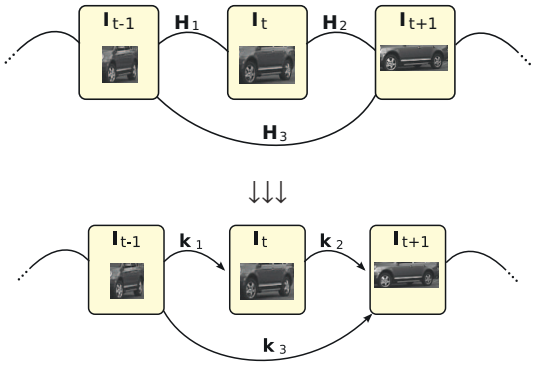

Figure 4: Using the forshortening factor leads to a directed graph with which we can determine the part which is shown in its most frontal view.

3. **Determining parts which are presented in their most frontal view.** In order to find parts that are typical for certain views of the general object class, we must find representations of parts that can be easily compared. This is achieved by determining the representation of parts shown in their most frontal view. Consider the different object parts that we obtained previously, each represented by a small set of equivalent parts. By using the *foreshortening factor* (4) we are able to determine representations of parts which are the least slanted.
Equivalent parts are internally related by known transformations that we obtained by successfully matching parts

Figure 3: Object parts in their most frontal view. We use histograms based on interest points in order to robustly describe the visual appearance of object parts.

in frames using the RANSAC algorithm. Hence we obtain a directed graph by evaluating the *foreshortening factor* for each known transformation. Moreover, the sign of the foreshortening factor determines the edge's direction. This scenario is illustrated in figure 2 and 4. Under the assumption that the heuristic always leads to the less slanted representation of a part, parts shown in their most frontal view can be determined by the nodes the longest path leads to [8]. Two examples of parts which are represented in their most frontal view are depicted in figure 3.

3.2 Characteristic Object Parts

Each part that was previously determined as *most frontal view* is a representation of a particular part from a single object. In the following we determine similarities between parts of different object instances in order to determine parts that are highly characteristic for a certain view of the desired object class. Our experiments have shown that these characteristic parts are sufficient to represent the whole three dimensional appearance of an object class. Our algorithm determines a small set of characteristic object parts, each being represented by a one-class support vector machine. These can be used to detect the presence of object class members in images. We obtain this set of support vector machines in three steps:

Figure 5: Two of our 1065 codebook words which we use to describe the visual appearance of object parts.

1. **Robustly describing the visual appearance of object parts.** In order to determine visual similarities between parts we have to describe them in a robust manner. We have chosen a bag-of-features representation [1]. Our codebook is based on SIFT features containing 1065 words (figure 5 shows two examples). We form a histogram for each part in its most frontal view based on the similarity of the corresponding local detections $\mathfrak{F}$ and the codebook words. We additionally include the orientation of interest points based on the local gradient main directions [6] which dramatically improves the results in our experiments. For this we use eight bins per word instead of one in order to embed the orientation information of local patches. Such a histogram is illustrated in figure 6.

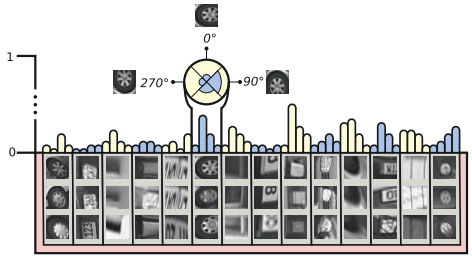

Figure 6: Object parts are represented by orientation histograms reflecting the probability of the occurrences and orientations of codebook words.

2. **Grouping similar parts.** We determine characteristic object parts by clustering the bag-of-words representations of the part candidates in their most frontal view. We use agglomerative clustering with average-linkage [5] in order to group parts that are similar with respect to their histograms. Hence parts which are similar between different object instances form a larger cluster than parts that are rare or untypical for a certain object class. Figure 7 shows three examples of a cluster representing a characteristic object part.

Figure 7: Example of a cluster representing a characteristic object part showing the number plate and headlights.

3. **Training of support vector machines.** We train one-class support vector machines using the histogram features of cluster representing characteristic object parts. More precisely, we train a support vector machine for each representative cluster containing more than five parts of different object instances. The necessary number of five parts was determined experimentally. Training is done by using a histogram intersection kernel. These support vector machines can finally be used to detect characteristic object parts in query images fast and robust, which is successfully demonstrated in our experiments.

