Sparsity Potentials for Detecting Objects with the Hough Transform

Nima Razavi[1]
nrazavi@vision.ee.ethz.ch
Nima Sedaghat Alvar[2]
n.sedaghat@gmail.com
Juergen Gall[3]
juergen.gall@tue.mpg.de
Luc van Gool[1,4]
luc.vangool@esat.kuleuven.be

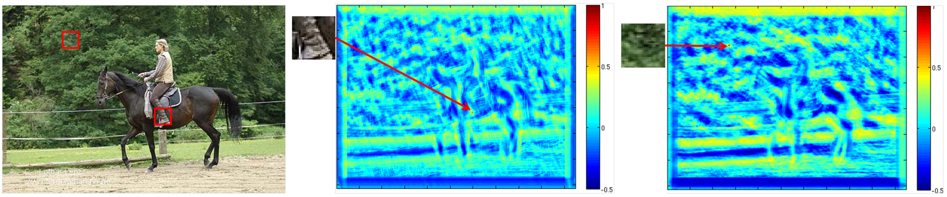
[1] Computer Vision Laboratory
ETH Zurich, Switzerland
[2] Machine Vision and Intelligence Lab
Sharif University of Technology, Iran
[3] Perceiving Systems Department
MPI for Intelligent Systems, Germany
[4] IBBT/ESAT-PSI
K.U. Leuven, Belgium

Abstract

Hough transform based object detectors divide an object into a number of patches and combine them using a shape model. For efficient combination of patches into the shape model, the individual patches are assumed to be independent of one another. Although this independence assumption is key for fast inference, it requires the individual patches to have a high discriminative power in predicting the class and location of objects. In this paper, we argue that the sparsity of the appearance of a patch in its neighborhood can be a very powerful measure to increase the discriminative power of a local patch and incorporate it as a sparsity potential for object detection. Further, we show that this potential shall depend on the appearance of the patch to adapt to the statistics of the neighborhood specific to the type of appearance (e.g. texture or structure) it represents. We have evaluated our method on challenging datasets including the PASCAL VOC 2007 dataset and show that using the proposed sparsity potential result in a substantial improvement in the detection accuracy.

1 Introduction

Despite recent progresses, object detection remains a challenging task in computer vision. One of the most important reasons behind this difficulty is the problem of dealing with the intra-class variability among instances of an object class and being able to generalize to unseen instances. Hough-transform based object detectors [9, 14] deal with this problem by dividing an object into a number of local patches aiming at less variation within patches. These patches are then combined using a shape model to ensure spatial consistency among them. In these models, the hope is that although the overall appearance of a new instance may have not been observed in the training data, at the local patch level, similar patches have been observed. To this end, the local appearances of the training data are clustered to form a codebook which should ideally encode all the possible patterns that a foreground patch can take [2, 9, 14].



(a) (b) (c)

Figure 1: The patches in an image exhibit different sparsity values. (a) Consider the two patches on the foot of the rider and the leaves of a tree. The images in (b) and (c) show the normalized cross correlation of these patches with the image. While the self-similarity of a non-texture patch like the foot to its neighboring patch is low, the patch on the tree is less sparse and much more similar to the patches in its neighborhood. Base on this observation, in this paper, we introduce a measure which captures the *sparseness* of a patch within its neighborhood and incorporate it as a "sparsity potential" for object detection.

When detecting an object with the Hough transform, given an object hypothesis, the individual features are considered to be independent of one another. Although this independence assumption makes the inference with these models very efficient, it requires the codebook entries to have high discriminative power in predicting the class and location of objects. Previously, considerable work has been done on how to discriminatively learn a codebook of local appearances [9, 14, 16, 26]. Yet, very little attention has been given to the statistics of the patterns in a test image which is unobserved at training time.

The patterns in an image can range from simple textures to complex shapes. Likewise a local patch in an image can be an element of a textured region or a complex shape region. In the case that the local patch is a part of a textured region, we expect to see many similar patches in its neighborhood which is the way a texture is defined. In contrast, if this local patch is part of a complex structure in an image, it is unlikely for its neighboring patches to have similar appearances.

