



Accurate Object Localization with Shape Masks

Marcin Marszałek

Cordelia Schmid

INRIA, LEAR - LJK

665 av de l'Europe, 38330 Montbonnot, France

Marcin.Marszalek@inrialpes.fr Cordelia.Schmid@inrialpes.fr

Abstract

This paper proposes an approach for object class localization which goes beyond bounding boxes, as it also determines the outline of the object. Unlike most current localization methods, our approach does not require any hypothesis parameter space to be defined. Instead, it directly generates, evaluates and clusters shape masks. Thus, the presented framework produces more informative results for object class localization. For example, it easily learns and detects possible object viewpoints and articulations, which are often well characterized by the object outline. We evaluate the proposed approach on the challenging natural-scene Graz-02 object classes dataset. The results demonstrate the extended localization capabilities of our method.

1. Introduction

Object category localization is one of the most complex tasks in computer vision. Solving the localization problem requires not only detecting an object¹, but also determining the precise location of the object in an image.

The criteria of measuring localization accuracy have evolved over time. Agarwal and Roth [1] evaluated the center point of an object and classified localization as correct when the marked point was in the close neighborhood of the real center of the object. During the PASCAL Visual Object Classes challenge [3] the participants had to return bounding boxes for the objects. Even though localization is today commonly measured with a bounding box, we believe that modern localization methods should go even further, e.g., return some additional information about object pose (viewpoint, articulation), aspect (sub-type) or even state and properties. This can, to some extent, be achieved by returning the object outline. Given the outline of the object, one may for example determine the direction in which a bike is heading, distinguish between a sedan and a mini-van or decide whether a person is fat or thin. And indeed, a few methods which perform interleaved object detection and segmentation have been developed recently—see for

example the work of Leibe and Schiele [7] and of Opelt and Pinz [5]. Those methods, however, attack the segmentation problem after the decision about the object location is already made. And since the authors solve the localization problem by voting in the generalized Hough space, they are limited to parametrized hypotheses. Thus, the richness of information present in the object segmentation can not be fully exploited in the context of the localization problem. In contrast, we utilize object *shape masks* at each stage of our localization framework.

We would like to underline a significant difference in our object class localization approach compared to the recent object class segmentation approaches, like the ones of Winn and Jojic [25], Todorovic and Ahuja [23] or Russell et al. [18]. We perform object localization instead of scene segmentation. Our goal is to localize separate object instances within a test image, handling occlusions and strong background clutter. Our method does not use any segmentation or edge information of a test image and therefore we expect to get only approximate shapes for the localized objects—shapes that reveal additional object properties and do not segment out the visible object parts. Pixel-level accurate segmentation, however, should be easier after the object localization problem that we address in this paper is solved.

It was shown that local image features can generate good object location hypotheses² even in heavily cluttered and occluded scenes [10]. However, as mentioned earlier, the evidence is usually collected in the generalized Hough space, which assumes that the description of object location is parametrized—the Hough space cannot deal directly with arbitrary shapes due to their high dimensionality. This generates a few problems with the otherwise very successful Implicit Shape Model of Leibe and Schiele [7]. Firstly, the low dimensionality of the hypotheses causes the final answers to reveal problems with global consistency. This was addressed by Leibe et al. [8], but only in form of a post-processing step appended to the original ISM. We approach this problem directly by using the high-dimensional shape masks as hypotheses. Such hypotheses can be considered

¹Sometimes the word *detection* is used in the literature as a synonym for *localization*. We consider detection to be an image classification task, where the presence or absence of an object in an image is determined.

²We use the term *hypothesis* for an initial estimation of object location, which can be then evaluated and processed. The final localization *decision* can result from a set of hypotheses.



(a) bikes

(b) cars

(c) people

Figure 1. Sample Graz-02 images. Note the high intra-class variations, significant amount of background clutter and difficult occlusions.

similar only if the object outlines are globally similar. Secondly, the low dimensionality of the hypotheses makes it difficult to deal with multiple object viewpoints and articulations. This was addressed by Seemann et al. [20], but as aspect parametrization is difficult in a general case, the proposed solution was limited to aspect clustering and treating each aspect separately. This, however, prohibits aspect combination during recognition. A possible solution to a similar problem was at the same time proposed by Thomas et al. [22], but finding the multiview tracks that link single-view detectors requires a special training procedure with over 10 viewpoints of each training object. We implicitly deal with multiple viewpoints and articulations. Object aspects are detected during training and the similar ones can be combined during recognition.