4 Experiments and Results

We use the image database of the PASCAL Visual Object Challenge 2006 [2] to evaluate our algorithm and compare our results to the results of the competition 2 by challenging the object class "car": "Train in any (non-test) data, classify object present/absent". A brief sketch of the procedure of our experiment is given in the following:

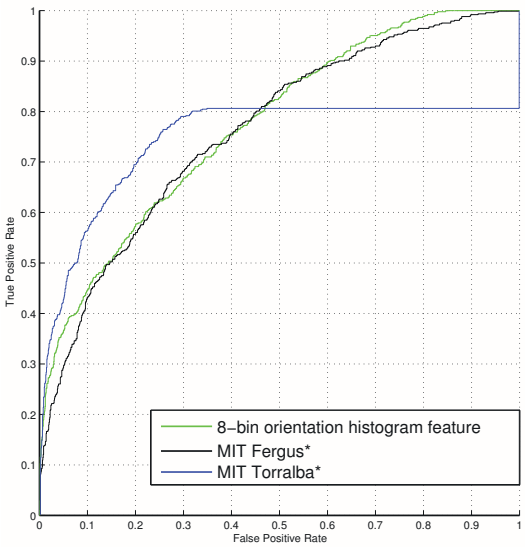

Figure 8: This ROC curves depicts the results from Rob Fergus and Antonio Torralba [2] as well as our own results.

We first use our video segmentation algorithm in order to obtain training data from videos containing moving cars. The videos are taken from a curve of the road in order to get sequences of images showing cars from different positions. Then we determine the characteristic parts of the object class car using the algorithm described in section 3. As a result we obtain a set of four one class support vector machines, each representing one characteristic part. The classification task is finally performed by successively scanning a query image using rectangular sliding windows of different sizes. A bag-of-features representation based on local features lying in the current search window area is computed at every sliding window position. Characteristic parts are detected by classifying these histograms using the support vector machines. If at least one characteristic part is found, a car has been detected. A ROC graph representing our results as well as results of Antonio Torralba and Rob Fergus [2] is depicted in figure 8. Compared to the results of Antonio Torralba and Rob Fergus we obtain similar performance but learning is done in an unsupervised way. Some results which are sorted with respect to their confidence values are shown in figure 9.

5 Conclusion

We proposed a novel object class learning algorithm and performed experiments leading to promising results. The main advantage of our algorithm is the fact that it learns an abstract object class representation in an unsupervised manner. Our algorithm obtains an abstract model description of a particular object class, consisting only of highly discriminating characteristic object parts. Our experiments have proven this statement by showing that a small set of characteristic parts is sufficient to robustly describe the whole object class. Hence we are able to detect objects in arbitrary poses. Furthermore we have shown that our results are comparable to other approaches using the results of the PASCAL "Object Class Challenge 2006".

Acknowledgement

This study was supported by the Excellence Initiative of the German Federal and State Governments (EXC 294)

Figure 9: The highest rated results of an example query are depicted in the figure above. Images are sorted with respect to their evidence values.

References

[1] C. Dance, J. Willamowski, L. Fan, C. Bray, and G. Csurka. Visual categorization with bags of keypoints. In *ECCV International Workshop on Statistical Learning in Computer Vision*, 2004.

[2] M. Everingham, A. Zisserman, C. K. I. Williams, and L. Van Gool. The PASCAL Visual Object Classes Challenge 2006 (VOC2006) Results. www.pascal-network.org/challenges/VOC/voc2006/results.pdf.

[3] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[4] B. K. Horn and B. G. Schunck. Determining optical flow. Technical report, Cambridge, MA, USA, 1980.

[5] A. Jain and R. Dubes. *Algorithms for Clustering Data*. Prentice-Hall, Inc. Upper Saddle River, NJ, USA, 1988.

[6] D. Lowe. Distinctive image features from scale-invariant keypoints. In *International Journal of Computer Vision*, volume 20, pages 91–110, 2003.

[7] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool. A comparison of affine region detectors. *Int. J. Comput. Vision*, 65(1-2):43–72, 2005.

[8] S. Savarese and L. Fei-Fei. 3D generic object categorization, localization and pose estimation. pages 1–8, 2007.

[9] H. Skibbe. Detection of rigid object class instances in videos using the geometrical configuration of local image features. Master's thesis, Chair of Pattern Recognition and Image Processing, University of Freiburg, Germany, 2008.

[10] A. Thomas, V. Ferrari, B. Leibe, T. Tuytelaars, B. Schiele, and L. V. Gool. Towards multi-view object class detection. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1589–1596, Washington, DC, USA, 2006. IEEE Computer Society.