Learning a Fast Emulator of a Binary Decision Process

Jan Šochman and Jiří Matas

Center for Machine Perception, Dept. of Cybernetics, Faculty of Elec. Eng.
Czech Technical University in Prague, Karlovo nám. 13, 121 35 Prague, Czech Rep.
{sochmj1,matas}@cmp.felk.cvut.cz

Abstract. Computation time is an important performance characteristic of computer vision algorithms. This paper shows how existing (slow) binary-valued decision algorithms can be approximated by a trained WaldBoost classifier, which minimises the decision time while guaranteeing predefined approximation precision. The core idea is to take an existing algorithm as a black box performing some useful binary decision task and to train the WaldBoost classifier as its emulator.

Two interest point detectors, Hessian-Laplace and Kadir-Brady saliency detector, are emulated to demonstrate the approach. The experiments show similar repeatability and matching score of the original and emulated algorithms while achieving a 70-fold speed-up for Kadir-Brady detector.

1 Introduction

Computation time is an important performance characteristic of computer vision algorithms. We show how existing (slow) binary-valued classifiers (detectors) can be approximated by a trained WaldBoost detector [1], which minimises the decision time while guaranteeing predefined approximation precision. The main idea is to look at an existing algorithm as a black box performing some useful binary decision task and to train a *sequential classifier* to emulate its behaviour.

We show how two interest point detectors, Hessian-Laplace [2] and Kadir-Brady [3] saliency detector, can be emulated by a sequential WaldBoost classifier [1]. However, the approach is very general and is applicable in other areas as well (e.g. texture analysis, edge detection).

The main advantage of the approach is that instead of spending man-months on optimising and finding a fast and still precise enough approximation to the original algorithm (which can be sometimes very difficult for humans), the main effort is put into finding a suitable set of features which are then automatically combined into a WaldBoost ensemble. Another motivation could be an automatic speedup of a slow implementation of one's own detector.

A classical approach to optimisation of time-to-decision is to speed-up an already working approach. This includes heuristic code optimisations (e.g. FastSIFT [4] or SURF [5]) but also very profound change of architecture (e.g. classifier cascade [6]). A less common way is to formalise the problem and try to solve the error/time trade-off in a single optimisation task.

Y. Yagi et al. (Eds.): ACCV 2007, Part II, LNCS 4844, pp. 236–245, 2007.

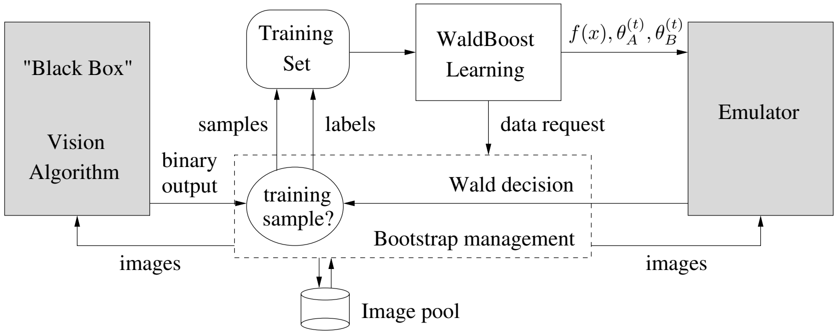


Fig. 1. The proposed learning scheme

Our contribution is a proposal of a general framework for speeding up existing algorithms by a sequential classifier learned by the WaldBoost algorithm. Two examples of interest point detectors were selected to demonstrate the approach. The experiments show a significant speed-up of the emulated algorithms while achieving comparable detection characteristics.

There has been much work on the interest point detection problem [7] but to our knowledge, learning techniques has been applied only to subproblems but not to the interest point detection as a whole. Lepetit and Fua [8] treated matching of detected points of interest as a classification problem, learning the descriptor. Rosten and Drummond [9] used learning techniques to find parameters of a hand-designed tree-based Harris corner classifier. Their motivation was to speed-up the detection process, but the approach is limited to the Harris corner detection. Martin et al. [10] learned a classifier for edge detection, but without considering the decision time and with significant manual tuning. Nevertheless, they tested a number of classifier types and concluded that a boosted classifier was comparable in performance to these classifiers and was preferable for its low model complexity and low computational cost.

The rest of the paper is structured as follows. The approximation of a black-box binary valued decision algorithm by a WaldBoost classifier is discussed in §2. Application of the approach to interest point detectors is described in §3. Experiments are given in §4 and the paper is concluded in §5.

