Efficient Rotation Invariant Object Detection using Boosted Random Ferns

Michael Villamizar, Francesc Moreno-Noguer, Juan Andrade-Cetto, Alberto Sanfeliu
Institut de Robòtica i Informàtica Industrial, CSIC-UPC
Llorens i Artigas 4-6, 08028. Barcelona, Spain
{mvillami,fmoreno,cetto,sanfeliu}@iri.upc.edu

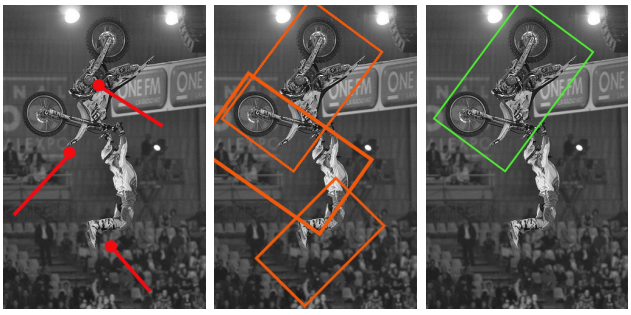
Abstract

We present a new approach for building an efficient and robust classifier for the two class problem, that localizes objects that may appear in the image under different orientations. In contrast to other works that address this problem using multiple classifiers, each one specialized for a specific orientation, we propose a simple two-step approach with an estimation stage and a classification stage. The estimator yields an initial set of potential object poses that are then validated by the classifier. This methodology allows reducing the time complexity of the algorithm while classification results remain high.

The classifier we use in both stages is based on a boosted combination of Random Ferns over local histograms of oriented gradients (HOGs), which we compute during a preprocessing step. Both the use of supervised learning and working on the gradient space makes our approach robust while being efficient at run-time. We show these properties by thorough testing on standard databases and on a new database made of motorbikes under planar rotations, and with challenging conditions such as cluttered backgrounds, changing illumination conditions and partial occlusions.

1. Introduction

We present a novel approach for detecting objects of a specific category that may appear in images under different planar rotations, as shown in Figure 1. This problem has been traditionally addressed from a multi-class perspective by using classifiers specifically trained at different orientations [8]. These methods however, suffer from two limitations. First, the computational cost for both the training and test stages increases with the number of classifiers, and second, the use of multiple classifiers increases the number of false positives.



(a) (b) (c)

Figure 1. The proposed two-step rotation invariant object detection approach. (a) First, the estimator yields an initial set of potential object poses (location, scale and orientation). (b) Each hypothesis is steered back to a canonical orientation and validated by the classifier. (c) The hypotheses that remain after non-maxima suppression are considered object instances.

In this paper, we introduce an approach that overcomes both these limitations. We achieve this by splitting the object detection task in two different steps: pose estimation and classification. We initially train an estimator using images under all orientations. This estimator can be very efficiently computed over the whole image, yielding a large number of candidates, many of them false positives, along with estimates of their location, orientation and scale. The second step learns an orientation-specific classifier that, by means of a simple steering procedure, can be efficiently tested on each hypothesis according to its estimated pose. Results of this detection are shown in Figure 1.

In addition, and in contrast to other works that use codebook appearance techniques [5, 9, 13, 18], our method is based on Random Ferns [17], densely computed over local HOGs. These Random Ferns are probabilistically computed using a boosting algorithm, and as shown in the results Section, allow for the computation of robust features in a very simple and efficient manner.

The rest of paper is organized as follows. Section 2, places our contribution in context with related work. In Sec-

This work has been partially funded by the Spanish Ministry of Science and Innovation under projects UbROB DPI2007-61452, PAU DPI2008-06022, and MIPRCV Consolider Ingenio 2010 CSD2007-00018; and by the EU project GARNICS FP7-247947. The first author is funded by the Technical University of Catalonia (UPC).

tion 3 we describe all the elements of our approach, including the computation of the binary features over HOGs, the Random Ferns, and the object pose estimator and classifier. In Section 4 we present the results over several datasets, and we discuss some implementation and computational cost details in Section 5.

2. Related Work and Contributions

The problem of detecting object categories in images is known to be very challenging and needs to address several issues such as large intra-class object variations, changes in object pose, cluttered backgrounds or illumination changes.

Yet, many recent methods have shown a remarkable success when are used in conjunction with machine learning techniques such as boosting[10, 14, 22, 26, 28] or Support Vector Machines (SVMs) [2, 4, 9, 15, 18]. However, these methods have been effectively used mostly for standard datasets [1, 3, 5, 11] for which the objects only appear in a relatively reduced number of poses [7, 20, 24].

In this paper, we are interested in object categorization under general in-plane rotations. We show that the problem can be solved by splitting it in two stages. A pose estimation step, followed by a classification step. In fact this is similar to what was done in [18] for detecting cars under general 3D poses, although estimator they use requires from several and relatively complex steps.