Based on the self-similarity of a patch with the patches in its local neighborhood, we define the sparsity of a patch as a measure of the uniqueness of its appearance in its local neighborhood. We argue that this measure can be used to compactly summarize the information in an immediate context of a patch and thus increase its discriminative power. In particular, we make the following two observations

- the similarity in appearance of patches in a neighborhood of a central patch exhibit different sparsity values when the central patch appears on an object as opposed to a background region.
- the codebook entries associated with texture or simple edge patterns are consistently less sparse in their neighborhood as opposed to entries which are associated to more complex patterns (see Fig. 1).

Based on these observations, we utilize the sparsity of the local self-similarity in conjunction with the appearance of the central patch to increase discriminativity of local patches. Specifically, for every codebook entry a classifier is trained on the sparsity values to classify a patch. Calculating the self-similarities with Normalized Cross Correlation [20] or image specific codebooks [4] is computationally very expensive. to overcome this limitation, the local self-similarities are efficiently calculated using the ISM codebook by assuming the patches with similar assignments to be self-similar.

We have evaluated our method on the challenging PASCAL VOC 2007 dataset [5] and show that the proposed sparsity measure substantially improves the results although coming at very little computational overhead. The proposed potential is also quite general and can be easily integrated in other Hough-transform based detectors or used in other domains.

2 Related Work

In the recent years, several methods have been proposed for measuring saliency of image [11, 12, 23] to focus the attention on regions with distinctive patterns. Similarly a number of approaches have been proposed to detect salient key points in an image. These works are mainly used to extract sparse image features that are invariant under certain geometric transformations, e.g. scale [13, 15] or affine [17] invariance. Similar to those approaches, in this work, we are also aiming at measuring the sparsity of regions in an image to separate foreground from background. However, we differ from these approaches since firstly our sparsity measure is class-specific instead of a generic measure as in those works. Secondly, in our approach we use dense features and no non-sparse image region is discarded a-priori. Although detecting objects with sparse features can be quite fast, it has been shown previously [2, 6] that superior performance can be achieved by dense sampling of features.

Learning the codebook of local appearances has been the subject of many investigations as well [2, 9, 14]. Although the codebook is obtained generatively by clustering local patch appearances in [14], the class label and spatial distribution of the patches are used in [9] to discriminatively learn the codebook in a random forest framework. While these approaches cluster training patches, Boiman *et al.* [2] take a non-parametric approach and directly retrieve the nearest training patches without performing any quantization. Learning the spatial distributions of a codebook entry is done generatively in [9, 14] whereas in [16] and [26], a max-margin approach is taken to learn weights for codebook entries and training images respectively.

Our self-similarity based sparsity potential is also related to the self similarity features. [20] proposed the local self-similarity (LSS) feature as an appearance independent feature which encodes the spatial location of self-similar patches in a neighborhood of a patch. The LSS has subsequently been integrated in several image classification benchmarks and have shown to consistently improve classification accuracy (e.g. see [10]). This feature has been extended by [4] as a global self-similarity measure. Our proposed sparsity potential can also be interpreted as a self-similarity feature. However, in our approach, the self-similarity is used to measure the sparsity of a patch in its neighborhood. To this end, this measure is used conditioned on the appearance of the central patch to classify a patch as foreground or background.

Several approaches have previously used the information around a single patch for stronger classification. Similar to our work, in [21] all patches are assigned to a codebook and a classifier is trained on the cluster assignment of neighboring patches to improve the classification performance. Similarly, in [8] a bag of words model [3, 22] is learned on neighboring superpixels for classification of a superpixel. In [25], according to certain proximity rules, the close patches have also been grouped together to cast more accurate votes for object detection. Yet, to the best of our knowledge, we are the first to propose the self-similarity sparsity of a neighborhood for object detection.

3 Object Detection with Sparsity Potentials

Before explaining how the detection is done with the proposed sparsity potentials, we briefly give an overview of probabilistic object detection with the Implicit Shape Model [1, 14]. In this model, for detecting objects in a test image I, a number of features $\mathbf{f} = \{f_i = (I_i, l_i, s_i)\}$ are first extracted from I, where I_i represents the appearance of the feature and l_i and s_i its location and scale in the image pyramid. An object hypothesis $h \in \mathcal{H}$ in the ISM is parametrized by a category label c_h, a location l_h and a scale s_h. The detection approach is to estimate the probability density $p(h|\mathbf{f})$.

