Int J Comput Vis (2009) 83: 149–163
DOI 10.1007/s11263-009-0229-x

Learning Fast Emulators of Binary Decision Processes

Jan Šochman · Jiří Matas

Received: 6 August 2008 / Accepted: 3 March 2009 / Published online: 19 March 2009

Abstract Computation time is an important performance characteristic of computer vision algorithms. The paper shows how existing (slow) binary decision algorithms can be approximated by a (fast) trained WaldBoost classifier.

WaldBoost learning minimises the decision time of the classifier while guaranteeing predefined precision. We show that the WaldBoost algorithm together with bootstrapping is able to efficiently handle an effectively unlimited number of training examples provided by the implementation of the approximated algorithm.

Two interest point detectors, the Hessian-Laplace and the Kadir-Brady saliency detectors, are emulated to demonstrate the approach. Experiments show that while the repeatability and matching scores are similar for the original and emulated algorithms, a 9-fold speed-up for the Hessian-Laplace detector and a 142-fold speed-up for the Kadir-Brady detector is achieved. For the Hessian-Laplace detector, the achieved speed is similar to SURF, a popular and very fast handcrafted modification of Hessian-Laplace; the WaldBoost emulator approximates the output of the Hessian-Laplace detector more precisely.

Keywords Boosting · AdaBoost · Sequential probability ratio test · Sequential decision making · WaldBoost · Interest point detectors · Machine learning

J. Šochman (✉) · J. Matas
Center for Machine Perception, Faculty of Electrical Engineering, Department of Cybernetics, Czech Technical University in Prague, Technicka 2, 166 27 Prague 6, Czech Republic
e-mail: jan.sochman@cmp.felk.cvut.cz

1 Introduction

In this paper, we propose a novel way of speeding up existing binary decision processes, such as detectors or two-class classifiers. We show that large efficiency gains are obtained automatically, by using a statistical machine learning method that produce a fast and accurate approximation of the original process. The approach is successfully demonstrated on two commonly-used interest point detectors. Nevertheless, the approach is general and is applicable to other areas e.g. edge detection.

Standard learning algorithms like SVM, AdaBoost or neural networks are designed primarily with the objective of error minimisation and generalisation to unseen data; small training size performance is an important concern. Typical evaluation of learning algorithms reflects this focus—measures like the precision-recall curve or the false positive and the false negative rates are usually computed on a test set. However, for the problem of efficient approximation, another aspect of the trained classifier becomes critical—the time-to-decision, a property directly optimised by very few machine learning methods.

Another uncommon feature of our setting is the training set size. Since our objective is to learn an emulator of an existing binary-decision process, labelled training samples are obtained by running the process on unlabelled data. If unlabelled data are easily accessible, which is common, a training set of arbitrary size can be collected at effectively zero cost. The speeding up problem thus becomes a problem of learning the algorithm's outputs on the (very large) training set while optimising the classification speed.

Very few approaches consider time-to-decision as an integral part of the learning task. In this work, the WaldBoost learning algorithm (Šochman and Matas 2005) was adopted as it handles the precision-speed trade-off automatically and produces a quasi-optimal *sequential classifier* minimising the decision time while guaranteeing the predefined emulation precision. The user influences the emulation process by defining suitable feature sets from which the emulator is built and by specifying constraints on the classifier's precision.

We demonstrate the framework by emulating two interest point detectors, Hessian-Laplace (Mikolajczyk and Schmid 2004) and Kadir-Brady (Kadir and Brady 2001) saliency detector. The Hessian-Laplace is a state-of-the-art detector of blob-like structures. Moreover, a handcrafted simplified version of Hessian-Laplace called SURF (Bay et al. 2008), designed for maximum speed, is available for comparison. The Kadir-Brady detector incorporates entropy measure to find salient regions which has been successfully used in several recognition tasks (Fergus et al. 2005; Zhu et al. 2006).