The simplest strategy for dealing with in-plane rotations would be to rotate the image or steer the object classifier to multiple orientations. However, this approach would have a high computational cost, because it would require to test the classifier several times over the image, one for each discretized orientation. In addition, it would produce a large number of false positives because the classifier would have to be evaluated significantly more times.

In [13] objects at different orientations are detected by means of an Implicit Shape Model and rotation invariant features. Yet, the method is computationally expensive, as it requires to compute SIFT descriptor over edges, and apply a PCA analysis followed by a voting strategy. Other approaches address the problem as a multi-class one using different boosting versions [8, 25]. However, since they decompose the problem into several classes it requires from more features and a higher computational effort. In [25], features are shared among classes also through a hierarchical structure, also requiring an expensive learning step.

In this paper we show that decoupling the orientation estimation from object classification allows to reduce the computation time both for learning and testing, while the detection results remain high. The estimation step is used as a pre-filter that generates object hypotheses. Given these hypotheses an orientation-specific classifier is appropriately steered and verifyied according to the estimated orientation

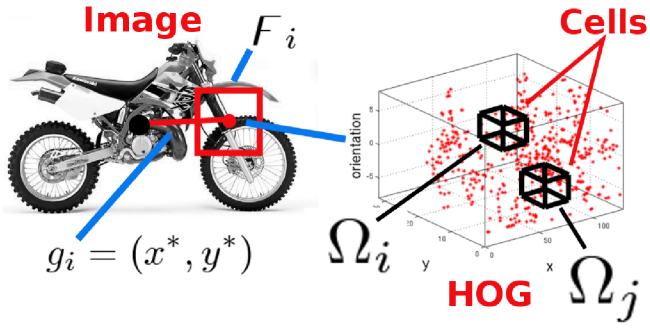


Figure 2. Local Binary Feature. Our features are computed from binary comparisons between different bins of the HOG.

in a similar way as was done in [19] for all the pixels in the image.

An additional contribution of our approach is the use Fern-based [17] binary features over local histograms of oriented gradients, exploiting their simplicity, rapid computation, and the robustness they offer to illumination changes. Local binary features have been traditionally computed in the intensity domain, for specific object detection and segmentation tasks [16, 17, 23]. And finally, another contribution is the use of a boosting step to learn the most discriminative set of Ferns. In contrast to the original work [17] where these features were randomly chosen, we incorporate a Real Adaboost algorithm [21] to select the most meaningful Ferns according to their classification power over training samples.

3. Two-Step Orientation Invariant Object Detection

In this section we describe each of the individual ingredients of our two-step approach. Section 3.1 explains how local binary features over HOGs are computed. In Section 3.2 we describe the Random Ferns using the likelihood-ratio between classes. The object pose estimator and classifier are presented in Sections 3.3 and 3.4.

3.1. HOG-based Features

A Local Binary Feature (LBF) maps an image sample x to a boolean space in the form,

$$f : x \rightarrow \{0, 1\}, \quad x \in X, \tag{1}$$

by simple comparison between a pair of image values (e.g pixel intensities). Traditionally, LBFs are computed in the image intensity domain yielding successful detection results for specific objects. We extend the same idea and propose to compute LBFs in the HOG domain instead, since HOG-based features have demonstrated remarkable results for object categorization showing robustness to illumination and

object appearance changes. Therefore, we define the HOG-LBF as a signed comparison between two HOG cells,

$$f(x) = \begin{cases} 1 & x_{\Omega_i} > x_{\Omega_j} \\ 0 & x_{\Omega_i} \leq x_{\Omega_j} \end{cases}, \quad \Omega \in \mathbb{R}3 \tag{2}$$

where Ω_i and Ω_j are the feature component locations defined by spatial and orientation bin coordinates (u, v, θ). Figure 2 shows one LBF instance for a local HOG.

3.2. Random Ferns on the HOG Space

In order to compute object features, we use the Random Ferns proposed in [17] for keypoint classification. However, and in contrast to this original formulation of the Random Ferns, we write the Ferns expression in terms of likelihood ratios between classes. This allows us to seek for the feature combinations that maximize this ratio, by means of a boosting algorithm.

Our goal is to model the posterior object class probability given a set of n features (LBF). This can be expressed by the Bayes rule as,

$$P(C|f_1, f_2, ..f_n) = \frac{P(f_1, f_2, ..f_n|C)P(C)}{P(f_1, f_2, ..f_n)}, \tag{3}$$

where C refers to the category and f_i is a feature. An equivalent expression may be written for the background (B) class. We seek to maximize the object class posterior probability ratio with respect to the background class. By removing the priors $P(f_1, f_2, ..f_n)$, common for all the classes, assuming uniform prior probabilities $P(C) = P(B)$, and considering logarithms, the ratio of probabilities may be written as,

$$\log \frac{P(C|f_1, f_2, ..f_n)}{P(B|f_1, f_2, ..f_n)} = \log \frac{P(f_1, f_2, ..f_n|C)}{P(f_1, f_2, ..f_n|B)}. \tag{4}$$