Shape masks cast by local features have recently been used by Marszalek and Schmid [12] to improve image classification results. However, the approximate segmentations computed by their spatial weighting procedure cannot be used for localization, as the method does not allow to distinguish between separate object instances. Therefore, in this paper we propose an online shape masks clustering algorithm that allows to collect evidence about possible object locations and outlines, resulting in precise localizations. Moreover, we show that the same clustering principle can be used to cluster the outlines of the training objects. This allows to detect possible object aspects.

Fritz et al. [4] have recently shown that combining the power of generative modeling with a discriminative classifier allows to obtain good results for object category localization. They extend the Implicit Shape Model mentioned earlier by appending a Support Vector Machine classifier to its output. In our framework, however, we propose to evaluate the hypotheses (shape masks cast using local features) before the evidence collection step. This allows to easily deal with false hypotheses caused by local ambiguities and makes the search for maxima in the hypothesis space easier.

In this paper we propose a localization framework based on object shape masks. We show that it is beneficial to avoid reducing the localization hypotheses to parametrized shapes like bounding boxes. By employing shape masks as hypotheses one can enrich the localization answers, implicitly handle the global consistency issues and address multiple object aspects. Moreover, we propose to evaluate the localization hypotheses cast in a generative manner using a discriminative classifier. This allows to clean up the

hypothesis-space before the search for maxima.

The localization task is especially challenging in the presence of pose changes, intra-class variation, occlusion and background clutter. As more and more methods reach high precision and recall on relatively uniform datasets, for example images from the cow video sequences [16], it is important to consider difficult natural-scene conditions, where objects in various poses are surrounded by a complex environment. We choose the Graz-02 [14] dataset to evaluate our framework, as it contains natural real-world images with significant amount of intra-class variations, occlusions and background clutter (cf. fig. 1).

The paper is organized as follows. In section 2 we introduce some background material, which allows us to describe the training and recognition procedures of our framework in section 3. In subsection 3.1 we describe the training procedure focusing on the multi-aspect feature of our method, and in subsection 3.2 we explain the recognition procedure including the details of our online shape masks clustering method. In section 4 we evaluate our approach on Graz-02 and compare to the state-of-the-art. We conclude the paper in section 5.

2. Background material

We first introduce in subsection 2.1 the sparse local features and explain how to use the feature parameters to align the shape masks. We then define a *shape mask* of an object and a similarity measure for these masks in subsection 2.2. Finally, we describe in subsection 2.3, how to construct a classifier that evaluates the confidence of a shape mask to lie on an object.

2.1. Sparse local features and alignment

Given an image, we use the Harris-Laplace [13] or the Laplacian interest point detector [9] to find a sparse set of salient image features. In our experiments the Harris-Laplace detector usually produces comparable results with a lower number of detections, thus it is our preferred choice for efficiency reasons. However, for small images, like the ones in Shotton's horses dataset, we choose the Laplacian detector that detects enough interest points for our method to work. Both detectors are invariant to scale transformations, they output circular regions at a characteristic scale. It is also possible to achieve rotation or affine invariance [6, 11]. Note, however, that it is unreasonable to use

more invariance than required for a given data set [26]. For most natural object data sets the vertical direction is well defined and, therefore, the orientation of the features contains valuable information. Thus, even though our framework supports affinely adapted features, in our experiments we use only the scale-invariant version of the detectors.

To compute appearance-based descriptors on the patches obtained by the detectors, we employ the SIFT [11] descriptor. It computes a gradient orientation histogram within the normalized support region determined by the detector and produces a 128-dimensional feature vector for each region.

Rectification parameters complement the invariant local description of an interest point. For example, if a local region description is invariant to scale transformations, the rectification parameters have to include the scale to compensate for this invariance. Precisely, if a description d of an local image region i is invariant to a transformation $T(i, \rho)$ with parameters ρ , then for each local image region j the parameters of this transformation $\rho_j \in D_\rho(T)$ are included in its rectification $\theta_j \in \Theta$. Precisely,

$$\Theta = \Pi_{T \in \mathcal{T}} D_\rho(T), \quad \mathcal{T} = \{T : \forall \rho, i \ d(i) = d(T(i, \rho))\} \quad (1)$$

where Π is a cartesian product and $D_\rho(T)$ is a domain of transformation parameters.