In an ISM, all features f_i in an image are considered to be independent of one another and a uniform prior is assumed over all patches and hypotheses. Therefore, $p(h|\mathbf{f})$ is approximated by the product of the individual probabilities $p(h|f_i)$:

$$p(h|\mathbf{f}) \approx \prod_{i=1}^{|I|} p(h|f_i). \tag{1}$$

The probabilities $p(h|f_i)$ are obtained by matching a feature's appearance I_i to a codebook of local appearances $\Omega = \{\omega_1, \dots, \omega_j, \dots, \omega_{|\Omega|}\}$ as

$$p(h|f_i) = \sum_{j=1}^{|\Omega|} p(h|f_i, \omega_j) p(\omega_j|f_i) \tag{2}$$

where $p(h|f_i, \omega_j)$ encodes the conditional votes of the feature f_i for hypothesis h given the assignment of f_i to ω_j. The conditional votes are estimated non-parametrically by storing the relative offsets of training patches with respect to training object centers and optionally the class probability at each codebook entry ω_j. As mentioned earlier, several previous approaches have been proposed for creating the codebook Ω. In this paper, we use the Hough Forests [9] for this purpose, where the leaves of the decision trees are interpreted as codebook entries.

It is common among practitioners to use the sum of the probabilities as in [14] instead of their products [1] to get an outlier insensitive estimate [18]:

$$p(h|\mathbf{f}) \approx \sum_{i=1}^{|I|} \sum_{j=1}^{|\Omega|} p(h|f_i, \omega_j) p(\omega_j|f_i) \tag{3}$$

The assignment probability $p(\omega_j|f_i)$ usually depends only on the appearance of the feature, I_i, the class probability of c_h depends only on the matched codebook entry ω_j, and the location l_h and scale s_h of the hypotheses on the location l_i and scale s_i of the feature and ω_j. A common ISM model is therefore given by

$$p(h|\mathbf{f}) \approx \sum_{i=1}^{|I|} \sum_{j=1}^{|\Omega|} p(l_h, s_h|l_i, s_i, \omega_j, c_h) p(c_h|\omega_j) p(\omega_j|I_i). \tag{4}$$

Once the posterior probability over all object hypotheses is estimated, the objects are detected as local maxima of this distribution by mean-shift mode seeking [14, 24] or iteratively found by detecting the most probable one and removing its features [1]. In this paper, we use the feature removal technique as it produces more accurate results and does not require any further non-maxima suppression.

3.1 Sparsity Potentials

In object detection with the Hough transform, the probabilities $p(h|f_i,\omega_j)$ and $p(\omega_j|I_i)$ depend on a codebook that captures the appearance of relatively small patches. As explained earlier in Section 1, the information in this local observation is limited which makes the estimation less reliable. In this paper, we argue that the sparsity of the appearance of a patch in its neighborhood can be a very powerful measure for summarizing the information of a much larger patch and thus can be used to increase the discriminative power of the local patches. To this end, we assume that $p(h|f_i)$ in Eq. (2) not only depends on its appearance I_i, but also on the appearance of the features appearing in a neighborhood around it denoted by $\mathcal{N}^i$. Thus, (3) becomes

$$p(h|\mathbf{f}) \approx \sum_{i=1}^{|I|}\sum_{j=1}^{|\Omega|} p(h|f_i,\omega_j,\mathcal{N}^i)p(\omega_j|f_i,\mathcal{N}^i). \tag{5}$$

In this work, we focus only on the class probability term $p(c_h|\omega_j)$ that is now estimated by $p(c_h|\omega_j,\mathcal{N}^i)$ while the other terms are still estimated independently of the appearance neighborhood. Equation (4) thus becomes

$$p(h|\mathbf{f}) \approx \sum_{i=1}^{|I|}\sum_{j=1}^{|\Omega|} p(l_h,s_h|l_i,s_i,\omega_j,c_h)p(c_h|\omega_j,\mathcal{N}^i)p(\omega_j|I_i). \tag{6}$$

The probability $p(c_h|\omega_j,\mathcal{N}^i)$ can now be modeled such that not only the discriminative power of the patch estimated from the training data, but also its ability in distinguish between foreground and background in a particular test image is taken into account. To this end, we measure the sparsity of the feature within a local neighborhood in the test image.

