Spatial Configuration of Local Shape Features for Discriminative Object Detection

Lech Szumilas[1] and Horst Wildenauer[2]

[1] Industrial Research Institute for Automation and Measurements
Al. Jerozolimskie 202, 02-486 Warsaw, Poland
lech.szumilas@gmail.com
http://www.piap.pl
[2] Vienna University of Technology
Favoritenstraße 9/1832, A-1040 Wien, Austria
horst.wildenauer@gmail.com
http://www.acin.tuwien.at.at

Abstract. This paper proposes a discriminative object class detection and recognition based on spatial configuration of local shape features. We show how simple, redundant edge based features overcome the problem of edge fragmentation while the efficient use of geometrically related feature pairs allows us to construct a robust object shape matcher, invariant to translation, scale and rotation. These prerequisites are used for weakly supervised learning of object models as well as object class detection. The object models employing pairwise combination of redundant shape features exhibit remarkably accurate localization of similar objects even in the presence of clutter and moderate view point changes which is further exploited for model building, object detection and recognition.

1 Introduction

The study of shape for object description and recognition has a long research tradition, dating back into the early days of the computer vision field. Several recent works have explored the idea of coupling local, contour-based features together with their geometric relations as effective means of discriminating object categories using shape. Promising results in the context of object class recognition and object localization have been achieved, solely operating on boundary-based representations. In [12,10] codebooks of class discriminative shape features, drawn from a corpus of training images, are augmented with geometric relations encoded in pointers to an object instances centroid. A similar representation was suggested by Ferrari et al. [5], however building on a more generic alphabet of shape features, derived from groups of adjacent contour segments. In contrast, [9,3] model global shape by means of ensembles of pairwise relations between local contour features.

In our work we exploit pairwise relationships between local shape fragments to construct a robust shape matching technique that is invariant to translation, scale and rotation. This method allows us to localize objects in the scene using the model based on spatial arrangement of local shape fragments discussed in Section 3. The use of this technique for model extraction as well as object detection and classification is investigated in Sections 4 and 5.

G. Bebis et al. (Eds.): ISVC 2009, Part I, LNCS 5875, pp. 22–33, 2009.

2 The Shape Based Object Model

In this work, an object's shape is represented by an overcomplete set of weak local features together with their spatial relations: Local, edge-based features are embedded in a global geometric shape model, organized as a star-like configuration around an object's centroid. The aspects of this type of representation has been extensively investigated in the context of part-based object class recognition [11,4], showing the advantage of considerably reduced computational burden during training and testing compared to fully connected constellation models. Star-like representations also found successful application in contour-based object detection methods: Opelt et al. [10] and Shotton et al. [12] concentrated on the use of class-discriminative codebooks of boundary fragments, while Ferrari et al. [5] employ a generic alphabet of shape features, derived from groups of adjacent contour segments.

To further increase the explanatory power of the rather weak features, we group them into pairs and exploit their pairwise constraints and their geometric relationship to the centroid to arrive at an efficient matching process which is invariant to changes in translation, scale, and orientation while still being able to handle moderate shape deformations. Ensembles of pairwise relations between edge-based features have been previously used in the same context by [9]. However, the proposed encoding only allowed for translational invariance during the matching process.

3 Features and Matching

Our choice of shape features is based on two criteria: Achieving invariance to translation, scaling and rotation and minimizing the sensitivity w.r.t. edge fragmentation. We start with edges obtained by Canny's edge detector and then coarsely segment each edge into a chain of straight segments by splitting at high curvature points (see Figure 1). Similar in spirit to [5], these edge fragments are used to construct a basic feature that represent local contours in the form of a pair of adjacent segments and the key point at the segment intersection. Segment adjacency is defined in terms of overall distance from the key point to the associated segment boundaries (referred later as "relaxed segment adjacency") – thus allowing pairing of segments that correspond to different edges or non-consecutive segments along the same edge. Although this is a very weak feature that does not allow for reliable scale and orientation estimation, the negative effect of edge fragmentation is compensated by relaxed segment adjacency and the introduced redundancy.