2 Emulating a Binary-Valued Black Box Algorithm with WaldBoost

The structure of the approach is shown in Figure 1. The black box algorithm is any binary-valued decision algorithm. Its positive and negative outputs form a labelled training set. The WaldBoost learning algorithm builds a classifier sequentially and when new training samples are needed, it bootstraps the training set by running the black box algorithm on new images. Only the samples not decided yet by the so far trained classifier are used for training. The result of the process is a WaldBoost sequential classifier which emulates the original black box algorithm.

The bootstrapping loop uses the fact that the black box algorithm can provide practically unlimited number of training data. This is in contrast to commonly used human labelled data which are difficult to obtain.

Next, a brief overview of the WaldBoost learning algorithm is presented.

2.1 WaldBoost

WaldBoost [1] is a greedy learning algorithm which finds a quasi-optimal sequential strategy for a given binary-valued decision problem. WaldBoost finds a sequential strategy S^* such that

$$S^* = \arg\min_S \bar{T}_S \qquad \text{subject to} \quad \beta_S \leq \beta, \quad \alpha_S \leq \alpha \tag{1}$$

for specified α and β. $\bar{T}_S$ is average time-to-decision, α_S is false negative and β_S false positive rate of the sequential strategy S.

A sequential strategy is any algorithm (in our case a classifier) which evaluates one measurement at a time. Based on the set of measurements obtained up to that time, it either decides for one of the classes or postpones the decision. In the latter case, the decision process continues by taking another measurement.

To find the optimal sequential strategy S^*, the WaldBoost algorithm combines the AdaBoost algorithm [11] for feature (measurement) selection and Wald's sequential probability ratio test (SPRT) [12] for finding the thresholds which are used for making the decisions.

The input of the algorithm is a labelled training set of positive and negative samples, a set of features $\mathcal{F}$ - the building blocks of the classifier, and the bounds on the final false negative rate α and the false positive rate β. The output is an ordered set of weak classifiers $h^{(t)}$, $t \in \{1, \ldots, T\}$ each one corresponding to one feature and a set of thresholds $\theta_A^{(t)}$, $\theta_B^{(t)}$ on the response of the strong classifier for all lengths t. During the evaluation of the classifier on a new observation x, one weak classifier is evaluated at time t and its response is added to the response function

$$f_t(x) = \sum_{q=1}^{t} h^{(q)}(x). \tag{2}$$

The response function f_t is then compared to the corresponding thresholds and the sample is either classified as positive or negative, or the next weak classifier is evaluated and the process continues

$$H_t(x) = \begin{cases} +1, & f_t(x) \geq \theta_B^{(t)} \\ -1, & f_t(x) \leq \theta_A^{(t)} \\ \text{continue}, & \theta_A^{(t)} < f_t(x) < \theta_B^{(t)} \end{cases}. \tag{3}$$

If a sample x is not classified even after evaluation of the last weak classifier, a threshold γ is imposed on the real-valued response $f_T(x)$.

Early decisions made in classifier evaluation during training also affect the training set. Whenever a part of the training set is removed according to eq. 3, new training samples are collected (bootstrapped) from yet unseen images.

In the experiments we use the same asymmetric version of WaldBoost as used in [1]. When setting the β parameter to zero, the strategy becomes

$$H_t(x) = \begin{cases} -1, & f_t(x) \leq \theta_A^{(t)} \\ \text{continue}, & \theta_A^{(t)} < f_t(x) \end{cases} \tag{4}$$

and only decisions for the negative class are made during the sequential evaluation of the classifier. A (rare) positive decision can only be reached after evaluating all T classifiers in the ensemble.

In the context of fast black box algorithm emulation, what distinguishes training for different algorithms is the feature set $\mathcal{F}$. A suitable set has to be found for every algorithm. Hence, instead of optimising the algorithm itself, the main burden of development lies in finding a proper set $\mathcal{F}$. The set $\mathcal{F}$ can be very large if one is not sure which features are the best. The WaldBoost algorithm selects a suitable subset together with optimising the time-to-decision.