Since computing the complete joint probability for a large feature set is not feasible, we split the previous equation into m subsets ($F = \{f_1, f_2, ..f_r\}$), with $r = n/m$. These feature subsets are known as Ferns, and assuming they are independent, their joint log-probability is computed as,

$$\log \frac{\prod_{i=1}^{m} P(F_i|C)}{\prod_{i=1}^{m} P(F_i|B)} = \sum_{i=1}^{m} \log \frac{P(F_i|C)}{P(F_i|B)}, \tag{5}$$

Each Fern captures the co-occurrence of r binary features computed on the HOG space, and encodes object local appearances. Its response is represented by a combination of boolean outputs. For instance, the observation z_i of a Fern F_i made of $r = 3$ features with binary outputs $0, 1, 1$, would be $(011)_2 = 3$. In other words, each Fern maps 2D image samples to a $K = 2^r$-dimensional space,

$$F : x \rightarrow z, \quad x \in X, \quad z \in \mathbb{R}. \tag{6}$$

Then, the probability of each Fern F_i may be written using its feature set observation z_i conditioned to each class,

$$\sum_{i=1}^{m} \log \frac{P(z_i = k|C, g_i)}{P(z_i = k|B, g_i)}, \quad k = 1, 2, ...K, \tag{7}$$

where k corresponds to the observation index and g_i ($g \in \mathbb{R}2$) to image spatial location where the Fern F_i is evaluated, measured from the object image center (Figure 2).

3.3. Object Pose Estimator

We build a robust object pose estimator as a linear combination of weak classifiers, where each of them is based on a Random Fern with an associated spatial image location. More formally, we want to build an object estimator classifier $E(x)$, yielding the most discriminative Ferns F_i and locations g_i, that is, the Ferns and locations that maximize Eq. 7. This is achieved by means of a Real Adaboost algorithm [21], that iteratively assembles weak classifiers and adapts their weighting values. Then, the estimator is defined as a sum of T weak classifiers,

$$E(x) = \sum_{t=1}^{T} h_t(x) > \beta_e, \tag{8}$$

where $h_t(x)$ is a weak classifier and β_e is the estimator threshold whose default value is zero. In practice, when computing the pose estimator, each weak classifier incorporates an additional orientation parameter w that is a label assigned to each training image sample indicating the object orientation that has been applied by rotating training data to L in-plane rotations. Therefore, a weak classifier is defined by the co-occurrence of a Random Fern observation z and an image orientation label w,

$$h_t(x) = \frac{1}{2} \log \left(\frac{P(z_t, w|C, g_t) + \epsilon}{P(z_t, w|B, g_t) + \epsilon} \right), \tag{9}$$

where ϵ is a smoothing factor.

The estimator seeks to maximize this co-occurrence during the learning step by evaluating different Random Ferns and keeping the most discriminative ones. This is done at each iteration t of the boosting step by calling a weak learner to compute and select the most discriminative classifier according to a sample weight distribution $D_t(i)$,

$$P(z_t = k, w = l|C, g_t) = \sum_{i:z_t(x_i)=k \wedge w(x_i)=l} D_t(i), \tag{10}$$

being $l = 1, 2, ..L$ and $k = 1, 2, ..K$.

At each iteration, the weak classifier that maximized the classification power in terms of the following Bhattacharyya distance is selected

$$Q_t = 2 \sum_{l=1}^{L} \sum_{k=1}^{K} \sqrt{P(z_t = k, w = l|C, g_t) P(z_t = k, w = l|B, g_t)}. \tag{11}$$