In order to create a feature that is fully invariant to similarity transformation and increase its discriminative power we pair individual key points as shown in Figure 1 (lower-right corner). The key point pair is described by two sets of parameters:

- matching features $f_{ij} = [\beta_{ij1}, \beta_{ij2}, \beta_{ij1}, \beta_{ij2}] \in \mathbb{F}^4$ where $\mathbb{F}$ represents real numbers in the range $-\pi..\pi$ that describe segment angles relative to the vector connecting key points i and j. Note that f_{ij} is invariant to similarity transformations.
- geometrical relationships used for estimation of relative scale, orientation and object centroid location during feature matching $g_{ij} = [d_{ij}, \alpha_{ij}, \Delta x_{ij}, \Delta y_{ij}, \Delta x_{ic}, \Delta y_{ic}]$.

The Δx_{ic} and Δy_{ic} represent spatial relation between $i-th$ key point and the object centroid c. Object centroid is either known (model) or estimated during object detection.

Note that such defined feature pair is an ordered set of key points i and j.

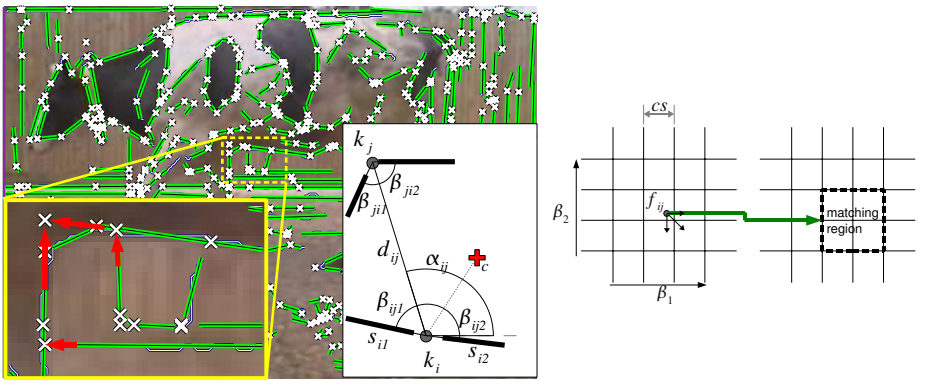


Fig. 1. Left: Example of local shape features. Green lines and white markers depict edge segments and the associated key points respectively. The lower-left corner shows an example of key point association to segments belonging to different edges or non-consecutive segments along the same edge. The lower-right corner shows an example of key point pair. Right: Example of feature discretization (2D case for clarity) that allows for matching of feature sub-sets instead of exostive correspondence estimation (see Section 3.1).

Geometrical relations between local image features have been previously used to disambiguate feature correspondences in object recognition, see Section 2. Here we extend the use of feature pairs (referred as features from now on) to obtain feature matching, registration and object detection that is invariant to translation, scale and orientation.

The problem of feature matching and subsequently fitting the object model m to the feature set from target image t is defined as a three stage process:

1. Feature matching that estimates feature similarity, relative scale and orientation between the model and the target sets as well as the centroid position in the target image. Feature matching also produces soft correspondences between model and target features, where each feature in the model set correspond to k most similar target features. We have chosen $k = 20$ which gives a good balance between accuracy and efficiency of the model fitting.
2. Estimation of potential centroid locations in the target image with Hough-style voting.
3. Iterative model fitting around detected centroids combined with feature correspondence pruning. The model fitting establishes relative scale and orientation between uniquely corresponding features that minimizes global fitting error.

3.1 Feature Matching

The feature matching between model and target sets involves estimation of similarity between key point pairs in the compared features, estimation of relative scale, orientation as well as centroid location for the target features. The dissimilarity between feature p, corresponding to key point pairs ij, from the model set m and feature q, corresponding to key point pairs $i'j'$, from the target set t is obtained by comparing $f_{m,p}$ and $f_{t,q}$:

$$\epsilon_f(p,q) = (|f_{m,p} - f_{t,q}| \mod \pi) \tag{1}$$

The relative scale and orientation are given by $\zeta_{p,q} = d_p/d_q$ and $\omega_{p,q} = ((\alpha_p - \alpha_q) \bmod \pi)$ respectively. Note that because of using key point pairs the relative scale and orientation are non-ambiguous. The estimation of centroid location in the target image is given by:

$$\begin{aligned} x_{ct}(p,q) = \Big(& \Delta x_{ic}(\Delta x_p \Delta x_q + \Delta y_p \Delta y_q) + \\ & \Delta y_{ic}(\Delta y_p \Delta x_q - \Delta x_p \Delta y_q) \Big) / d_p^2 + x_{i'} \\ y_{ct}(p,q) = \Big(& \Delta x_{ic}(\Delta x_p \Delta y_q - \Delta y_p \Delta x_q) + \\ & \Delta y_{ic}(\Delta x_p \Delta x_q + \Delta y_p \Delta y_q) \Big) / d_p^2 + y_{i'} \end{aligned} \tag{2}$$

where $x_{i'}$ and $y_{i'}$ correspond to the position of the first key point in the feature q.

The negative aspect of using feature pairs is higher computational complexity. In a typical case feature matching would compare all possible combinations of features in two feature sets – if the model and target sets contain $K = 1000$ key points each[1] the matching procedure has to compare $(K^2 - K) \times (K^2 - K) \approx 10^{12}$ pair combinations which corresponds to quadratic complexity and leads to prohibitively high execution times. However, due to the simplicity of the feature descriptor we can partition features into a 4D array $\mathbf{F}$ representing a discrete space of $\mathbb{F}^4$. Each cell of the array $\mathbf{F}$ contains a sub-set of features corresponding to the cell span in $\mathbb{F}^4$, thus the matching of features in a single cell of the array $\mathbf{F}_m$ (model features) is confined only to the same cell in the array $\mathbf{F}_t$ (target features) and adjacent cells as shown in Figure 1. The cell span cs defines the similarity threshold at which relative segment angles are no longer compared. We have chosen a conservative value of $cs = 30^\circ$ which allows maximum angle difference between two segments of 45° and produces 12^4 cells in the array. The efficiency benefit of this solution depends on the particular distribution of features in $\mathbf{F}_{m,t}$ spaces e.g. when features are distributed uniformly the speed up factor equals to $\frac{1}{3}12^4$. Typical matching times range from below a second ($K = 200$) to about 30 seconds ($K = 1000$ - complex scenes) on a 3GHz multi-core processor (which can be further improved by using GPU)[2].

The feature matching produces a set of soft correspondences, allowing association between a single model feature and k target features, which is a necessary measure to account for feature ambiguity and presence of multiple similar objects in the scene. The feature correspondences produce a centroid estimates according to (2) which are accumulated to generate hypotheses about the object location in the analyzed scene using a

[1] The images in the tested databases produce between 200 and 1000 key points.

[2] For simplicity we assumed that model and target have identical number of key points.

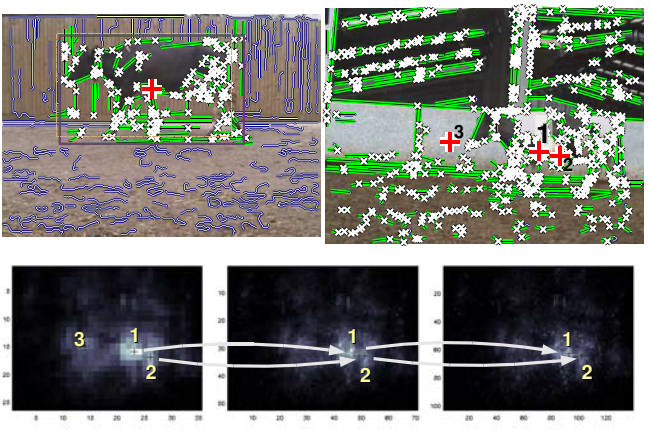


Fig. 2. The top row shows locations of estimated centroids in the target image (right image) given the model obtained from the features enclosed by the bounding box (left image). Despite the noisy model that contains also elements of the background, significant edge fragmentation and the differences in appearance (textures resulting in additional clutter) the strongest voting maximum is closely aligned with the true location of the object centroid. The bottom row shows accumulator arrays and voting maxima tracking across different resolutions. Images are best viewed in color.