3 Emulated Scale Invariant Interest Point Detectors

In order to demonstrate the approach, two similarity invariant interest point detectors have been chosen to be emulated: (i) Hessian-Laplace [2] detector, which is a state of the art similarity invariant detector, and (ii) Kadir-Brady [3] saliency detector, which has been found valuable for categorisation, but is about 100× slower. Binaries of both detectors are publicly available[1]. We follow standard test protocols for evaluation as described in [7]. Both detectors are similarity invariant (not affine), which is easily implemented via a scanning window over positions and scales plus a sequential test.

For both detectors, the set $\mathcal{F}$ contains the Haar-like features proposed by Viola and Jones [6], plus a centre-surround feature from [13], which has been shown to be useful for blob-like structure detectors [4]. Haar-like features were chosen for their high evaluation speed (due to integral image representation) and since they have a potential to emulate the Hessian-Laplace detections [4]. For the Kadir-Brady saliency detector emulation, however, the Haar-like features turned out not to be able to emulate the entropy based detections. To overcome this, and still keep the efficiency high, "energy" features based on the integral images of squared intensities were introduced. They represent intensity variance in a given rectangle.

To collect positive and negative samples for training, a corresponding detector is run on a set of images of various sizes and content. The considered detectors assign a scale to each detected point. Square patches of the size twice the scale are used as positive samples. The negative samples representing the "background"

[1] `http://www.robots.ox.ac.uk/~vgg/research/affine/`

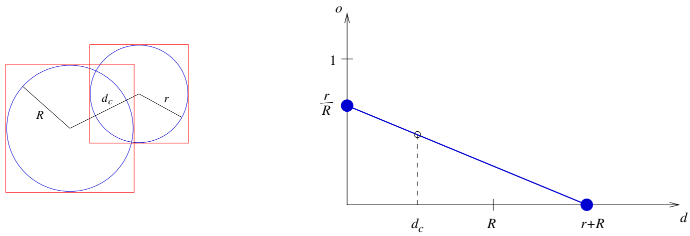


Fig. 2. The non-maximum suppression algorithm scheme for two detections

class are collected from the same images at positions and scales not covered by positive samples.

Setting α. There is no error-free classification, the positive and negative classes are highly overlapping in feature space. As a consequence, the WaldBoost classifier responses on many positions and scales – false positives. One way of removing less reliable detections is to threshold the final response function f_T at some higher value γ. This would lead to less false positives, more false negatives and *very slow classifier* (whole classifier evaluated for most samples). A better option is to set α to a higher value and let the training to prune the negative class sequentially. Again, it results in less false positives and controllable amount of false negatives. Additionally, the classifier becomes much faster due to early decisions.

An essential part of a detector is the **non-maximum suppression algorithm**. Here the output differs from that obtained from the original detectors. Instead of having a real-valued map over whole image, sparse responses are returned by the WaldBoost detector due to early decisions – value of f_t, $t < T$ available for early decisions is not comparable to f_T of positive detections. Thus a typical cubic interpolation and a local maximum search cannot be applied. Instead, the following algorithm is used.

Any two detections are grouped together if their overlap is higher than a given threshold (parameter of the application). Only the detection with maximal f_T in each group is preserved. The overlap computation is schematically shown in Figure 2. Each detection is represented by a circle inscribed to the box (scanning window) reported as a detection (Figure 2, left). For two such circles, let us denote radius of the smaller circle as r and radius of the bigger one as R. The distance of circle centres will be denoted by d. The following approximation to the actual circles overlap is used to avoid computationally demanding goniometric functions.

The measure has an easy interpretation in two cases. First, when the circle centres coincide, the overlap is approximated as r/R. It equals to one for two circles of the same radius and decreases as the radiuses become different. Second, when two circles have just one point in common ($d = r + R$), the overlap is zero. These two situations are marked in Figure 2, right by blue dots. Linear interpolation (blue solid line in Figure 2, right) is used to approximate the overlap between these two states. Given two radiuses r and R where $r \leq R$ and circle centres distance d_c, the overlap o is computed as

$$o = \frac{r}{R}\left(1 - \frac{d_c}{r+R}\right).$$

4 Experiments

This section describes experiments with two WaldBoost-emulated detectors - Hessian-Laplace [2] and Kadir-Brady [3] saliency detector. The Hessian-Laplace detector is expected to be easily emulated due to its blob-like detections. This allows to keep the first experiment more transparent. The Kadir-Brady detector is more complex one due to its entropy based detections. Kadir-Brady detector shows rather poor results in classical repeatability tests [7] but has been successfully used in several recognition tasks [14]. However, its main weakness for practical applications is its very long computation time (in order of minutes per image!).