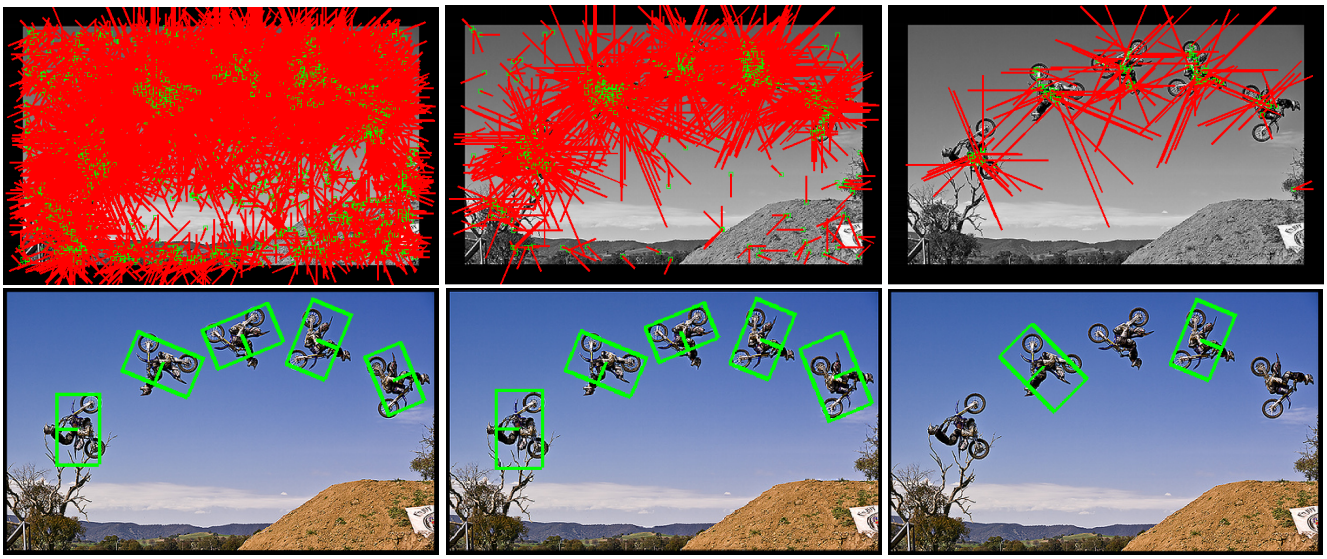


Figure 3. Object estimation and classification. First row: Object hypotheses. Second row: Classification results. Each column corresponds to a different value of the parameter $\beta_e = \{0, 2, 4\}$.

Algorithm 1 Object Orientation Estimator

1: Given a number of weak classifiers T and a dataset S consisting of N image samples labeled $(x_1, y_1, w_1)...(x_n, y_n, w_n)$, where $y_i \in \{+1, -1\}$ is the label for category and background classes, respectively; and $w_i = \{w_1, w_2, ..., w_L\}$ the orientation label.
2: Construct a pool of M Random Ferns densely computed over the whole image.
3: Initialize sample weights $D_1(i) = \frac{1}{N}$.
4: **for** $t = 1$ to T **do**
5: **for** $m = 1$ to M **do**
6: Under current distribution $D_t(i)$, calculate $h_m(x)$ and its Bhattacharyya distance Q_m.
7: **end for**
8: Select the h_t that minimizes Q_m.
9: Update sample weights.
 $D_{t+1}(i) = \frac{D_t(i) \exp[-y_i h_t(x_i)]}{\sum_{i=1}^{N} D_t(i) \exp[-y_i h_t(x_i)]}$
10: **end for**
11: Final strong classifier.
 $E(x) = sign\left(\sum_{t=1}^{T} h_t(x) - \beta_e\right)$

The weak classifiers built using this methodology are focused on Random Ferns that are both discriminative for their observations and for their orientation distributions. Thus, if one weak classifier tends to favor some orientations, subsequent classifiers are forced to classify those samples labelled as missclassified orientations. Details on this methodology for computing the estimator are given in the pseudocode of Algorithm 1.

Orientation Estimation

In order to compute the object orientation at runtime, the estimator is evaluated according to the following expression

$$E(x) = \frac{T}{2} \sum_{t=1}^{T} \log \frac{P(z_t|C, g_t)}{P(z_t|B, g_t)} + \frac{T}{2} \sum_{t=1}^{T} \log \frac{P(w|C, g_t, z_t)}{P(w|B, g_t, z_t)}. \quad (12)$$

The left-hand side of this equation, is the *root classifier* Φ and corresponds to the ratio of observation probability of the T selected Random Ferns. Note that it does not consider the orientation parameter w, and hence, this classifier responds to object instances under multiple in-plane rotations. By setting a threshold $\Phi > \beta_e$, we can choose a large number of potential hypotheses at runtime. The right-hand side of the Eq. 12, is the *orientation estimation* term, which is made by the combination of local orientation estimations given by the observations of the Boosted Random Ferns. According to this distribution, the object orientation is assigned to

$$\phi = \arg\max_k \sum_{t=1}^{T} \log \frac{P(w = k|C, g_t, z_t)}{P(w = k|B, g_t, z_t)}. \quad (13)$$

The method is exemplified in Figure 3. The upper row shows the initial object hypotheses for different values of the root detector threshold β_e. The object pose (location, scale and orientation) is represented by means of red lines. The images in the second row show the object detection results after testing the steered object classifier over the initial hypotheses. The β_e parameter controls the number of false positives of the estimator and, consequently, the computational cost of the algorithm, since the classification step is

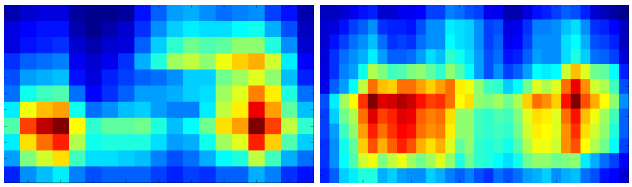


Figure 4. Boosted Random Ferns. Spatial locations of boosted Random Ferns for motorbike and car categories, respectively.

only evaluated over the initial hypotheses. Therefore, the choice of this parameter is a tradeoff between false positives and computational burden.