Fig. 3. Example of the object localisation by matching and fitting a noisy model (contents of the bounding box in the left image). The second image shows the features extracted from the target image and the estimated location of the object centroid (strongest voting maximum). The right figure shows the alignment of the model (green, thick lines) with the uniquely corresponding features in the target image (red, thin lines). Note the amount of clutter in the target image.

Hough-like voting scheme [2]. The spread of the accumulated centroid votes depends on factors such as the amount of clutter present, overall shape similarity between model and target objects, and the relative scale at which features have been extracted [7]. Since this cannot be established a-priori, we adopt a simple multi-resolution refinement step, searching for voting maxima which are stable across different levels of granularity of the accumulator array as shown in Figure 2.

3.2 Model Fitting

Depending on the allowable degrees of freedom (e.g., rigid and non-rigid deformations), finding correspondences between model and image features often poses a costly combinatorial problem which gets quickly out of hands for more than a rather moderate

number of features involved. Typically, efficient strategies for searching less then optimal matching solutions are adopted to make the problem more tractable. Among those, approaches based on Integer Quadratic Programming [3], graph cuts [13], and spectral matching [8] have been shown to give excellent results in the context of object class recognition.

However, despite their efficiency, the number of features that can be coped with is limited to few hundred. Since our approach operates on a large number of feature pairs, the amount of initial pair correspondences to be optimized requires a more efficient approach. E.g. for $K = 1000$ model key points and $k = 20$ correspondences per key point pair we obtain up to $(K^2 - K) \times k \approx 20 \times 10^6$ soft correspondences, unfortunately ruling out the use of the aforementioned methods. Therefore, we adopted a more practically usable and efficient procedure based on iterations of coarse model alignment and feature pruning.

Specifically, an initial model position is obtained from the estimated centroid, while relative scale and orientation are estimated from the soft feature correspondences that casted votes for the centroid[3]. Since the initial estimation of position, scale and orientation cannot be expected to be accurate, it is optimized in an iterative process that combines model fitting and soft-correspondence pruning until unique correspondences are found. Here, due to the centroidal alignment, only a moderately sized sub-set of soft correspondences voting on the centroid has to be processed in subsequent iterations of the fitting procedure.

The following simplified fitting procedure is repeated for every centroid:

1. Obtain a list of soft correspondences that casted votes for the centroid (the list is produced during voting for each maximum in the voting accumulator at lowest resolution) $\mathbf{C} = [(p_1, q_1), (p_2, q_2), ..., (p_M, q_M)]$, where (p_m, q_m) are indexes of corresponding pairs in the model and target sets respectively. The correspondences are weighted (w_m) inversely proportional to the distance between their vote and the position of maximum in the voting accumulator.
2. Estimate scale $\overline{\varsigma}$ and orientation $\overline{\omega}$ of model relative to the corresponding target features:
$$\overline{\varsigma} = \frac{1}{\sum_M w_m} \exp\left(\sum_M w_m \log\left(\frac{d(q_m)}{d(p_m)}\right)\right) \tag{3}$$
where $d(p_m)$ and $d(q_m)$ are spatial distances between key point pairs p and q respectively.
$$\overline{\omega} = \frac{1}{\sum_M w_m} \sum_M w_m \left((\alpha_q - \alpha_p) \mod \pi\right) \tag{4}$$
3. Transform the model: scale by the factor $\overline{\varsigma}$, rotate by $\overline{\omega}$ and translate to the target centroid.
4. Estimate a similarity score $s_{p,q}$ for corresponding features that is a combination of spatial misalignment $\epsilon_s(p, q)$ and feature similarity $\epsilon_f(p, q)$:
$$s(p, q) = \exp\left(-\frac{\left(\epsilon_s(p, q) + \overline{\varsigma}\sigma_s\epsilon_f(p, q)\right)^2}{2\left(\overline{\varsigma}\sigma_s\right)^2}\right) \tag{5}$$

[3] The fitting is repeated for each centroid detected in the target image.

where $\epsilon_s(p,q)$ is an Euclidean distance between transformed model key points and target key points in the corresponding features and σ_s is a parameter which binds spatial and angular alignment errors. Proposed measure produces similarity score 1 for perfectly aligned features ($\epsilon_s(p,q) = 0$ and $\epsilon_f(p,q) = 0$) and approaches 0 when $\epsilon_s(p,q) >> \sigma_s$ or $\epsilon_f(p,q) \rightarrow 2\pi$. All results presented in this paper were achieved with σ_s set to 0.1 of the maximum model extent (bounding box) although our experiments has shown that range between 0.05 and 0.2 produces almost identical fitting results.