3.2 The Sparsity Measure

In this work, we base our sparsity or distinctiveness measure on self-similarity [4, 20]. Let us assume that we have a metric that measures the similarity of a patch f_i with all patches in its neighborhood, $\{f_n|n \in \mathcal{N}^i\}$, e.g., NCC as in [20]. Further, we assume that the returned similarity is normalized to be in $[0,1]$ with 1 representing the most similar and 0 the most dissimilar patch. In this case, one is getting a real valued self-similarity vector $\mathbf{ss}_i = (ss_1,\ldots,ss_{|\mathcal{N}^i|})$ where each element ss_n records the normalized similarity of I_n to I_i.

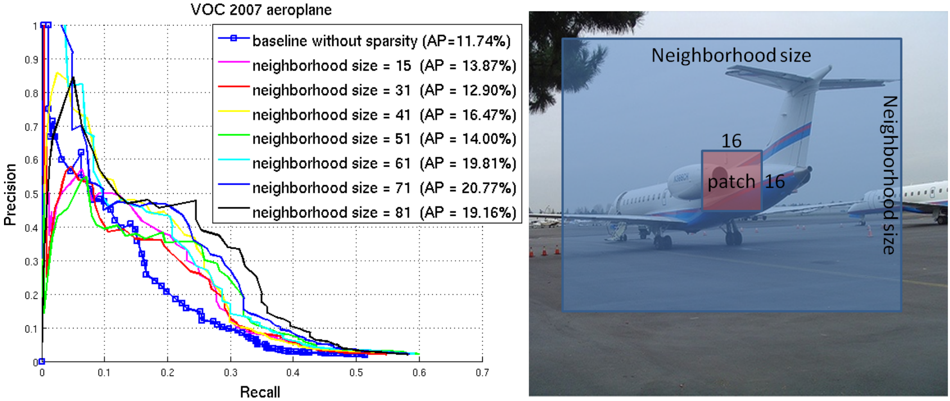
The sparsity of the self-similarity vector $\mathbf{ss}_i$ can be measured in many different ways, e.g., by using entropy or various vector norms. In this work, we use the L1-norm,

$$||\mathbf{ss}_i||_1 = \sum_{n\in\mathcal{N}^i} |ss_n| \tag{7}$$

which is both simple and fast to calculate.

3.3 Calculating the Self-Similarity Efficiently

Measuring the self-similarity by cross correlation of patches in the feature space, as done in [20], can be quite time consuming and is not appropriate for object detection. To efficiently calculate the self-similarities, Deselaers *et al.* [4] proposes to cluster the patches in an image into an image specific codebook and assume that the patches assigned to the



(a) (b)

Figure 2: This figure evaluates the effect of the neighborhood size used for calculating the sparsity on the accuracy of the detector. (a) The comparison of the detection performance of our baseline [19] with our proposed sparsity measure with various window sizes on the test set of ‘aeroplane’ category of the PASCAL VOC 2007 dataset. As can be seen, the proposed sparsity potential always improves the accuracy. The performance tends to increase with the window size until it saturates at around 71 pixels, almost doubling the average precision (AP) compared to the baseline. (b) The sparsity potential is calculated on a square neighborhood of every 16 × 16 patch.

same codebook entry are self-similar. Although, this can be done faster, performing this on the image pyramid densely as required in our setup is quite challenging. However, since in detection with an ISM, the image patches are already assigned to the entries of a large codebook, these assignments can be directly used to calculate the self-similarities.