4.1 Hessian-Laplace Emulation

The training set for the WaldBoost emulation of Hessian-Laplace is created from 36 images of various sizes and content (nature, urban environment, hand drawn, etc.) as described in §3. The Hessian-Laplace detector is used with threshold 1000 to generate the training set. The same threshold is used throughout all the experiments for both learning and evaluation.

Training has been run for $T = 20$ (training steps) with $\alpha = 0.2$ and $\beta = 0$. The higher α allows fast pruning of less trustworthy detections during sequential evaluation of the detector.

The detector has been assessed in standard tests proposed by Mikolajczyk et al. [7]. First, repeatability of the trained WaldBoost detector has been compared with the original Hessian-Laplace detector on several image sequences with variations in scale and rotation. The results on two selected sequences, BOAT and EAST_SOUTH, from [15] are shown in Figure 3 (top row). The WaldBoost detector achieves similar repeatability as the original Hessian-Laplace detector.

In order to test the trained detectors for their applicability, a matching application scenario is used. To that effect, a slightly different definition of matching score is used than that of Mikolajczyk [7]. Matching score as defined in [7] is computed as the number of *correct matches* divided by the smaller number of correspondences in common part of the two images. However, the matches are computed only pairwise for correspondences determined by the geometry ground truth. Here, the same definition of the matching score is used, but the definition of a correct match differs. First, tentative matches using the SIFT detector are computed and mutually nearest matches are found. These matches are then verified by the geometry ground truth and only the verified matches are called correct.

Comparison of the trainer and the trainee outputs on two sequences is given in Figure 3 (bottom row). The WaldBoost detector achieves similar matching score on both sequences while producing consistently more detections and matches.

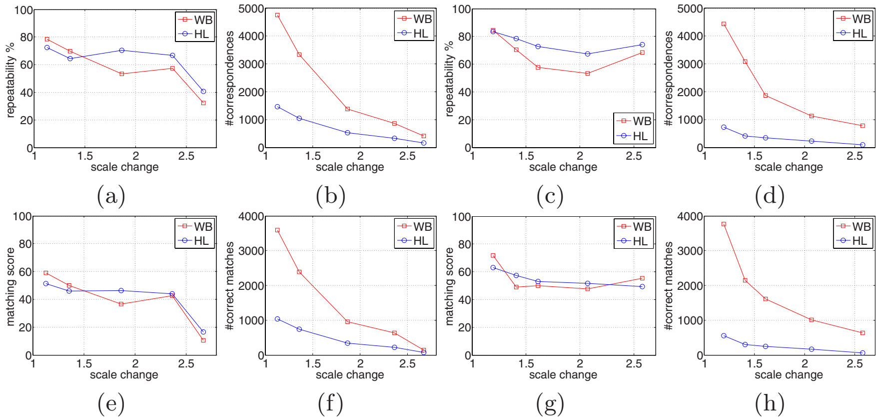


Fig. 3. Comparison of Hessian-Laplace detector and its WaldBoost emulation. Top row: Repeatability on BOAT (a) and EAST_SOUTH (c) sequences and corresponding number of detections (b), (d). Bottom row: Matching score (e), (g) and corresponding number of correct matches (f), (h) on the same sequences.

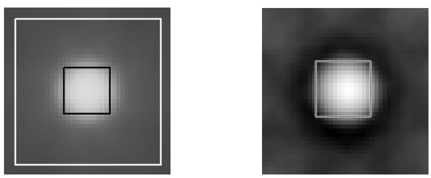
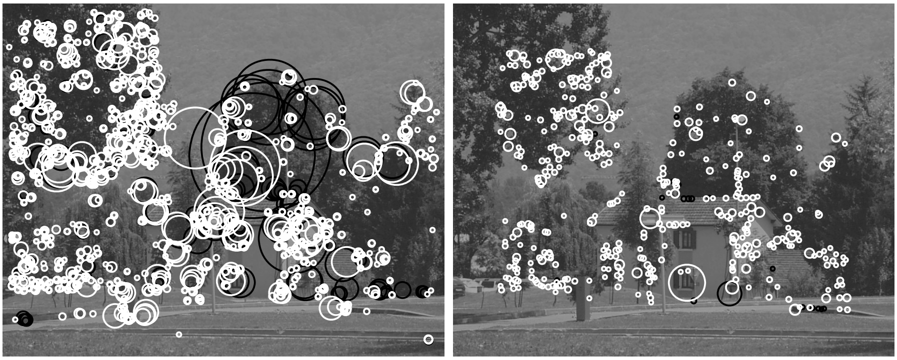


Fig. 4. First centre-surround and energy feature found in WaldBoost Hessian-Laplace (left) and Kadir-Brady (right) emulated detector. The underlying image is generated as $E(|x_i - 127.5|)$ and $E(x_i)$ respectively, where $E()$ is the average operator and x_i is the i-th positive training example.