The contributions of the paper are: (i) an introduction of a general method for improving performance of computer vision algorithms by a machine learning method, (ii) showing how to apply WaldBoost learning algorithm for emulation of interest point detectors, (iii) impressive speed-up of one (Hessian-Laplace) and two (Kadir-Brady) orders of magnitude respectively for the emulated detectors, (iv) a detector comparable in performance and speed with the SURF detector, a de facto standard for very fast interest point detection.

The rest of the paper is structured as follows. First a brief state of the art overview is given in Sect. 2. The approximation of a black-box binary decision algorithm by a WaldBoost classifier is discussed in Sect. 3. Application of the approach to interest point detection is described in Sect. 4. Experiments are presented in Sect. 5 and the paper is concluded in Sect. 6. A conference version of this paper appeared in Šochman and Matas (2007).

2 State of the Art

Learning to Be Fast The history of the formulation of a classification task with time-to-decision vs. precision trade-off dates back to Wald's sequential analysis (Wald 1947). Wald posed the problem as a constrained optimisation and found a quasi-optimal solution to it—the sequential probability ratio test (SPRT). Wald's theory assumes knowledge of the class conditional probabilities and it does not consider learning and estimation issues. The theory of sequential decision-making has been further developed and enriched (Siegmund 1985) and is now used as a basic and well known tool in statistics. An overview of computer vision methods based on the SPRT can be found in Matas and Šochman (2007).

In 1987, Rivest (1987) studied learnability of decision lists (which could be seen as sequential classifiers) in the context of Boolean functions but without optimising the evaluation time. Baker and Nayar (1996) looked at the problem of efficiency of classification in the context of multiclass classification, where the task is to effectively distinguish one class out of many. To this end, they developed a theory of pattern rejectors which can be interpreted as sequential classifiers in the class space. A practical learning approach to the time-to-decision vs. precision trade-off has been proposed by Viola and Jones (2001), who build an ordered set of increasingly complex classifiers that were applied sequentially to a progressively smaller fraction of the data. The "classifier cascade" method requires the user to define the complexities of individual classifiers and does not optimise the time vs. precision trade-off directly. Consequently, many variations on the method have appeared in the literature (Xiao et al. 2003; Huang et al. 2007; Brubaker et al. 2008). The SoftCascade (Bourdev and Brandt 2005) algorithm presents a systematic but a rather brute force approach to the precision vs. speed optimisation problem.

Learning Interest Point Detectors There has been much work on the general interest point detection problem (Mikolajczyk et al. 2005). To our knowledge, learning techniques have been applied only to parameter tuning, not to the whole process of interest point detector design. Lepetit et al. (2005) treated interest points matching as a classification problem, learning the descriptor. Rosten and Drummond (2006) used learning techniques to find parameters of a hand-designed tree-based Harris-like corner classifier. Their motivation was to speed-up the detection process, but the approach is limited to the Harris corner detection. Martin et al. (2004) learned a classifier for edge detection, but without considering the decision time and with significant manual tuning. They tested a number of classifier types with the conclusion that a boosted classifier was comparable in performance to other classifiers and was preferable for its low model complexity and low computational cost.

The most closely related approach to our method is that of Dollár et al. (2006) who use learning techniques to train an edge detector. The paper shows impressive examples of applications of such detector. Nevertheless, Dollár et al. were primarily concerned with the accuracy of the detector and did not consider speed. There has also been significant interest in speeding up various interest point detectors manually, i.e. without training. Grabner et al. (2006) proposed a fast version of the SIFT detector and Bay et al. (2008) proposed a fast approximation-based interest point detector called SURF.