In our framework the descriptor is made invariant to a chosen set of affine transformations by normalizing the local image region before computing the description. Therefore, the rectification matrix θ_i transforming the image coordinates to the normalized patch coordinates [17] can be used to encode the rectification parameters of a feature i . Given a match between features i and j , we can project the mask associated with feature i to the reference frame of feature j (we call this *mask alignment*) by composing the shape mask with the transformation matrix P_{ij} computed as

$$P_{ij} = \theta_i^{-1} \theta_j \quad (2)$$

2.2. Shape masks

The *shape mask* $S : \mathbf{R}^2 \rightarrow \mathbf{R}$ is a natural generalization of the discrete binary segmentation mask $S_b : \mathbf{Z}^2 \rightarrow \{0, 1\}$. To measure the *shape mask similarity* we adapt a commonly used overlap area measure defined as the ratio of overlap area to the union area. For binary masks Q_b and R_b the overlap area measure can be written as

$$o_b(Q_b, R_b) = \frac{|Q_b^1 \cap R_b^1|}{|Q_b^1 \cup R_b^1|} = \frac{\sum \min(Q_b, R_b)}{\sum \max(Q_b, R_b)} \quad (3)$$

where Q_b^1 resp. R_b^1 denotes the level set of mask Q_b resp. R_b at 1, $\min(Q_b, Q_r)(x, y) = \min(Q_b(x, y), Q_r(x, y))$ and the sum is taken over the whole domain. Thus, we define the overlap based similarity measure o_s for shape masks Q and R as

$$o_s(Q, R) = \frac{\int \min(Q, R)}{\int \max(Q, R)} \quad (4)$$

Note, that this similarity measure will return 1 for identical shape masks and 0 for non-overlapping ones.

A straightforward implementation of the similarity measure given in eq. (4) leads to very inefficient code. Note, however, that it can be rewritten as

$$o_s(Q, R) = \frac{C}{\int Q + \int R - C}, \quad C = \int \min(Q, R) \quad (5)$$

Sums of all mask pixels can be cached and C needs to be computed only on the intersection of the supports of the shape masks. This makes the computation very efficient.

Finally, we need to compute a similarity measure o_f between two shape masks ζ_i and ζ_j associated with features i and j after aligning them. We define it as

$$o_f(i, j) = o_s(\zeta_i \circ P_{ij}, \zeta_j) = o_s(\zeta_i, \zeta_j \circ P_{ji}) \quad (6)$$

where o_s is defined by eq. (4) and P_{ij} (P_{ji}) by eq. (2). We call such similarity measure between two features a *featured shape mask similarity*.

2.3. SVM with χ^2 kernel

To evaluate the shape masks we use a bag-of-keypoints representation and a non-linear Support Vector Machine (SVM) with χ^2 kernel [26].

Given a visual vocabulary [24], we can represent the appearance of the image part covered with the shape mask as a histogram of vocabulary words occurrences. Each histogram entry $h_{ij} \in H_i$ is the proportion of all image features covered by the shape mask i and assigned to a vocabulary word j to the total number of features covered by the shape mask. Such a histogram can be computed for any shape mask and is then passed to the SVM classifier [19].

The classifier is trained to distinguish between objects and background. We use the training shape masks covering the objects as the positive set and the image areas with no objects as the negative one. After the classifier is trained, any shape mask can be evaluated.

We use an extended Gaussian kernel [2]:

$$K(H_i, H_j) = e^{-\frac{1}{A} D(H_i, H_j)} \quad (7)$$

where $H_i = \{h_{in}\}$ and $H_j = \{h_{jn}\}$ are the histograms and $D(H_i, H_j)$ is the χ^2 distance defined as

$$D(H_i, H_j) = \frac{1}{2} \sum_{n=1}^V \frac{(h_{in} - h_{jn})^2}{h_{in} + h_{jn}} \quad (8)$$

where V is the vocabulary size. The parameter A is the mean value of the distances between all training samples.

3. Shape Masks framework

3.1. Training

The overview of the training procedure is given in fig. 2, the details of each step follow. Two main blocks of the ag-

glomerative aspect clustering, i.e., computation of the aspect similarities and merge of the two most similar aspects, are performed iteratively until no more merges are possible.

Compute sparse local features. First, sparse local features are computed over all training images as described in subsection 2.1. The descriptors are clustered using k-means with $k = 1000$; cluster centers form a visual vocabulary. Given object segmentations, we discard the features which do not lie on any object. For the remaining features we keep the pointer to the shape mask created from the relevant object segmentation. For each feature we also keep its rectification parameters.

Compute feature similarities. We assume that two aspects are similar if they result in globally similar object shape and are also supported by similar local features appearing on the object at approximately the same locations. Thus, for each feature cluster (visual vocabulary word) we consider all feature pairs and compute the featured shape mask similarity as defined by eq. (6). Thresholding the similarity measure at $T = 0.85$ allows us to find visually matching (belonging to one feature cluster) feature pairs that would cast similar (as defined by eq. (4)) shape masks.

Vote for shape mask pairs. Each pair of matching features determined in the previous step casts a vote for the aspect pair they support (point to). The pair of shape masks with the highest number of votes is considered for the merge. This assures that the aspects result in similar object outlines (above the threshold T) and the aspects with many matched features (similar appearance) are merged first. If there are no more merge candidates left, the iterative part ends and singletons are pruned in the next step.