The WaldBoost classifier evaluates on average 2.5 features per examined position and scale. This is much less than any reported speed for face detection [1]. The evaluation times are compared in Table 1. The WaldBoost emulation speed is comparable to manually tuned Hessian-Laplace detector.

The Hessian-Laplace detector finds blob-like structures. The structure of the trained WaldBoost emulation should reflect this property. As shown in Figure 4, the first selected feature is of a centre-surround type which gives high responses to blob-like structures.

The outputs of the trained WaldBoost emulation of Hessian-Laplace and the original algorithm are compared in Figure 5. To find the original Hessian-Laplace detection correctly found by the WaldBoost emulator, correspondences based on Mikolajczyk's overlap criterion [7] have been found between the original and WaldBoost detections. The white circles show repeated correspondences. The black circles show the detections not found by the WaldBoost emulation. Note that most of the missed detections have a correct detection nearby, so the



(a) (b)

Fig. 5. Comparison of the outputs of the original and WaldBoost-emulated (a) Hessian-Laplace and (b) Kadir-Brady saliency detectors. The white circles show repeated detection. The black circles highlight the original detections not found by the WaldBoost detector. Note that for most of missed detections there is a nearby detection on the same image structure. The accuracy of the emulation is 85 % for Hessian-Laplace and 96 % for Kadir-Brady saliency detector. Note that the publicly available Kadir-Brady algorithm does not detect points close to image edges.

corresponding image structure is actually found. The percentage of repeated detections of the original algorithm is 85 %.

To conclude, the WaldBoost emulator of the Hessian-Laplace detector is able to detect points with similar repeatability and matching score while its speed is comparable to speed of the original algorithm. This indicates that the proposed approach is able to minimise the decision time down to a manually tuned algorithm speed.

4.2 Fast Saliency Detector

The emulation of the Kadir-Brady saliency detector [3] was trained on the same set of images as the WaldBoost Hessian-Laplace emulator. The saliency threshold of the original detector was set to 2 to limit the positive examples only to those with higher saliency. Note, that as opposed to the Hessian-Laplace emulation where rather low threshold was chosen, it is meaningful to use only the top most salient features from the Kadir-Brady detector. This is not true for Hessian-Laplace detector since its response does not correspond to the importance of the feature.

The Haar-like feature set was extended by the "energy" feature described in §3. The training was run for $T = 20$ (training steps) with $\alpha = 0.2$ and $\beta = 0$.

The same experiments as for the Hessian-Laplace detector have been performed. The repeatability and the matching score of the Kadir-Brady detector and its WaldBoost emulation on BOAT and EAST_SOUTH sequences are shown in Figure 6. The trained detector performs slightly better than the original one.

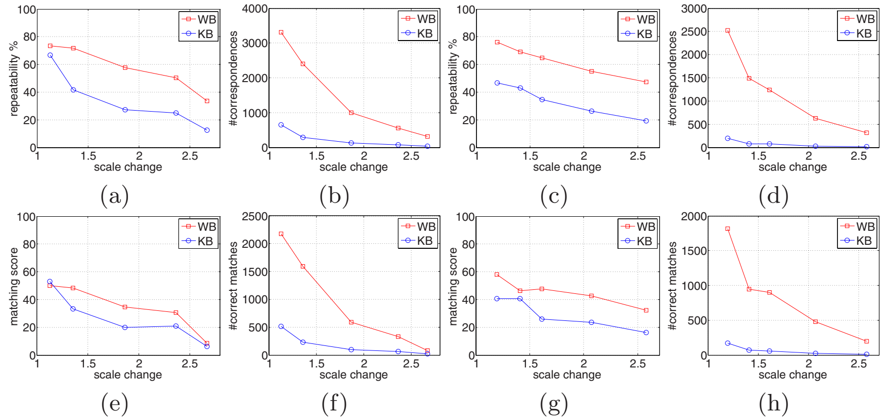


Fig. 6. Comparison of Kadir-Brady detector and its WaldBoost emulation. Top row: Repeatability on BOAT (a) and EAST_SOUTH (c) sequences and corresponding number of detections (b), (d). Bottom row: Matching score (e), (g) and corresponding number of correct matches (f), (h) on the same sequences.