3 Emulating a Binary-Decision Black Box Algorithm with WaldBoost

The main idea of the proposed approach is to look at an existing algorithm as a black box performing some useful bi-

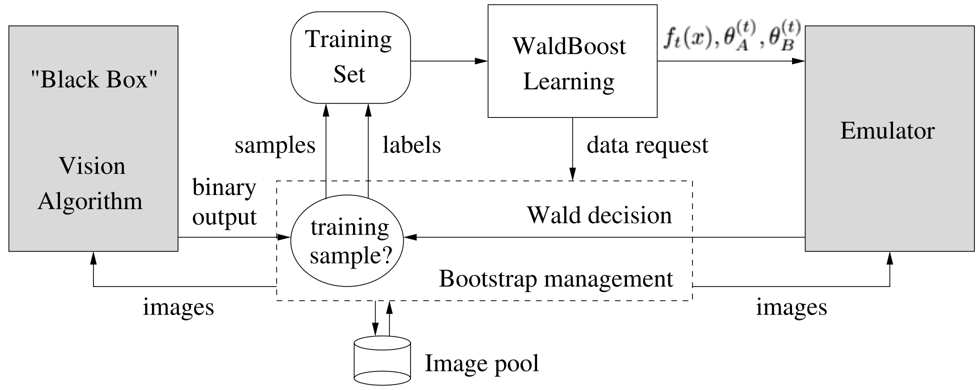


Fig. 1 The proposed learning scheme

nary decision task. The black box algorithm is run on a large dataset of images which provides almost unlimited number of training samples which are used to train a sequential classifier emulating the black box algorithm behaviour. The user's optimisation effort is thus transformed into a much simpler task of finding a suitable set of features which are used in the WaldBoost training.

The main components of the proposed learning system are shown in Fig. 1. The black box algorithm provides positive and negative outputs that form a labelled training set. The WaldBoost learning algorithm (see Sect. 3.1) 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 undecided by the current classifier are used for further training. The result of the process is a WaldBoost sequential classifier which emulates the original black box algorithm.

The training loop uses the fact that the black box algorithm can provide practically unlimited number of labelled training samples. Note that this is in contrast to commonly used human labelled data which are difficult to obtain. The bootstrapping technique (Sung and Poggio 1998) is used to effectively update the training set.

Next, a brief overview of the WaldBoost learning algorithm, the core unit in the emulation scheme, is given.

3.1 WaldBoost

WaldBoost (Šochman and Matas 2005) is a greedy learning algorithm which finds a quasi-optimal sequential strategy minimising the average evaluation time while preserving required quality of the decision for a given binary decision problem. More formally, WaldBoost finds a *sequential decision strategy* S^* such that

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

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

A sequential decision strategy is a sequence of decision functions $S = S_1, S_2, \ldots,$ where $S_t : (x_1, \ldots, x_t) \to \{-1, +1, \sharp\}$. The strategy S takes one more measurement, x_t, at a time and in step t makes a decision S_t based on $(x_1, \ldots, x_t)$. The '$\sharp$' sign stands for a "continue" (do not decide yet) decision. If a decision is '$\sharp$', x_{t+1} is measured and S_{t+1} is evaluated. Otherwise, the output of S is the class returned by S_t.

To find the optimal sequential strategy S^* to the problem (1), the WaldBoost algorithm combines the AdaBoost algorithm (Schapire and Singer 1999) for measurement selection and Wald's sequential probability ratio test (SPRT) (Wald 1947) for finding thresholds which are used for making the decisions.

The SPRT was proved to be an optimal strategy for the problem (1). The SPRT is very simple—in each step it compares likelihood ratio with a fixed threshold. Such test is easy to evaluate for i.i.d. measurements where the likelihood ratio is easy to estimate. However, when the measurements are not i.i.d., the likelihood ratio estimation easily becomes intractable and the ordering of the measurements has to be specified. To overcome these problems, WaldBoost uses the AdaBoost algorithm as a measurement selector and also for projecting measurements (see (2)) to a 1D subspace where likelihood ratio estimation is tractable. This is justified by the fact that the response of AdaBoost, $f_t(x)$, converges to the likelihood ratio (Friedman et al. 1998).

The AdaBoost algorithm greedily selects *weak classifiers* $h^{(t)}: \mathcal{X} \to \mathbb{R}$ which are combined linearly into a strong classifier

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

The domain-partitioning weak classifiers (Schapire and Singer 1999) are used, each one based on a single (visual) feature (see Fig. 2). The response of the weak classifiers found by the AdaBoost algorithm are used as measurements for the sequential strategy in the WaldBoost algorithm.