3.4. Object Classifier

The orientation-specific classifier is built in the same way as the estimator $E(x)$ but using training images oriented to a canonical orientation. Furthermore, at runtime, the classifier may be steered to each specific orientation given by the estimator, which prevents from having to train a different classifier for each possible orientation. We can then write the orientation-specific classifier based on Random Ferns as,

$$H(x) = \sum_{t=1}^{T} h_t(x) > \beta_c, \tag{14}$$

where β_c is the classifier threshold and h_t is a weak classifier defined by

$$h_t(x) = \frac{1}{2} \log \left(\frac{P(z_t = k|C, g_t) + \epsilon}{P(z_t = k|B, g_t) + \epsilon} \right). \tag{15}$$

At iteration t, the probability $P(z_t|C, g_t)$ is computed under the distribution of sample weights $D_t(i)$ by

$$P(z_t = k|C, g_t) = \sum_{i:z_t(x_i)=k} D_t(i), k = 1, .., K. \tag{16}$$

Following the same idea as for the estimator, we choose the weak classifier that minimizes the following Bhattacharyya distance

$$Q_t = 2 \sum_{k=1}^{K} \sqrt{P(z_t = k|C, g_t) P(z_t = k|B, g_t)}. \tag{17}$$

Figure 4 shows how the boosting step extracts discriminative Random Ferns for a given class (i.e., motorbikes). For this case, features occur mainly in semantic object parts like wheel and handlebars.

Steering the Object Classifier

For each object hypothesis made by the estimator $E(x)$, the classifier described above is steered and evaluated. This is performed by simply rotating the coordinates of each Local Binary Feature Ω in the HOG as follows,

$$\Omega^* = \begin{bmatrix} \cos(\phi) & -\sin(\phi) & 0 \\ \sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & (1+p) \end{bmatrix} \Omega, \tag{18}$$

where $\Omega = [u, v, \theta]'$, ϕ is the rotation angle and p is the angular translation increment defined by $\frac{\phi * L}{\pi \theta}$.

Method	UIUC Multi-scale	UIUC Single scale	TUD motorbikes
[1]	39.6%	76.5%	-
[5]	-	88.5%	-
[6]	87.8%	88.6%	-
[15]	90.6%	99.9%	-
[22]	-	92.8%	-
[13]	94.7%	-	89.0%
[11]	95.0%	97.5%	87.0%
[9]	98.6%	98.5%	-
[12]	-	-	92.8%
*	**98.5%**	**98.2%**	**89.3%**

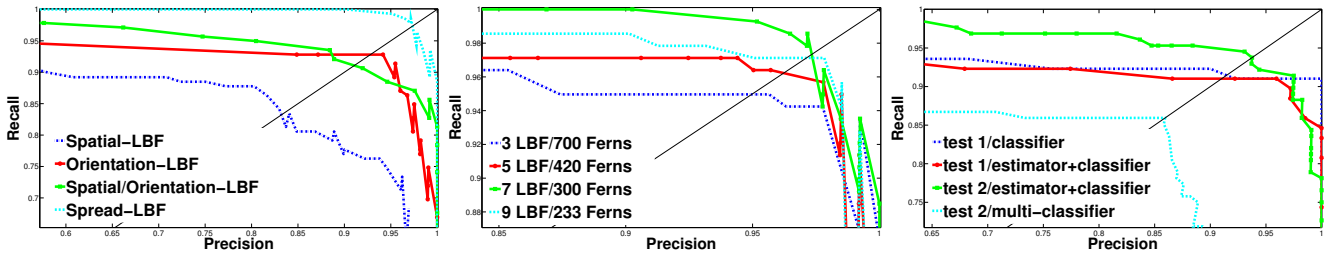
Table 1. Category detection rates for public datasets.

4. Experiments

The proposed approach has been extensively validated using different datasets. To test the orientation-specific classifier and to compare its performance to other state-of-the-art approaches, we initially evaluated it with standard datasets without explicit in-plane rotations. For this purpose, we used the well-known *UIUC car* dataset [1] and the *TUD motorbike* dataset [6]. We also created a new dataset containing motorbikes under planar rotations, which allowed us to test the combined orientation estimation and classification approach. In the following, we will denote this dataset as *Freestyle Motocross*.

UIUC Car dataset - This dataset contains car-sides under difficult imaging conditions such as illumination changes, cluttered backgrounds and mild occlusions. This dataset has two sets of images for testing. The first one has 170 images containing 200 car instances with similar scale to that of the training samples (40x100 pixels). The second one has 108 images consisting of 139 cars at different scales, varying from 36x89 to 85x212. Unlike the method presented in [22], the proposed work does not need to test each image twice, and the detector is able to simultaneously detect cars facing to the left or to the right. The best achieved detection rates for this dataset are 98.2% and 98.5% Equal Error Rate (EER) for single and multi-scale tests, respectively. The top two rows of Figure 8 show some samples of the detection results.