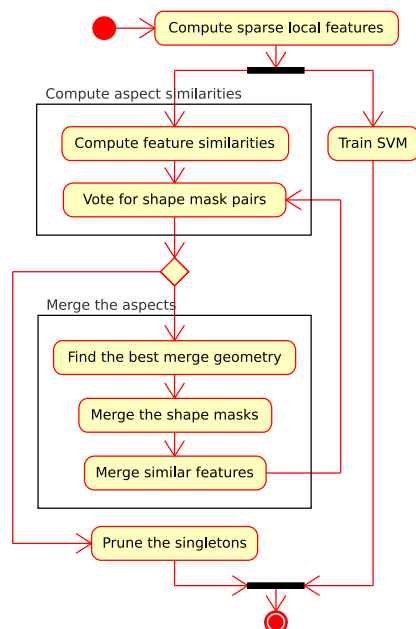


Figure 2. Overview of the training procedure. The main operation blocks are executed iteratively.

Find the best merge geometry. To merge two shape masks we first determine a geometrical transformation between them. We choose the transformation defined by the feature pair with the highest featured shape mask similarity. This assures good overlap of both shape masks (high similarity of the aligned shape masks) and features (they get aligned according to the well matched feature pair).

Merge the shape masks. After the transformation is determined, the common reference frame is established by the feature with a higher scale parameter (in practical implementation this assures the best mask resolution). The other shape mask is transformed according to eq. (2). The same transformation is applied to features associated with the transformed shape mask. The weighted average of the registered shape masks is computed and features pointing to the masks being merged are associated with the new mask. The featured shape mask similarities affected by the merge are recomputed before continuing.

Merge similar features. To reduce the number of considered features we merge the features that are redundant. After the shape mask merging step we expect to encounter many similar features appearing at approximately the same location. Thus, it is desirable to compute the weighted average of features that are visually similar (belong to one feature cluster), point to the same shape mask and would cast similar shape masks, i.e., their featured shape similarity is above the threshold T . The last condition assures, that also the rectification parameters of the merged features are similar, i.e., that we do not merge, e.g., front and back wheels of a car. The featured shape mask similarities for merged features have to be recomputed before launching a new clustering iteration.

Prune the singletons. As our experiments show, it can be beneficial to prune the single training shape masks that are not merged with any other shape mask during the agglomerative shape mask clustering procedure.

Train the SVM. A Support Vector Machine is trained to evaluate the hypotheses as described in subsection 2.3. We train a binary SVM classifier for each object category. In theory, a separate SVM could be trained for each object aspect, which could be beneficial. Our experiments, however, have shown that, probably due to a resulting small number of training examples per aspect, this can be inferior in situations where a limited amount of training data is available.

3.2. Recognition

The overview of the recognition procedure is given in fig. 4, the details of each step are given below.

Compute sparse local features. Given a test image, a set of sparse local features is computed as described in subsection 2.1. The feature space is then quantized using the vocabulary created during training, i.e., each feature is assigned to the nearest vocabulary word.



Figure 3. Main points of our framework. (a) Ambiguities introduced by local features may generate false hypotheses (left). Hypothesis evaluation helps to avoid them in our framework (right). (b) Occlusion weakens the discriminative classifier response and the object may be missed (left). This is reduced in our framework by collecting the local evidence provided by agreeing features (right).

Cast hypotheses. The hypotheses are generated by investigating all test image features in arbitrary order. For each test feature we consider all similar training features, i.e., the training features assigned to the same vocabulary word. Each training feature points to a shape mask. The rectification parameters of a training feature and a test feature determine the alignment, as shown in eq. (2). Thus, we can project the training shape masks into the test image and therefore cast the initial hypotheses about possible object location.

Evaluate hypotheses. The hypotheses are evaluated with a SVM classifier as described in subsection 2.3. Only the hypotheses for which a positive confidence measure is returned are kept. The confidence measure is stored with each hypothesis. Performing the evaluation step immediately after the hypothesis is cast allows to easily deal with the ambiguities intrinsic to local features before they could influence the hypothesis-space clustering, see left part of fig. 3. It also cleans up the hypothesis-space from wrong hypotheses caused by background clutter.

Cluster hypotheses. After the hypotheses are evaluated, we could look for the strongest ones and consider them as localization decisions. We use, however, a discriminative classifier, which is relatively sensitive to occlusions. Thus, combining its output with the generative evidence provided by local features should be beneficial, see right part of fig. 3. To collect the evidence from multiple hypotheses we perform online agglomerative hypothesis clustering.