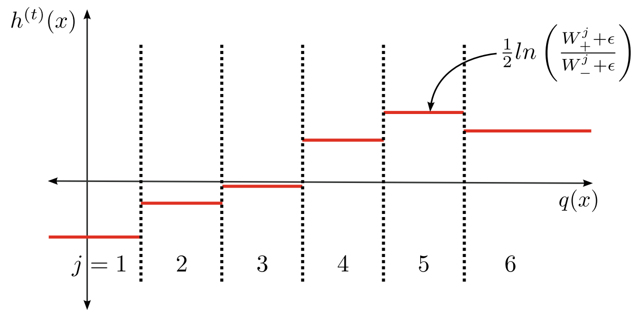


Fig. 2 The domain-partitioning weak classifier. The response of feature $q(x)$ on object x is partitioned into bins $j = 1, \dots, K$. The leftmost and the rightmost bins cover respective half-spaces. In each bin j, the response of the weak classifier $h(x)$ is computed from the sum of positive (W_+^j) and negative (W_-^j) weights of training samples falling into the bin. To avoid numerical problems, a smoothing factor ϵ is used (Schapire and Singer 1999)

The input of the learning algorithm (Algorithm 1) is a pool of positive and negative samples $\mathcal{P}$, 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 $h^{(t)}$, $t \in \{1, \dots, T\}$ (i.e. measurements) and a set of SPRT thresholds $\theta_A^{(t)}$, $\theta_B^{(t)}$ optimising (1) for all lengths $t = 1, \dots, T$. The thresholds are applied directly to the strong classifier response f_t (not to the likelihood ration as in SPRT) and are set to $\pm\infty$ if no threshold was found in some learning step.

In the learning, the selection of a weak classifier is by far the most time consuming operation. To keep the speed and memory requirements of the training process acceptable, a subset $\mathcal{T}$ is sampled out of the large sample pool $\mathcal{P}$; the selection of the best weak classifier is based on $\mathcal{T}$. The SPRT thresholds are efficiently computed on the whole pool.

The sequential nature of the WaldBoost classifier also affects the sample pool and the training set during the learning. In each round, the already decidable samples in the pool (see explanation for WaldBoost evaluation below) are removed from the learning process and a new training set $\mathcal{T}$ is sampled from the reduced pool.

During evaluation of the classifier (Algorithm 2) on a new sample x, one weak classifier is evaluated at time t and its response is added to the strong classifier response function f_t. It 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

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

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

Algorithm 1 WaldBoost Learning

Input:

- sample pool $\mathcal{P} = \{(x_1, y_1), \dots, (x_m, y_m)\}$; $x_i \in \mathcal{X}$, $y_i \in \{-1, 1\}$,
- set of features $\mathcal{F} = \{q_s\}$,
- desired final false negative rate α and false positive rate β,
- the number of iterations T.

Sample randomly the initial training set $\mathcal{T}$ from the pool $\mathcal{P}$

for $t = 1, \dots, T$

1. Find $h^{(t)}$ by AdaBoost using $\mathcal{F}$ and $\mathcal{T}$ and add it to the strong classifier

$$f_t(x) = \sum_{r=1}^{t} h^{(r)}(x)$$

2. Find decision thresholds $\theta_A^{(t)}$ and $\theta_B^{(t)}$ for f_t using $\mathcal{P}$
3. Bootstrap: update the sample pool $\mathcal{P}$ and sample a new training set $\mathcal{T}$

end

Output: ordered set of weak classifiers $h^{(t)}$ and thresholds $\theta_A^{(t)}$ and $\theta_B^{(t)}$.

Algorithm 2 WaldBoost Classification

Given: $h^{(t)}, \theta_A^{(t)}, \theta_B^{(t)}, \gamma \quad (t = 1, \dots, T)$

Input: a classified object x.

for $t = 1, \dots, T$

If $f_t(x) \geq \theta_B^{(t)}$, classify x to the class $+1$ and terminate

If $f_t(x) \leq \theta_A^{(t)}$, classify x to the class -1 and terminate

end

If $f_T(x) > \gamma$, then classify x as $+1$ else classify x as -1.