TUD Motorbike Dataset - This dataset consists of 115 images containing 125 motorbike instances under occlusions and different scales. For training, 400 motorbike images



(a) (b) (c)

Figure 5. Detection performances. (a) Detection curves over UIUC car dataset for different LBP configurations. (b) Feature co-occurrence evaluation over UIUC car dataset. (c) Detection performances for Freestyle Motocross dataset assessing different detector approaches.

from the Caltech motorbike dataset [5] have been used. The achieved detection rate for this dataset is of 89.3% EER. Table 1 shows some detection performances for this dataset and in the middle row of Figure 8 we plot some detection results.

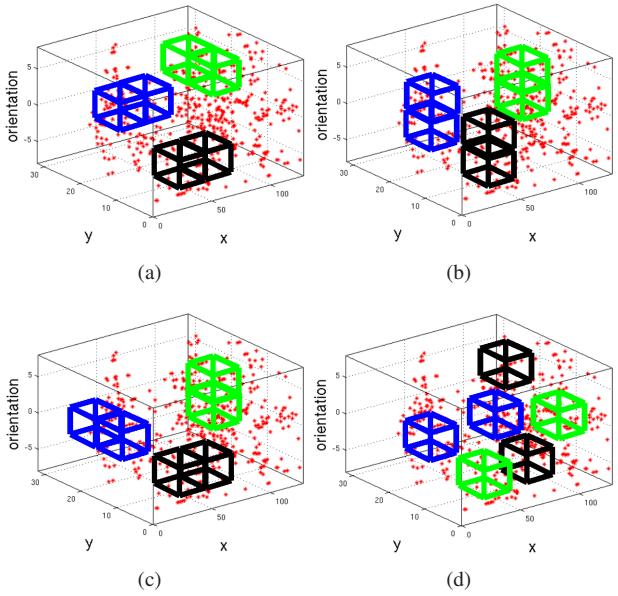
Freestyle Motocross - This dataset has been built in order to explicitly evaluate the proposed algorithm to rotations in the plane. The images were extracted from the internet and correspond to motorbikes with challenging conditions such as extreme illumination, multiple scales and partial occlusion. Moreover, some instances show some degree of out-of-plane rotations (see the two bottom rows of Figure 8). There are two sets of images for testing. The first set has 69 images with 78 motorbikes without in-plane rotations while the second one has 100 images with 128 motorbikes instances with multiple rotations in the plane. The learning was done using 800 images from the Caltech motorbike dataset [5].

Two validate the orientation estimation-classification method, two types of experiments were performed. In the first experiment, two detection approaches are considered; one that only uses the object classifier and another where the classifier is tested in combination with the estimator. Both methods show a similar detection performance achieving a detection rate of 91.03% EER. In the second experiment, two detection approaches are considered again. The first one uses the estimator/classifier combination and the second one tests the classifier to multiple orientations. The detection rate for combining estimator and classifier is 93.75% EER while for detection under multiple rotations is 85.94%.

4.1. Discussion

The proposed approach achieves remarkable results in comparison to state-of-the-art methods (see Table 1), with the advantage of being more efficient computationally, since it does not require complex feature computation or the combination of multiple cues [12].

To show how the layout of Random Fern features inside a local HOG affects the detection performance, four different types of LBF configurations have been evaluated. The first



(a) (b) (c) (d)

Figure 6. HOG-based LBF configurations. (a) Spatial-LBF (b) Orientation-LBF (c) Spatial/Orientation-LBF (d) Spread-LBF.

three have in common that feature comparisons are carried out between adjacent HOG cells in spatial and orientation directions, and in a combination of both. These LBF configurations resemble to Haar-like features computed in the HOG domain [27]. Finally, a spread configuration is proposed in which features are distributed over the whole local HOG (Figure 6). Detection performances for the UIUC dataset are shown in Figure 5(a). Spread-LBF outperforms others because this type of configuration does not constraint feature locations. Therefore, in the rest of paper, spread features are chosen to construct the object estimator and classifier.

Similarly, the number of features (LBF) per Fern has been evaluated to measure the importance of feature co-occurrence. Four orientation-specific classifiers have been learned using the same number of LBF (2100) but with different number of features per Fern. The results are depicted in Figure 5(b). It shows that feature co-occurrence improves detection performance until a saturation point where there

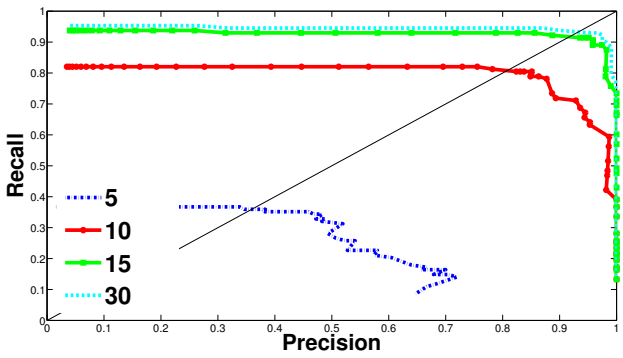


Figure 7. Orientation accuracy. Detection performances for different values of orientation estimation.

are many features for such local HOG size.