It is computationally prohibitive to store all generated shape masks in memory. To overcome this problem, we use an online approach. We pipe the hypotheses through the

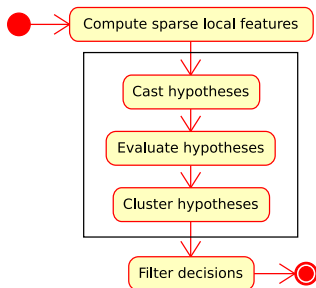


Figure 4. Overview of the recognition procedure. The main operation block is executed in a pipe to reduce memory requirements.

system as they are generated and we keep a limited number of them in memory at the same time. The number of hypotheses is reduced during the clustering step by merging similar shape masks and dropping non-promising ones. At this point of the algorithm each hypothesis consists of two elements—a shape mask and an associated confidence value, computed by the SVM in the previous step. We can measure the similarity between two shape masks as defined by eq. (4). When the number of collected shape masks exceeds the limit of $L = 100$ elements, the pair of hypotheses with the most similar masks is considered for merge. If the similarity is above the merge threshold $U = 0.7$ the hypotheses are merged. Otherwise, the hypothesis with the lowest confidence value is found and dropped.

When two hypotheses are merged, a combined shape mask needs to be computed. At each point, the resulting shape mask is the average of the masks being merged, weighted by the confidence values associated with each of the two mask. The confidence of the resulting hypothesis is the sum of the confidence values of the combined hypotheses. Thus, the merge of shape masks Q and R associated with confidence values η_Q and η_R can be expressed as

$$S = \frac{\eta_Q}{\eta_Q + \eta_R} \cdot Q + \frac{\eta_R}{\eta_Q + \eta_R} \cdot R \quad \eta_S = \eta_Q + \eta_R \quad (9)$$

where S is the resulting shape mask and η_S its confidence.

After all hypotheses are generated, evaluated and collected, the agglomerative clustering continues until no more hypotheses can be merged, i.e., all the remaining hypothesis pairs have the shape mask similarity below the threshold. The remaining hypotheses are then passed to the next step.

Filter decisions. Finally, the decisions are filtered to reduce the number of false positives. We have implemented a simple approach for situations where no significant self-occlusion of objects is expected. We reduce significantly overlapping decisions to the one with the highest confidence value. This allows us to avoid false positives resulting from subsequent detections of an already detected object.

4. Experimental results

In subsections 4.1 and 4.2 we evaluate our approach on the Graz-02 dataset. A comparison with the state-of-the art on the Weizmann horse dataset [21] is then presented in 4.3.

object class	cars	people	bicycles
no hypothesis evaluation	40.4%	28.4%	46.6%
no evidence collection	50.3%	40.3%	48.9%
our full framework	53.8%	44.1%	61.8%

Table 1. Pixel-based RPC EER measuring the impact of hypothesis evaluation and evidence collection.

4.1. Evaluation of the recognition components

In this section we evaluate our recognition components on the Graz-02 dataset using the original ground-truth annotation. These annotations do not give the outline for individual objects, but only the segmentation mask for each image, i.e., it is impossible to know how many objects are present. Clustering shape masks requires object specific annotations and will therefore not be used in this section. Here, we approximate the object shape mask with the segmentation mask of the entire image.

We run our framework on all three object classes: *bikes*, *cars* and *people* (cf. fig. 1). For each class we use the first 150 odd-numbered images for training and the first 150 even-numbered images for testing, i.e., we follow the experimental setup defined by Opelt and Pinz [15]. To evaluate the results, we use pixel-based recall precision curves (RPCs). Based on the ground-truth segmentation maps we count a pixel belonging to an object as a true positive when it is detected and as a false negative otherwise. The pixels incorrectly detected as object pixels are false positives.

Table 1 shows the equal error rates³ of the recall precision curves for each of the classes. We compare our full recognition system with image-based shape masks to two modified versions. “No hypothesis evaluation” does not use the hypothesis evaluation step and assumes the same confidence for each hypothesis cast by the local features. “No evidence collection” does not collect the evidence provided by the features, but selects the hypotheses with the highest classifier response instead. For each class the performance of our combined framework is significantly better than the performance of the approaches where the hypothesis evaluation or evidence merging are missing. This confirms that both elements are necessary in order to perform precise object class localization and proves that our framework is able to combine them. Note that evidence collection is crucial for bicycles, as the discriminative classifier may get easily distracted by the background surrounding thin bicycle parts.

4.2. Aspect clustering

Clustering shape masks requires additional annotations of the Graz-02 dataset as stated above. We have therefore extended the annotations by separating the available per-image annotations into per-object segmentations. More-

³Precisely, the point where the recall is equal to the precision is called *break even point*. For consistency with the literature we denote it as EER.