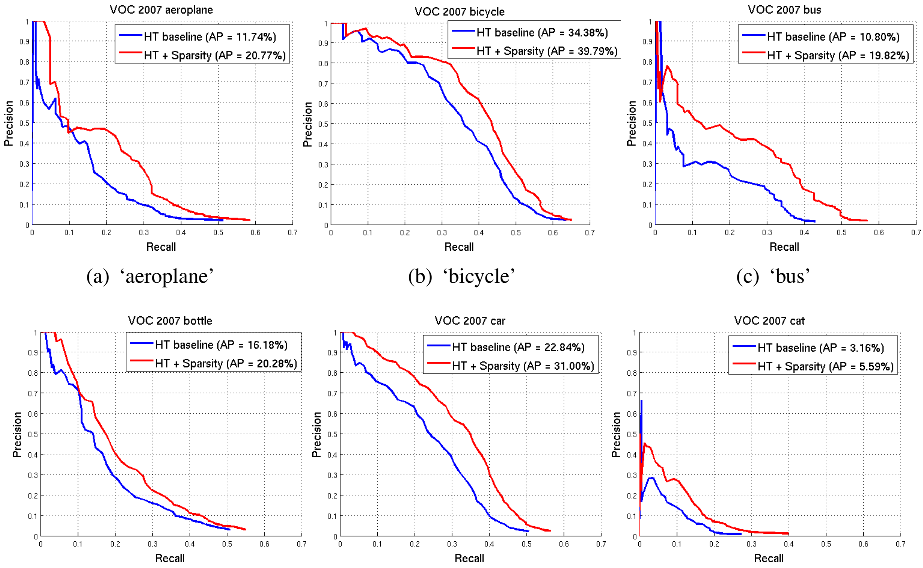
For each patch in the scale pyramid of a test image, a self-similarity score in Equation (7) needs to be measured. To this end, the features at one level are passed to the Hough Forest codebook [9], their leaf assignments are recorded and for each tree an assignment image is formed. Each pixel i in the assignment image records its matching leaf index, i.e., the codebook entry ω_j. For each image feature f_i with a leaf assignment ω_j, all the neighboring features which are also assigned to ω_j are considered to be self-similar and their similarity is set to 1. The similarity of all other patches in the neighborhood is set to zero. Therefore, (7) can be written as

$$||\mathbf{ss}_i||_1 = \sum_{k \in \mathcal{N}^i} \delta_{\omega_j,\omega_k} \tag{8}$$

where $\delta_{\omega_j,\omega_k}$ denotes the Kronecker delta. Given the assignment image, all sparsity measures can be efficiently calculated by decomposing the assignment matrix based on unique leaf assignments.

3.4 Discriminative Learning of the Sparsity Classifiers

For training the sparsity classifiers, first a set of features on the validation set, both on objects as well as background, are extracted and are assigned to one or more codebook entries ω_j. Given a neighborhood function, the sparsity measure of every feature is calculated. Next, for each ω_j and class label c, these sparsity measures for different class labels are collected and used to learn a simple threshold θ_{c,ω_j}. These thresholds are then used to calculate



(a) ‘aeroplane’ (b) ‘bicycle’ (c) ‘bus’

(d) ‘bottle’ (e) ‘car’ (f) ‘cat’

Figure 3: The precision recall curves for some categories of the PASCAL VOC 2007 [5]. As can be seen, the proposed sparsity potentials substantially improve the detection performance. The average precision (AP) is calculated by the integral under the curve.

$p(c_h|\omega_j,\mathcal{N}^i)$ in Equation (6) as

$$p(c|\omega_j,\mathcal{N}^i) \propto \begin{cases} p(c|\omega_j) & \text{if} \quad ||\mathbf{ss}_i||_1 \leq \theta_{c,\omega_j} \\ 0 & \text{otherwise,} \end{cases} \tag{9}$$

$p(c|\omega_j)$ is the class probability estimated at the codebook entry ω_j.

Using the sparsity measure as a single dimensional feature, the thresholds are learned such as to separate the positive and negatives with the best classification accuracy with zero false negatives on the training data.

4 Experiments

In the previous section we have proposed the sparsity potentials described how to detect objects with them. In this section, we evaluate the benefits of using these potentials for object detection. As a baseline for our comparisons, we use the class-specific Hough Forests [9] and all evaluations are done according to the rules of the ‘competition 3’ of PASCAL VOC 2007 detection challenge [5].