With regard to the estimator evaluation, the test 1 in the Freestyle Motocross dataset shows that incorporating the estimator does not affect the detection performance for object instances without rotations in the plane (see Figure 5c). It is because even when many object candidates are given by the estimator, the classification stage is still able to reject false hypotheses. For test 2, the combination of estimator and classifier shows better results than the classifier tested to multiple orientations since the latter has to be evaluated N times, being N the number of orientations. Hypotheses verification at multiple orientation increases the number of false positives and the computational cost.

Another experiment to measure orientation estimation accuracy is shown in Figure 7. For this experiment, a true positive detection is considered when the difference between the estimated orientation and ground truth orientation is below a given accuracy value. The figure shows different detection curves for different accuracy values. The proposed method provides good detection results, above 91% EER, for an error margin of 15 degrees. The accuracy of the orientation estimation could be improved if we consider more object orientations in the learning step. However, this is at the expense of increased computational cost and training time. For this experiment, we used 16 orientations.

5. Implementation Details

For computing gradients, Prewitt masks have been selected and their signs omitted to have unsigned gradients (0° - 180°). Since filter response is affected by the scale of image, just as image derivatives are, a HOG pyramid is built where in each level an integral histogram is computed in order to have an efficient estimator/classifier testbed. In this work, two scale levels per octave are used, being a good tradeoff between computational cost and detection performance [10]. For HOG computation, the cell size and local HOG size (block size) are set to $3x3$ pixels and $3x3$ cells, respectively. The number of gradient orientation bins for the classifier is set to 4 while for the estimator is set to 8. For orientation computation, the training image data is artificially rotated to 16 orientations. In all experiments, about 2000 spread LBF have been used for computing the category classifiers.

The most computationally expensive part of the algorithm is the convolution of the boosted Random Ferns over the whole image. That is, $O(N \times M \times P \times S)$ where N, M is the image size, and P and S are the number of Random Ferns and features per Fern, respectively. The decoupled approach we propose allows to initially evaluate the estimator and just consider the classifier when the estimator score exceeds a certain parameter β_e. In the worst case, the detector cost will equal the cost of applying both the estimator and classifier sequentially. This cost is of course lower than testing multiple independent classifiers at different orientations.

6. Conclusions

The presented work addresses the robust detection of specific categories that may appear in images under different rotations in the plane. The proposed approach decouples the problem in two stages, pose estimation and classification, that allow to detect objects in images efficiently. Computation of the pose estimator and the orientation-specific classifier are based on the boosted combination of Random Ferns which are evaluated densely over local HOGs.

The method has been validated in standard datasets containing category instances under challenging imaging conditions. Our detection results show the method to be competitive with state-of-the-art methods, with the advantage of being computationally more efficient.

References

[1] S. Agarwal and D. Roth. Learning a sparse representation for object detection. In *ECCV*, 2002.

[2] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[3] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge 2007 (voc2007) results. www.pascalnetwork.org/challenges/voc/voc2007/workshop.

[4] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR*, 2008.

[5] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR*, 2003.

[6] M. Fritz, B. Leibe, B. Caputo, and B. Schiele. Integrating representative and discriminant models for object category detection. In *ICCV*, 2005.

[7] D. Hoiem, C. Rother, and J. Winn. 3d layoutcrf for multiview object class recognition and segmentation. In *CVPR*, 2007.