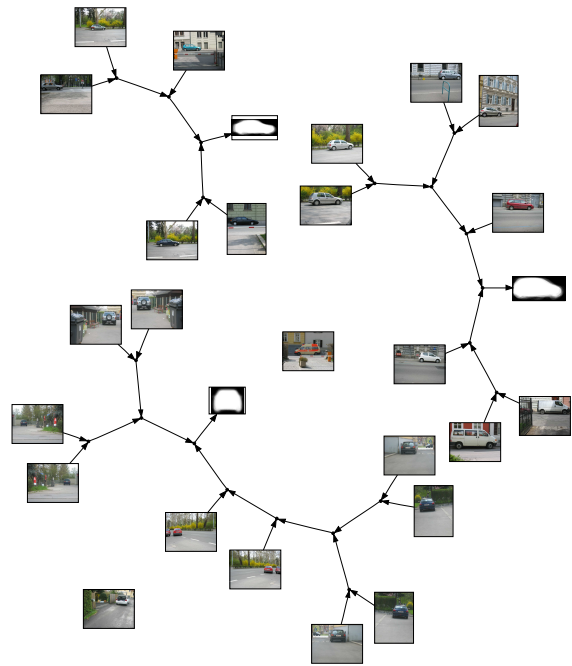


Figure 5. Several car aspects detected by agglomerative clustering.

over, each object has been marked as *truncated* by an image border or *difficult* to recognize if appropriate. The images were divided into equally large training and test sets. For training, images containing at least one non-truncated object were randomly drawn. The remaining images on which all the appearing objects could be marked by the annotators define the test set.⁴ The improved annotations and the training and test image numbers are available online at <http://lear.inrialpes.fr/data>

Figure 5 shows fragments of the agglomerative aspect clustering trees computed during training for the cars. We have chosen 3 aspect clusters from the 6 largest ones (grouping 17+ objects). Next to the shape mask resulting from the clustering we present the earliest merged (thus most similar) training objects that initialized each of the clusters. We can see that the detected aspects reveal more than just the viewpoint at which the object is observed. When the object outline is significant, different car types (like sedans and minivans) are clustered together and form separate aspects. Also 2 sample singletons (from the total number of 72) are shown. Zooming in, we can see that the singletons are true outliers.

The influence of aspect clustering on recognition accuracy for cars is presented in fig. 6. We use the shape mask overlap similarity (cf. eq. (4)) with threshold 0.3 as the criterion for correct localization and display recall as a function of false positives per image. We can observe a slight im-

⁴Due to image content requirements mentioned above some images were not used at all, but at the same time we have annotated and used some images that were not used in the original setup. There are 354 car images, 280 people images and 324 bike images in total.

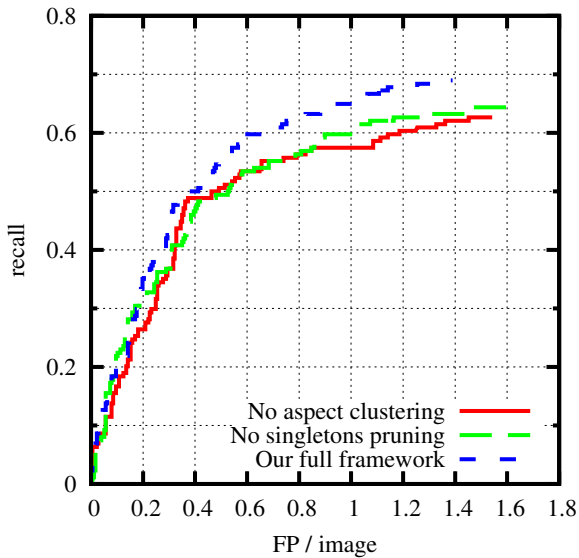


Figure 6. Recognition rate for cars given as recall in a function of FP per image. We can observe the impact of aspect clustering.

provement in the accuracy due to aspect clustering itself and further improvement due to singletons pruning performed after the clustering. Aspect clustering does not only improve the recognition speed and memory requirements (as there are less features and shape masks to be kept and considered during recognition), but also the localization accuracy. Furthermore, it opens the possibility of annotating the aspects with pose or sub-types information.

For people and bicycles, due to a large number of possible variations in articulations and poses, the training images are not sufficient to determine good clusters, which either have low support or become blurred (if we lower the threshold T for merging shapes). Still, with singleton pruning turned off, we can perform successful localization. For people, the recall is 43% for 5 FPs/image. The number of false positives may appear high. Note, however, that people are often small, close to each other and occluded. It may also be difficult to match the correct articulation and a shape mismatch can result in a false positive. For bicycles, with a localization accuracy criterion of 0.2, the recall is 59% for 2 FPs/image. We had to lower the criterion due to the transparent structure of a bike that lowers the overlap measure even for a small misalignment of the mask.