The trainings of the forests are done using the ‘trainval’ set of images. Prior to the training, all object bounding boxes together with a 10% around them are cropped and rescaled to the smallest possible box with a maximum width or height of at most 100 pixels and a least minimum width or height of 50 pixels. The boxes annotated with ‘difficult’ tag are removed from the training set and for every category the images that do not contain any object of that category are used as the negative set. 15 trees are trained for each class. For training each tree, 200 positive objects and 200 background images are chosen randomly and from each

VOC 2007	Aeroplane	Bicycle	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Dinning Table	Dog	Horse	Motorbike	Person	Potted Plant	Sheep	Sofa	Train	TV/Monitor	Average (mAP)
Our Approach																					
HF baseline [9]	11.7	34.4	2.0	0.7	16.2	10.8	22.8	3.2	3.5	35.1	4.2	2.2	25.3	31.9	3.6	15.8	19.9	4.3	6.6	42.3	14.82
HF + sparsity	20.8	39.8	1.5	0.8	20.3	19.8	31.0	5.6	4.0	34.5	6.7	4.5	30.3	38.2	5.4	16.4	25.2	9.1	8.7	45.3	20.68
Competing Approaches																					
HT ISK [26]	24.6	32.1	5.0	9.7	9.2	23.3	29.1	11.3	9.1	10.9	8.1	13.0	31.8	29.5	16.6	6.1	7.3	11.8	22.6	21.9	16.65
LSVM [7]	29.0	54.6	0.6	13.4	26.2	39.4	46.4	16.1	16.3	16.5	24.5	5.0	43.6	37.8	35.0	8.8	17.3	21.6	34.0	39.0	26.25
VOC best [5]	26.2	40.9	9.8	9.4	21.4	39.3	43.2	24.0	12.8	14.0	9.8	16.2	33.5	37.5	22.1	12.0	17.5	14.7	33.4	28.9	23.33

Table 1: The comparison of our detection results on the PASCAL VOC 2007 dataset [5]. The first block compares the performance of our proposed approach with sparsity potentials (HF + sparsity) to the Hough Forest baseline (HF baseline [9]). As can be seen, by using the sparsity potentials the performance has been substantially improved for most categories. The comparison to the state-of-the-art approaches is shown in the second block. We outperform the best previously published Hough transform based approach (ISK [26]) in mAP. The other two rows give a comparison of our approach compared to the latent part model (LSVM [7]) and the best results of the PASCAL VOC Challenge (VOC best [5]).

of which 250 16 × 16 patches are extracted and represented with the 32 channels HoG-like features as in [9]. The training of each tree is continued until either the maximum depth of 20 is reached or less than 10 patches are left in a leaf.

The multi-scale detection on an image in the the 'test' set is done by building an image pyramid of 18 scales; starting from an up scaled image of 1.8 times the original image size and a scaling factor of $\frac{1}{\sqrt[4]{2}}$. For every detection image, 40 candidate objects are detected and their bounding boxes are estimated from backprojection [19]. Similar to [1], the non-maxima suppression is done by considering all features contributing more than 0.0005 to a detection as its support and removing the votes of all patches at its position from the scale pyramid.

For measuring the sparsity of a patch, the patches in a square neighborhood around it are considered 2(b). The parameters of the self-similarity classifiers are trained by first running the detector on cropped positive objects and all negative images in the validation set of a category. Then two histograms, one for positives and one for negatives, is created which records the sparsity values for every codebook entry. Finally, a classifier is trained to separate the positives from negatives based on this value.

Figure 3 evaluates the performance, in Precision/Recall, of using our proposed sparsity potentials for detecting objects on the PASCAL VOC 2007 dataset. The full comparison of the proposed sparsity potential with our baseline and other state-of-the-arts methods for all categories is given in the Table 1. As can be seen, the proposed sparsity potentials significantly improve the detection performance on this challenging dataset. The Fig. 2, shows the effect of this neighborhood size on the detection performance. For all of these experiments, the sparsity is calculated on a square neighborhood size of 71 × 71 pixels.