Table 1. Speed comparison on the first image (850×680) from the BOAT sequence

	original	**WaldBoost**
Hessian-Laplace	1.3s	1.3s
Kadir-Brady	1m 44s	1.4s

The main advantage of the emulated saliency detector is its speed. The classifier evaluates on average 3.7 features per examined position and scale. Table 1 shows that the emulated detector is 70× faster than the original detector.

Our early experiments showed that the Haar-like features are not suitable to emulate the entropy-based saliency detector. With the energy features, the training was able to converge to a reasonable classifier. In fact, the energy feature is chosen for the first weak classifier in the WaldBoost ensemble (see Figure 4).

The outputs of the WaldBoost saliency detector and the original algorithm are compared in Figure 5. The coverage of the original detections is 96 %.

To conclude, the Kadir-Brady emulation gives slightly better repeatability and matching score. But, most importantly, the decision times of the emulated detector are about 70× lower than that of the original algorithm. That opens new possibilities for using the Kadir-Brady detector in time sensitive applications.

5 Conclusions and Future Work

In this paper a general learning framework for speeding up existing binary-valued decision algorithms by a sequential classifier learned by WaldBoost algorithm has been proposed. Two interest point detectors, Hessian-Laplace and Kadir-Brady saliency detector, have been used as black box algorithms and emulated by the

WaldBoost algorithm. The experiments show similar repeatability and matching scores of the original and emulated algorithms. The speed of the Hessian-Laplace emulator is comparable to the original manually tuned algorithm, while the Kadir-Brady detector was speeded up seventy times.

The proposed approach is general and can be applied to other algorithms as well. For future research, an interesting extension of the proposed approach would be to train an emulator with not only similar outputs to an existing algorithm but also with some additional quality like higher repeatability or specialisation to a given task.

Acknowledgement

The authors were supported by Czech Science Foundation Project 102/07/1317 (JM) and by EC project FP6-IST-027113 eTRIMS (JŠ).

References

1. Šochman, J., Matas, J.: WaldBoost - learning for time constrained sequential detection. In: CVPR, Los Alamitos, USA, vol. 2, pp. 150–157 (2005)
2. Mikolajczyk, K., Schmid, C.: Scale and affine invariant interest point detectors. IJCV 60(1), 63–86 (2004)
3. Kadir, T., Brady, M.: Saliency, scale and image description. IJCV 45(2) (2001)
4. Grabner, M., Grabner, H., Bischof, H.: Fast approximated SIFT. In: ACCV, pp. I:918–927 (2006)
5. Bay, H., Tuytelaars, T., Van Gool, L.: SURF: Speeded up robust features. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, Springer, Heidelberg (2006)
6. Viola, P., Jones, M.: Robust real time object detection. In: SCTV, Vancouver, Canada (2001)
7. Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., Van Gool, L.: A comparison of affine region detectors. In: IJCV (2005)
8. Lepetit, V., Lagger, P., Fua, P.: Randomized trees for real-time keypoint recognition. In: CVPR, vol. II, pp. 775–781 (2005)
9. Rosten, E., Drummond, T.: Machine learning for high-speed corner detection. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 430–443. Springer, Heidelberg (2006)
10. Martin, D., Fowlkes, C., Malik, J.: Learning to detect natural image boundaries using local brightness, color, and texture cues. PAMI 26(5), 530–549 (2004)
11. Schapire, R.E., Singer, Y.: Improved boosting algorithms using confidence-rated predictions. Machine Learning 37(3), 297–336 (1999)
12. Wald, A.: Sequential analysis. Dover, New York (1947)
13. Lienhart, R., Maydt, J.: An extended set of haar-like features for rapid object detection. In: ICIP (2002)
14. Fergus, R., Perona, P., Zisserman, A.: A sparse object category model for efficient learning and exhaustive recognition. In: CVPR, vol. 1, pp. 380–387 (2005)
15. Mikolajczyk, K.: Detection of local features invariant to affines transformations. PhD thesis, INPG, Grenoble (2002)