Figure 7 shows sample detections on the Graz-02 dataset. It demonstrates that our method is able to successfully localize object instances, even under difficult conditions of background clutter and occlusion. We can observe that the computed masks give information about the pose of

Shotton [21]	92.1%
Our framework ($T = 0.85$, with singletons)	94.6%
Our framework ($T = 0.7$, no singletons)	94.6%

Table 2. RPC EER for Weizmann horse dataset.

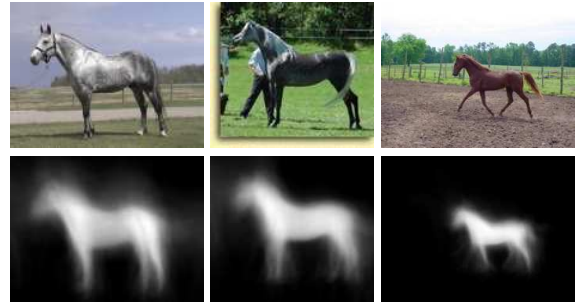


Figure 8. Results on Weizmann horses dataset. Note that the shape masks are very accurate: the horse articulations are visible.

the object. Note that in the case of the images with more than one object, the subsequent objects are localized with subsequent hypotheses (third row, first four images). Note that third hypothesis has a very low score (4.9) and can therefore be discarded. It is interesting to observe that it is probably due to the small car part in the bottom left corner.

4.3. Comparison to the state-of-the-art

To compare our method to the state-of-the-art we evaluate our framework on the Weizmann horse dataset. We closely follow the setup of Shotton et al. [21]—we use the first 50 images of horses and the corresponding object segmentation plus the first 50 background images for training. The next 277 images from each set are used for testing. We also use the same criterion for localization accuracy—we determine the centroid of the computed segmentation mask and compute the distance to the centroid in the ground-truth. If the distance is less than 25 pixels, the localization is considered to be correct. Note that we follow their protocol strictly by using their scale-normalized images, and running our system at a single scale.

Table 2 compares our results to Shotton et al. [21]. We can observe that our approach improves the performance. Our results are reported for a training procedure without singleton pruning and the standard shape merging threshold $T = 0.85$ as well as for a lower shape merging threshold with singleton pruning. If we do not lower the threshold, too few aspect clusters are formed due to the large number of horses articulations, i.e., most of the aspects are singletons. Figure 8 shows a few example results. We can observe that the shapes of the horses are detected very accurately in the test images. We can even judge from the detected shape masks if the horse is standing or running.

5. Summary

In this paper we have proposed an object localization framework that uses shape masks as localization hypotheses. We have demonstrated that this enriches the localization decisions by revealing additional information about object viewpoint, articulation, sub-type or state. At the same

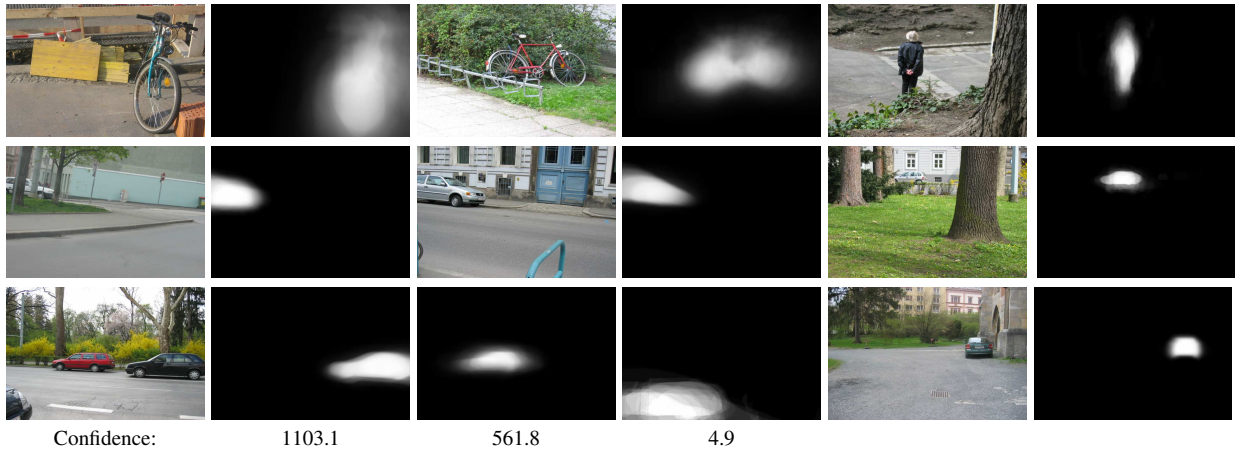


Figure 7. Results on Graz-02 dataset. Note the precise object shape estimations despite occlusions and background clutter. Multiple object instances are detected with subsequent hypotheses as is shown in the bottom row (4 left most columns).

time, the experimental results show that the standard localization performance of the method is comparable to the state-of-the-art. Our method performs well on natural images, robustly handling multiple object aspects, significant intra-class variations, occlusions and background clutter.