5 Conclusions

In this paper, we have introduced the sparsity potentials for object detection with the Hough transform. We have shown that the distinctiveness of the appearance of a patch in its neighborhood can be a strong cue for object detection. Further, we have proposed to measure this distinctiveness by the L1-norm of the self-similarity vector of a patch to its local neighborhood. Based on this measure and using the validation set, we have discriminatively trained a classifier for each codebook appearance and used it to separate foreground from background patches at test time. The proposed detector with sparsity potentials substantially outperforms the baseline detector and leads to a comparable accuracy to the state-of-the-art detectors on many categories of the PASCAL VOC 2007 dataset. In the future, it would be interesting to use this sparsity potentials in a multi-class setup to also discriminate classes from one another. Since the self-similar patches tend to belong to the same label, it would be also interesting to incorporate it as a higher order potential for image segmentation. Although we have used the simple L1-norm for sparsity, other popular measures (e.g. mutual information) can be used as well. Yet, efficient calculation of these measures remains a challenge.

Acknowledgements: This project has been funded in parts by Toyota Motor Corporation/Toyota Motor Europe and the European Commission projects RADHAR (FP7 ICT 248873) and IURO (FP7 ICT 248314).

References

[1] O. Barinova, V. Lempitsky, and P. Kohli. On detection of multiple object instances using hough transforms. In *CVPR*, 2010.

[2] O. Boiman, E. Shechtman, and M. Irani. In defense of nearest neighbor based image classification. In *CVPR*, 2008.

[3] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, volume 1, page 22, 2004.

[4] T. Deselaers and V. Ferrari. Global and efficient self-similarity for object classification and detection. In *CVPR*, 2010.

[5] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge 2007.

[6] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *CVPR*, 2005.

[7] P. F. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *TPAMI*, 32(9):1627 – 1645, 2009.

[8] B. Fulkerson, A. Vedaldi, and S. Soatto. Class segmentation and object localization with superpixel neighborhoods. In *ICCV*, 2009.

[9] J. J. Gall, A. Yao, N. Razavi, L. Van Gool, and V. Lempitsky. Hough forests for object detection, tracking, and action recognition. *TPAMI*, 33(11):2188–2202, 2011.

[10] P. V. Gehler and S. Nowozin. On feature combination for multiclass object detection. In *ICCV*, 2009.

[11] S. Goferman, L. Zelnik-Manor, and A. Tal. Context-aware saliency detection. In *CVPR*, 2010.

[12] X. Hou and L. Zhang. Saliency detection: A spectral residual approach. In *CVPR*, 2007.

[13] T. Kadir and M. Brady. Saliency, scale and image description. *IJCV*, 45(2):83–105, 2001.

[14] B. Leibe, A. Leonardis, and B. Schiele. Robust object detection with interleaved categorization and segmentation. *IJCV*, 77(1-3):259–289, 2008.

[15] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[16] S. Maji and J. Malik. Object detection using a max-margin hough transform. In *CVPR*, 2009.

[17] K. Mikolajczyk and C. Schmid. Scale & affine invariant interest point detectors. *IJCV*, 60(1):63–86, 2004.

[18] T. Minka. The summation hack as an outlier model. http://research.microsoft.com/~minka/papers, 2003.

[19] N. Razavi, J. Gall, and L. Van Gool. Backprojection revisited: Scalable multi-view object detection and similarity metrics for detections. In *ECCV*, 2010.

[20] E. Shechtman and M. Irani. Matching local self-similarities across images and videos. In *CVPR*, 2007.

[21] J. Shotton, M. Johnson, and R. Cipolla. Semantic texton forests for image categorization and segmentation. In *CVPR*, 2008.

[22] J. Sivic, B.C. Russell, A.A. Efros, A. Zisserman, and W.T. Freeman. Discovering objects and their location in images. In *ICCV*, 2005.

[23] R. Valenti, N. Sebe, and T. Gevers. Image saliency by isocentric curvedness and color. In *ICCV*, 2009.

[24] O.J. Woodford, M.T. Pham, A. Maki, F. Perbet, and B. Stenger. Demisting the hough transform. In *BMVC*, 2011.

[25] P. Yarlagadda, A. Monroy, and B. Ommer. Voting by grouping dependent parts. In *ECCV*, 2010.

[26] Y. Zhang and T. Chen. Implicit shape kernel for discriminative learning of the hough transform detector. In *BMVC*, 2010.