In our interest point detection application of WaldBoost, an arbitrary number of both positive and negative samples is available for bootstrapping. However, when positive samples were bootstrapped, i.e. early positive classification was allowed in (3), all early positive decisions had confidence close to $\theta_B^{(t)}$ and precise localisation via the non-maximum suppression algorithm (see Sect. 4) was not possible. Thus, we adopted the same asymmetric version of WaldBoost as used in Šochman and Matas (2005), i.e. setting β to zero. The strategy becomes

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

and only decisions for the negative class are made early during the sequential evaluation of the classifier. A (rare) positive decision can only be reached after evaluating all T classifiers in the ensemble. For problems where the non-maximum suppression algorithm is not applied, the strategy (3) can be used directly.

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 emulated algorithm. The set $\mathcal{F}$ can be very large and does not need to be homogeneous, i.e. it may contain Haar-like features (Viola and Jones 2001), LBP (Ojala et al. 2002; Froba and Ernst 2004), histograms of gradients, etc. The WaldBoost algorithm selects a suitable subset while optimising the time-to-decision. WaldBoost minimises the average number of evaluated measurements which is the same as minimisation of time-to-decision only when computational complexity of the different types of features is (roughly) the same. The condition is satisfied by the feature set $\mathcal{F}$ adopted in the experiments.

4 Emulated Scale Invariant Interest Point Detectors

In order to demonstrate the approach, two similarity-invariant interest point detectors have been chosen: (i) Hessian-Laplace (Mikolajczyk and Schmid 2004) detector, which is a state of the art similarity-invariant detector, and (ii) Kadir and Brady (2001) saliency detector, which has been found valuable for categorisation (Fergus et al. 2005; Zhu et al. 2006), but is about 100× slower than the Hessian-Laplace detector. Binaries of both detectors were downloaded from the web page (Mikolajczyk 2008a). We followed standard test protocols for evaluation as described in Mikolajczyk et al. (2005). Both detectors are similarity-invariant (not affine), so the detection can be easily implemented by running a sequential test at each position and scale in the scanning window approach (Viola and Jones 2001).

For both detectors, the set $\mathcal{F}$ includes the Haar-like features proposed by Viola and Jones (2001), plus a centre-surround feature from Lienhart and Maydt (2005), which has been shown to be useful for blob-like structure detectors (Grabner et al. 2006). Haar-like features were chosen for their high evaluation speed (due to integral image representation) and because they have a potential to emulate the Hessian-Laplace detections (Grabner et al. 2006). The only difference to the original Viola and Jones feature set is that the feature response is not normalised by a window standard deviation since the intensity contrast is important for both Hessian-Laplace and Kadir-Brady detectors.

For the entropy-based Kadir-Brady saliency detector emulation, however, the Haar-like features were not sufficiently accurate. To overcome this we introduced "variance" features based on the integral images of squared intensities. They are computed as an intensity variance in a given rectangle.

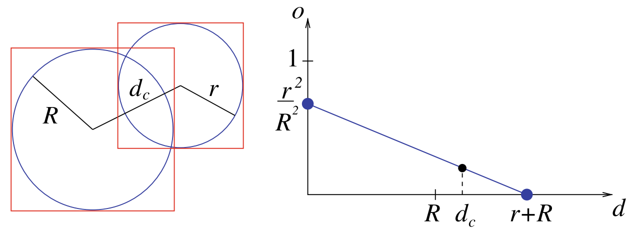


Fig. 3 Overlap definition for the non-maximum suppression scheme. For details, see the text

An essential part of a detector is the *non-maximum suppression* algorithm. Here the input to the non-maximum suppression differs from that obtained in the original detectors. Instead of having a real-valued feature response over whole image, sparse responses are returned by the WaldBoost detector. The accepted positions get the real-valued confidence value f_T, but the rejected positions have the "confidence" f_t around the $\theta_A^{(t)}$ value depending on the time t when they have been rejected. These values are incomparable, thus a typical quadratic interpolation and a local maximum search cannot be applied. Instead, the following algorithm is used.