We have also successfully combined the clustering of the generated hypotheses with a discriminative hypothesis classifier, showing that both elements are necessary for good localization accuracy.

Future research could focus on improving the generated hypotheses by using edges or image segmentations. We think that given a good object localization hypothesis, the object segmentation task should be less difficult and good segmentation accuracy easier to achieve. Furthermore, we will explore the possibility of detecting pose or sub-types by annotating the aspects.

Acknowledgments

M. Marszałek is supported by a grant from the European Community under the Marie-Curie project VISITOR. This work was supported by the European Network of Excellence PASCAL.

References

- [1] S. Agarwal and D. Roth. Learning a sparse representation for object detection. In *ECCV*, 2002.
- [2] O. Chapelle, P. Haffner, and V. Vapnik. Support vector machines for histogram-based image classification. *NN*, 10(5), 1999.
- [3] M. Everingham, A. Zisserman, C. Williams, L. V. Gool, et al. The 2005 PASCAL visual object classes challenge. In *First PASCAL Challenge Workshop*.
- [4] M. Fritz, B. Leibe, B. Caputo, and B. Schiele. Integrating representative and discriminant models for object category detection. In *ICCV*, 2005.
- [5] M. Fussenegger, A. Opelt, and A. Pinz. Object localization/segmentation using generic shape priors. In *ICPR*, 2006.
- [6] J. Gärding and T. Lindeberg. Direct computation of shape cues using scale-adapted spatial derivative operators. *IJCV*, 17(2), 1996.
- [7] B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *ECCV'04 Workshop on Statistical Learning in Computer Vision*, 2004.
- [8] B. Leibe, E. Seemann, and B. Schiele. Pedestrian detection in crowded scenes. In *CVPR*, 2005.
- [9] T. Lindeberg. Feature detection with automatic scale selection. *IJCV*, 30(2), 1998.
- [10] D. Lowe. Object recognition from local scale-invariant features. In *ICCV*, 1999.
- [11] D. Lowe. Distinctive image features form scale-invariant keypoints. *IJCV*, 60(2), 2004.
- [12] M. Marszałek and C. Schmid. Spatial weighting for bag-of-features. In *CVPR*, 2006.
- [13] K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *IJCV*, 60(1), 2004.
- [14] A. Opelt, M. Fussenegger, A. Pinz, and P. Auer. Generic object recognition with boosting. Technical Report TR-EMT-2004-01, TU Graz, 2004.
- [15] A. Opelt and A. Pinz. Object localization with boosting and weak supervision for generic object recognition. In *SCIA*, 2005.
- [16] A. Opelt, A. Pinz, and A. Zisserman. A boundary-fragment-model for object detection. In *ECCV*, 2006.
- [17] F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce. 3D object modeling and recognition using affine-invariant patches and multi-view spatial constraints. In *CVPR*, 2003.
- [18] B. Russell, A. Efros, J. Sivic, W. Freeman, and A. Zisserman. Using multiple segmentations to discover objects and their extents in image collections. In *CVPR*, 2006.
- [19] B. Schölkopf and A. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond*. MIT Press, Cambridge, MA, 2002.
- [20] E. Seemann, B. Leibe, and B. Schiele. Multi-aspect detection of articulated objects. In *CVPR*, 2006.
- [21] J. Shotton, A. Blake, and R. Cipolla. Contour-based learning for object detection. In *ICCV*, 2005.
- [22] A. Thomas, V. Ferrari, B. Leibe, T. Tuytelaars, B. Schiele, and L. V. Gool. Towards multi-view object class detection. In *CVPR*, 2006.
- [23] S. Todorovic and N. Ahuja. Extracting subimages of an unknown category from a set of images. In *CVPR*, 2006.
- [24] J. Willamowski, D. Arregui, G. Csurka, C. R. Dance, and L. Fan. Categorizing nine visual classes using local appearance descriptors. In *IWLAVS*, 2004.
- [25] J. Winn and N. Jojic. LOCUS: Learning object classes with unsupervised segmentation. In *ICCV*, 2005.
- [26] J. Zhang, M. Marszałek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. *IJCV*, 73(2), 2007.