5. Find all model features p_r that correspond to more than one target feature and for each feature p_r discard a correspondence that produced minimum similarity score.
6. Return to step 2 if any of the model features correspond to multiple target features.

Examples of feature matching and model fitting are shown in Figure 4.

4 Model Extraction

Our primary concern is the construction of object class model that contains a sufficient number of discriminative and repeatable features to maximize accuracy of object detection and classification.

In [10] and [12] the initial set of training features is reduced using a simple clustering technique and the discriminative features are selected by a training stage based on AdaBoost. Our approach follows this scheme. However, instead of initial feature reduction we produce a set of "sub-models" that represent groups of geometrically similar object instances in the training data set. The aim of sub-models is to capture a distinctive shape variations within the whole training set in terms of overall shape similarity and centroid localisation accuracy (see Figure 5). Such partitioning allows as to a) build more specific object models that increase fitting accuracy, b) minimize matching complexity and c) obtain more accurate feature alignment than it is possible with ordinary clustering approach.

The extraction of sub-models is a pre-processing step before the learning discriminative model, meant as a coarse data partitioning. The purpose of sub-models is to obtain a compact feature set from similar object instances and ensure that each sub-model preserves geometrical characteristics of represented shape. The sub-model extraction procedure consist of object instance grouping and feature compacting as follows:

1. Grouping starts with matching object instances in the training set, giving an estimate of global shape similarity and centroid estimation accuracy for every matched pair. The global shape similarity between an instance a and b is an average of feature similarities (5) obtained from model fitting $S_{a,b} = (\sum_M s(p_m, q_m))/M$ (instance a is the model and instance b is the target). Note that these estimates are asymmetric in general ($S_{a,b} \neq S_{b,a}$) due to different number of features in both instances and potential presence of non-repeatable background inside the bounding boxes. For that reason a symmetric similarity between two instances is defined as $\widehat{S}_{a,b} = S_{a,b} + S_{b,a}$.
2. Object instances are grouped using a hierarchical clustering on the global shape similarity $\widehat{S}$ with an additional constraint on maximum allowed centroid error $e_c(a,b)$