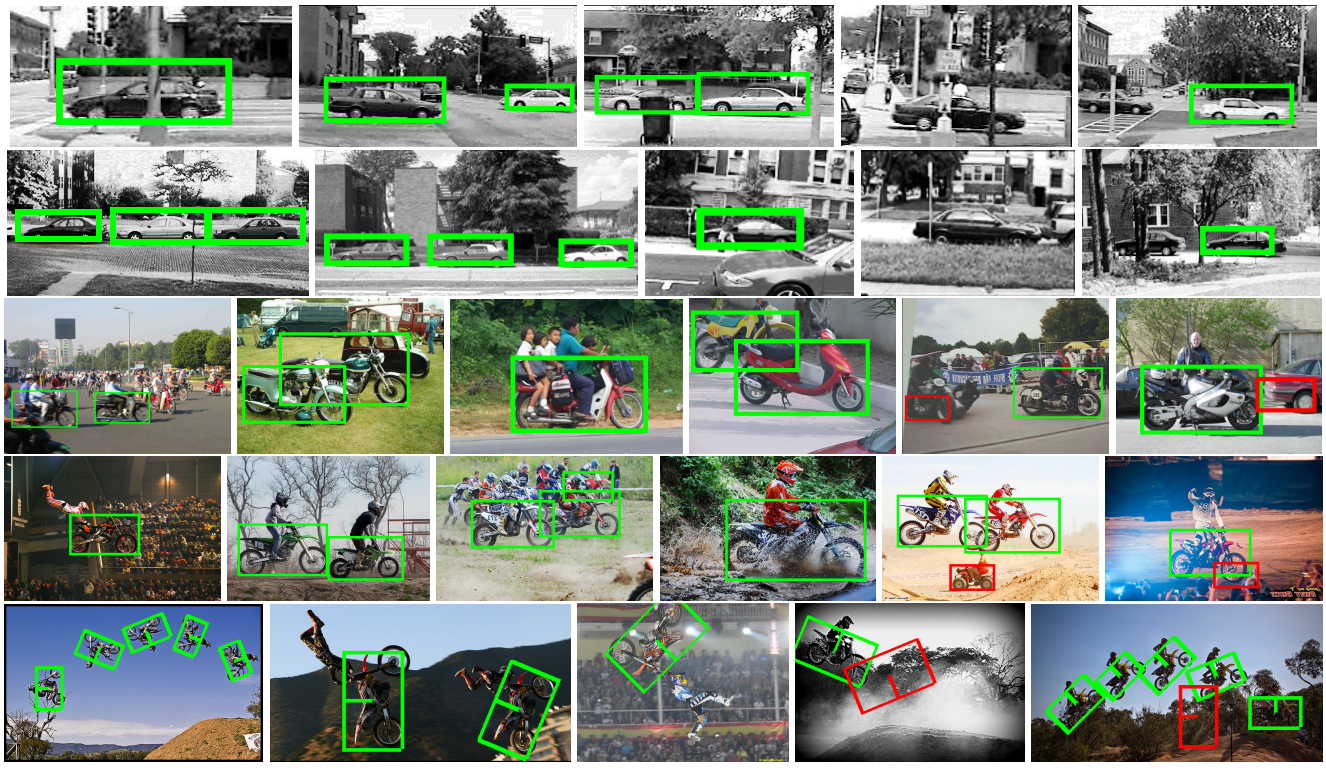


Figure 8. Detection results for different datasets. First row: UIUC multi-scale. Second row: UIUC single scale. Third row: TUD motorbikes. Fourth row: Freestyle motocross (Test 1). Fifth row: Freestyle motocross with in-plane rotations (Test 2).

[8] C. Huang, H. Ai, Y. Li, and S. Lao. Vector boosting for rotation invariant multi-view face detection. In *ICCV*, 2005.

[9] C. Lampert, M. Blaschko, and T. Hofmann. Beyond sliding windows: Object localization by efficient subwindow search. In *CVPR*, 2008.

[10] I. Laptev. Improving object detection using boosted histograms. *IVCJ*, 2009.

[11] B. Leibe, A. Leonardis, and B. Schiele. Robust object detection with interleaved categorization and segmentation. *IJCV*, 2008.

[12] B. Leibe, K. Mikolajczyk, and B. Schiele. Segmentation based multi-cue integration for object detection. In *BMCV*, 2006.

[13] K. Mikolajczyk, B. Leibe, and B. Schiele. Multiple object class detection with a generative model. In *CVPR*, 2006.

[14] T. Mita, T. Kaneko, B. Stenger, and O. Hori. Discriminative feature co-occurrence selection for object detection. *PAMI*, 2008.

[15] J. Mutch and D. Lowe. Multiclass object recognition with sparse, localized features. In *CVPR*, 2006.

[16] T. Ojala and M. Pietikainen. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *PAMI*, 2002.

[17] M. Ozuysal, P. Fua, and V. Lepetit. Fast keypoint recognition in ten lines of code. In *CVPR*, 2007.

[18] M. Ozuysal, V. Lepetit, and P. Fua. Pose estimation for category specific multiview object localization. In *CVPR*, 2008.

[19] H. A. Rowley, S. Baluja, and T. Kanade. Rotation invariant neural network-based face detection. In *CVPR*, 1998.

[20] S. Savarese and L. Fei-Fei. 3d generic object categorization, localization and pose estimation. In *ICCV*, 2007.

[21] R. E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. Machine Learning, 1999.

[22] J. Shotton, A. Blake, and R. Cipolla. Contour-based learning for object detection. In *ICCV*, 2005.

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

[24] A. Thomas, V. Ferrari, B. Leibe, T. Tuytelaars, B. Schiele, and L. V. Gool. Towards multi-view object class detection. In *CVPR*, 2006.

[25] A. Torralba, K. P. Murphy, and W. T. Freeman. Sharing visual features for multiclass and multiview object detection. *PAMI*, 2007.

[26] M. Villamizar, A. Sanfeliu, and J. Andrade-Cetto. Computation of rotation local invariant features using the integral image for real time object detection. In *ICPR*, 2006.

[27] W. Zhang, J. Sun, and X. Tang. Cat head detection - how to effectively exploit shape and texture features. In *ECCV*, 2008.

[28] Q. Zhu, S. Avidan, M. Ye, and K.-T. Cheng. Fast human detection using a cascade of histograms of oriented gradients. In *CVPR*, 2006.