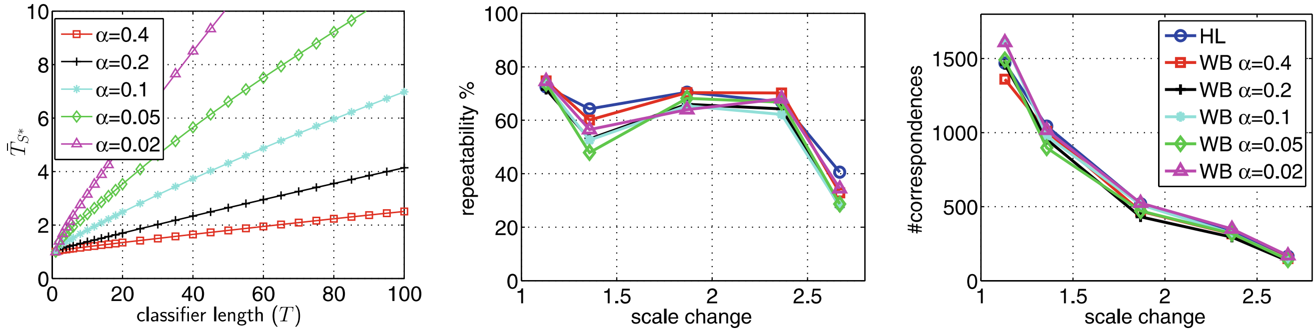
Any two detections are grouped together if their overlap is higher than a given threshold (parameter of the application). Only the detection with maximal f_T in each group is preserved. The overlap computation is schematically shown in Fig. 3. Each detection is represented by a circle inscribed to the corresponding scanning window (Fig. 3, left). For two such circles, let us denote the radius of the smaller circle as r, the radius of the bigger one as R, and the distance of the circle centres as d_c. Exact overlap can be easily computed in two cases. First, when the circle centres coincide, the overlap is $o = r^2/R^2$. It equals to one for two circles of the same radius and decreases as the radii become different. Second, when two circles have just one point in common ($d_c = r + R$), the overlap is zero. These two situations are marked by blue dots in Fig. 3, right. Linear interpolation (blue solid line in Fig. 3, right)

$$o = \frac{r^2}{R^2}\left(1 - \frac{d_c}{r + R}\right) \tag{5}$$

is used to approximate the overlap between these two states.

5 Experiments

Two detectors are emulated in the experiments: Hessian-Laplace (Mikolajczyk and Schmid 2004) and Kadir-Brady (Kadir and Brady 2001) saliency detector. The Hessian-Laplace is a state-of-the-art detector of blob-like structures used in many applications. Its simplicity allows easier analysis of obtained results. The Kadir-Brady detector incorporates entropy measure to find salient regions. It performs



(a) (b) (c)

α	0.02	0.05	0.1	0.2	0.4
♯det	11472	9427	7106	4203	2624

(d)

threshold	500	1000
♯det	3549	2207

(e)

Fig. 4 Selecting the false negative rate α. (**a**) The average evaluation speed for several values of α. All compared detectors are able to achieve similar number of correspondences and (**b**) repeatability score, (**c**)—measured for $T = 20$ for all detectors on the BOAT sequence. (**d**) The number of detections of the WaldBoost emulator on the first image from the BOAT sequence as a function of the α parameter, (**e**) the number of detections of Hessian-Laplace as a function of the final threshold

rather poor in classical repeatability tests (Mikolajczyk et al. 2005) but has been successfully used in several recognition tasks. However, its main weakness for practical applications is its very long computation time in order of minutes per image. Standard versions of the detectors provided by their authors were downloaded from the interest point detection web page (Mikolajczyk 2008a).