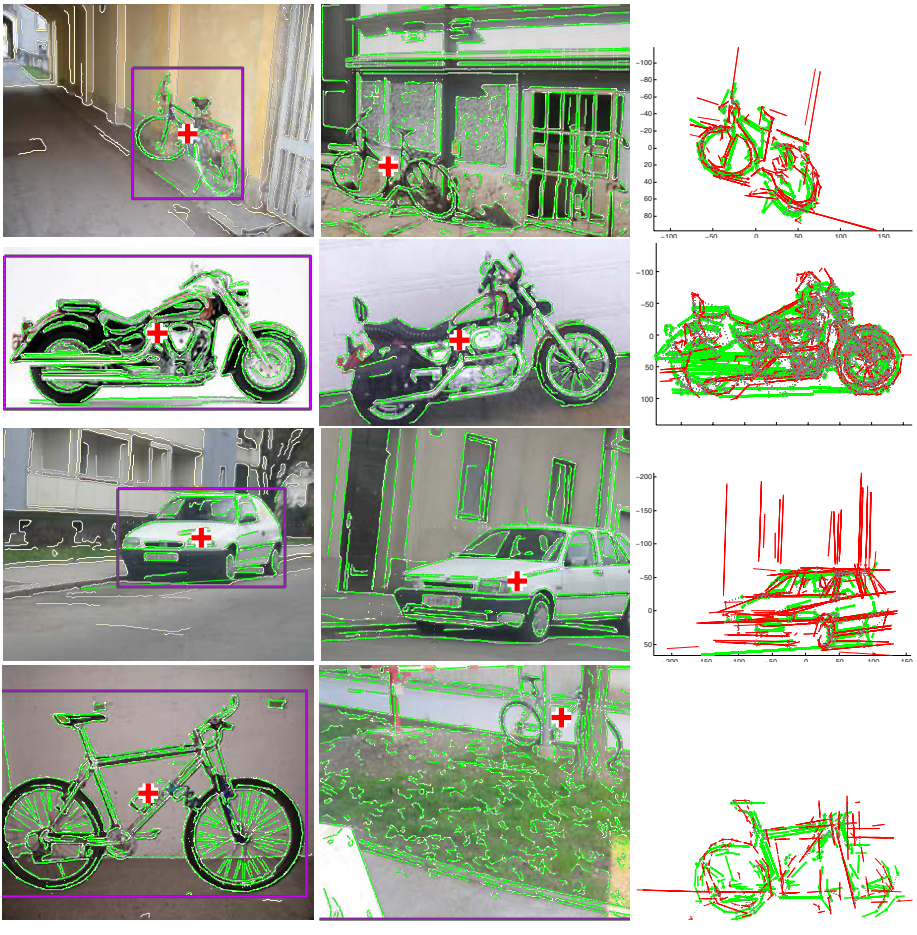


Fig. 4. Examples of object localisation (using noisy models) in the presence of scale, orientation, view point change and occlusion. The first column contains images of the model (enclosed by the bounding box) that are matched to the target images in the second column. Estimated centroids are shown in the target images while the model fitting is visualized in the third column.

and scale estimation error $e_s(a, b)$. The centroid localisation error is an Euclidean distance between detected and true positions of the centroid relative to the bounding box of instance b while the scale estimation error compares $\overline{\varsigma}$ (3) to the relative scale of bounding boxes in instances a and b. These constraints ensure that object instances with high centroid and scale estimation errors will not be grouped together. We have used conservative error thresholds $e_c < 2\sigma_s$ and $0.75 < e_s < 1.3$ (relative scale) for all evaluated image databases. The centroid and scale accuracy constraints typically result in 8-12 groups (see Figure 5 as an example).

3. In the final step features are compacted within each group of object instances. Corresponding key points from different object instances (exhibiting both feature similarity and global spatial alignment) form cliques that are averaged into a single

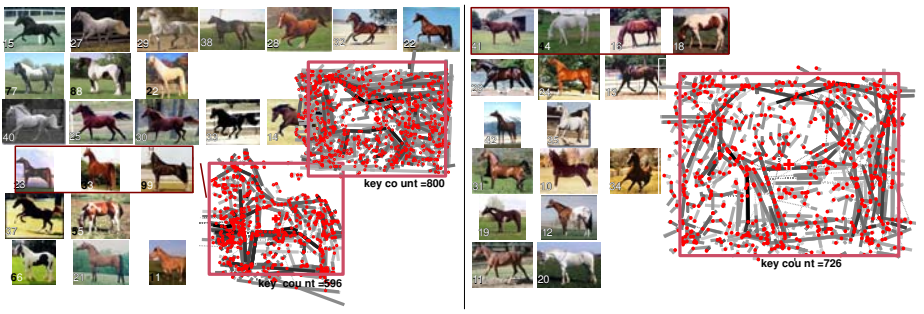


Fig. 5. Example of object instance grouping in the training data set based on overall shape similarity, centroid detection and scale estimation accuracy. Each row contains a group of similar object instances (note the figure is split into two columns). The final result depends on the training data, intra-class variability and the bounding box background variability. This example shows that it is possible to obtain meaningful groups of real objects with a low number of outliers. Resulting sub-models (only a subset is shown for clarity) display a reduced set of features, each visualized with a gray intensity corresponding to the associated strength.

model key point. A key point clique can be viewed as a connected graph with key point based nodes and correspondence based edges. The strength of the resulting key point is a sum of similarity scores (5) from all correspondences between merged key points and is used as a weight during casting centroid votes. Examples of sub-models produced by key point merging are shown in Figure 5.

The outline of our final feature selection and classifier learning is as follows. We combine sub-models into a global set of a spatially related features which will be pruned during the learning process. The sub-models are matched to the validation images to obtain a set of positive and negative training examples in terms of similarity and alignment of individual features. The role of sub-models is to localise and estimate pose of similar objects or shape structures in the validation images. The positive and negative examples however contain similarity scores (5) of every feature in the global model set that are transformed according to previously estimated pose and centroid location. Positive examples are obtained whenever one or more sub-models locates the same type of object in the validation image while negative examples are drawn from other object types and the background. The final object classifier and feature selection are produced by applying the Gentle-Boost learner to the set of positive and negative examples.

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

$$H(d) = \sum_{M} a_m \Big(s(p_{m,d}, q_{m,d}) > \theta_m \Big) + b_m \tag{6}$$

where a_m, b_m and θ_m are learned parameters, d indicates a particular centroid/pose detection (more than one possible per image) while $p_{m,d}$ and $q_{m,d}$ are corresponding model and target features (model features are transformed). Depending on the training data sets the typical number of discriminative features selected varied between 300 and 400. The features that were dropped during classifier training are also removed from sub-models.

The extraction of training examples plays a critical role in obtaining a robust classifier. These examples must account for inaccuracy in centroid and pose estimation that is caused by the intra-class shape variability or change of view related to projective transformation. The examples produced by matching of sub-models to the validation images must be therefore artificially expanded by injecting potential errors into centroid and pose estimation in a similar manner as in [6,12]. These additional examples are produced by computing alignment of features and thus similarity measures for displaced centroid positions and slight scale variations. This procedure produces not only an additional positive examples (around the true centroid position) but also negative examples when the model is shifted toward the boundary of the bounding box in the validation image.

The process of feature selection and classifier training is repeated for every database separately. The negative examples used for boosting are obtained from the training images of the trained class (background outside of the bounding box) and training sets of other classes. This is done to obtain an object class detector that is not only able to discriminate object of particular type from a typical background but also to discriminate it from other object classes.

5 Evaluation

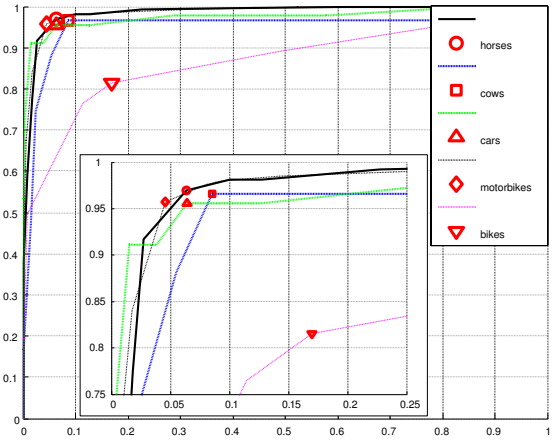
We test our approach on five databases listed in Table 1 that has been previously used for evaluation of other shape based detectors. We select a relatively small number of images ($< 10\%$) for sub-model extraction and another set of images to serve as validation data. The overall training data set do not exceed 25% of the whole database in each case. Tests were conducted on the combined set of test images drawn together from all databases.

To evaluate our approach we measure object detection and image classification accuracy for each object class separately. By object detection we understand localisation and classification of object instances as follows. We use sub-models to produce hypotheses d on object location (centroid) as described in Section 3.1 and 3.2. Next, the classification score $H(d)$ (6) is computed for each hypothesis. We use a simple non-maxima suppression on $|H(d)|$ to locally eliminate "weak" detections in overlapping regions (within 50% of the bounding box area). Remaining hypotheses d are classified as an "object" $H(d) > \Theta$ or "background" $H(d) \leqslant \Theta$, where Θ is a global confidence threshold regulating trade off between true and false positives. Resulting classification is compared against the ground truth to produce statistics on the number of true and false positives as a function of threshold Θ. A particular detection is associated with the object class if the area overlap between the detected (scaled) bounding box and the annotated bounding box is greater than 50% (assuming it contains the same object type) [1]. For the image classification results, the detection exhibiting the strongest classification confidence $\max_d |H(d)|$ is used to decide whether an instance of the object class is present in the image or not.

Table 1 provides the image classification and object detection accuracy along with the receiver operating characteristic (ROC) curve for the image classification case. The result of our evaluation gives an indication of how well the particular object class is discriminated against the background and *other object classes*. This is in our opinion

Table 1. Classification and detection performance on 5 image databases. The second column shows the true positive ratio for image classification at equal error rate (EER). Third column shows the area under ROC curve (ROC-AUC). The fourth column represents the area under Precision-Recall curve (PR-AUC) for the detection of object instances. Right: The ROC curve represents image classification accuracy for each of the tested databases, showing a trade off between true positives and false positives as the global confidence threshold is varied.

database	tp (EER)	ROC-AUC	PR-AUC
horses (Weizmann)	0.970	0.985	0.993
cows (Darmstadt)	0.966	0.947	0.829
cars (Darmstadt)	0.956	0.975	0.909
motorbikes (Caltech)	0.957	0.986	0.993
bikes (Graz)	0.815	0.867	0.782



a more realistic and challenging test scenario than the typical object detection against background only [12,10].

We benchmark our method against state-of-the-art approach from Shotton et al. [12]. Our approach achieves particularly good performance on the database of horses (PR-AUC of our method 0.993 vs. 0.968/0.785 in [12]) and bicycles (our 0.782 vs. 0.6959 [12]) considering that they were not split into side/front views as in [12]. Detection accuracy of motorbikes is almost identical in the two methods. Detection accuracy of Cows is worse than in [12], however the problem is primarily related to cows being confused with horses (not done in [12]) as well as imbalance in the number of test images (5:1) between these two databases.

6 Conclusions

We have presented a novel shape matcher and its application to discriminative object recognition. The shape matcher efficiently utilizes pairs of local shape fragments for robust model localisation. Although feature pairs have been previously exploited for matching and object recognition, we extend their use to provide invariance to rotation and scale effortlessly. Reported results show that our approach tolerates moderate view point changes, clutter and partial object occlusion (see Figures 4). Evaluation of object detection accuracy proves that the method is capable of outperforming state-of-the art detectors on challenging databases, containing multiple views of the same object class.

Our analysis of the method properties indicates that the combination of redundant features and the use of feature pairs plays a crucial role in object localisation and pose

estimation while the use of sub-models (Section 4) significantly improves object detection accuracy. The use of multiple object models per class, feature sharing between these models and verification of different model extraction approaches is a primary focus of future work.

Acknowledgements

The research has received funding from the EC grant FP6-2006-IST-6-045350 (robots@home).

References

1. http://www.pascal-network.org/challenges/VOC
2. Ballard, D.H.: Generalizing the hough transform to detect arbitrary shapes. Pattern Recognition 2(13) (1981)
3. Berg, A.C., Berg, T.L., Malik, K.: Shape matching and object recognition using low distortion correspondences. In: CVPR, pp. 26–33 (2005)
4. Crandall, D., Felzenszwalb, P., Huttenlocher, D.: Spatial priors for part-based recognition using statistical models. In: CVPR, pp. 10–17 (2005)
5. Ferrari, V., Fevrier, L., Jurie, F., Schmid, C.: Groups of adjacent contour segments for object detection. T-PAMI 30(1) (2008)
6. Laptev, I.: Improving object detection with boosted histograms. In: Image and Vision Computing (2008)
7. Leibe, B., Leonardis, A., Schiele, B.: Robust object detection with interleaved categorization and segmentation. IJCV 77(1-3), 259–289 (2008)
8. Leordeanu, M., Hebert, M.: A spectral technique for correspondence problems using pairwise constraints. In: ICCV, vol. 2, pp. 1482–1489 (2005)
9. Leordeanu, M., Hebert, M.: Beyond local appearance: Category recognition from pairwise interactions of simple features. In: CVPR, pp. 1–8 (2007)
10. Opelt, A., Pinz, A., Zisserman, A.: A boundary-fragment-model for object detection. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3952, pp. 575–588. Springer, Heidelberg (2006)
11. Fergus, P.P.R., Zisserman, A.: A sparse object category model for efficient learning and exhaustive recognition. In: CVPR, pp. 380–387 (2005)
12. Shotton, J., Blake, A., Cipolla, R.: Multiscale categorical object recognition using contour fragments. T-PAMI 30(7), 1270–1281 (2008)
13. Win, J., Jojic, N.: Locus: learning object classes with unsupervised segmentation. In: ICCV, pp. 756–763 (2005)