To collect positive and negative samples for training, an emulated detector is run on a set of images of various sizes and content (nature, urban environment, hand drawn, etc.). To create the sample pool we used 1300 images randomly chosen from the non-skin image database introduced in Jones and Rehg (2002). The detector assigns a scale to each detected point. Square patches of the size twice the scale were used as positive samples. Negative samples were collected from the same images at positions and scales not covered by positive samples.

The size of the training set $\mathcal{T}$ was 10,000 (half positive and half negative samples) in all experiments. The training set was sampled from the pool $\mathcal{P}$ by the quasi-random weighted sampling + trimming method (QWS+) (Kálal et al. 2008). The QWS+ sampling has been shown to reduce the variance of hypothesis error estimate and to improve the classifier performance compared to other sampling strategies. Moreover, with QWS+ sampling, AdaBoost performance becomes relatively insensitive to the training set size.

5.1 Hessian-Laplace Emulation

The Hessian-Laplace detector was used with threshold 1000 to generate the training set. The value was empirically chosen to achieve similar number of detections as in Mikolajczyk et al. (2005). The same threshold was used throughout all the experiments for both learning and evaluation.

The detector has been assessed in standard tests proposed by Mikolajczyk et al. (2005). The ground truth is given by a homography between the first and the other images in the sequence. The tests are based on two measures: (i) the repeatability measure, (ii) the matching score.

(i) *Repeatability Measure* To assess the quality of an interest point detector in varying acquisition conditions of the same scene the repeatability measure is used (Mikolajczyk et al. 2005). The measure is defined for two sets of elliptical regions—one set for one image. It is computed as the ratio between the number of region-to-region correspondences and the smaller of the number of regions in the pair of images. The mutual correspondence of two regions is claimed when the overlap error is smaller than some threshold. The measure takes into account several other technical issues such as uniqueness of matches and is fully defined by a Matlab script (Mikolajczyk 2008a). In all experiments, the overlap error threshold is fixed to 40% as in most of the experiments in Mikolajczyk et al. (2005).

(ii) *The Matching Score* test aims at predicting performance of the detectors in matching and correspondence finding applications. The matching score, defined in Mikolajczyk et al. (2005), is the number of correct matches divided by the smaller number of correspondences in the common part of the two images. A pair of elliptical regions is counted as a *correct match* if (1) their overlap error is smaller than 40%, and (2) their descriptors are sufficiently similar (for details, see Mikolajczyk et al. 2005).

Selection of the False Negative Rate α The value of α balances the trade-off between WaldBoost detector speed and precision. Figures 4(a)–(c) shows performance of the detector for several α values on the BOAT sequence. The value of α also significantly influences the number of detections before the final thresholding by γ (Fig. 4(d)).

For a certain range of α values, it is possible to set the final threshold γ (Algorithm 2) to reach the number of correspondences similar to that of the emulated detector (Fig. 4(b)). With such threshold γ, the repeatability and the number of correct correspondences is almost identical for all tested values of α throughout the test sequence (Fig. 4(c)).

Increasing α leads to faster evaluation (Fig. 4(a)) but also to fewer detections (Fig. 4(d)) before imposing the final threshold γ. In some applications it may be useful to produce more detections by changing the γ threshold.

Similarly to the original detector, the WaldBoost emulator imposes a threshold on the classifier response. We set α to 0.2 as a compromise: the classifier is already very fast (see Table 1) and yet the user can still control the number of detections by changing the γ threshold similarly to the original detector (Fig. 4(e)). Thus the value $\alpha = 0.2$ is used in all following experiments. The final threshold γ is the same in all experiments and is set empirically so that the detector produces similar number of detections as the original Hessian-Laplace detector.

Classifier Length Empirically we set the length of the classifier to $T = 20$ (number of weak classifiers). Longer classifiers slow down the evaluation (see Fig. 4(a)) and do not bring significant improvement in performance.

Repeatability The repeatability measure of the trained WaldBoost detector has been compared with the original Hessian-Laplace detector on standard image sequences with variations in scale and rotation, blur, affine deforma-

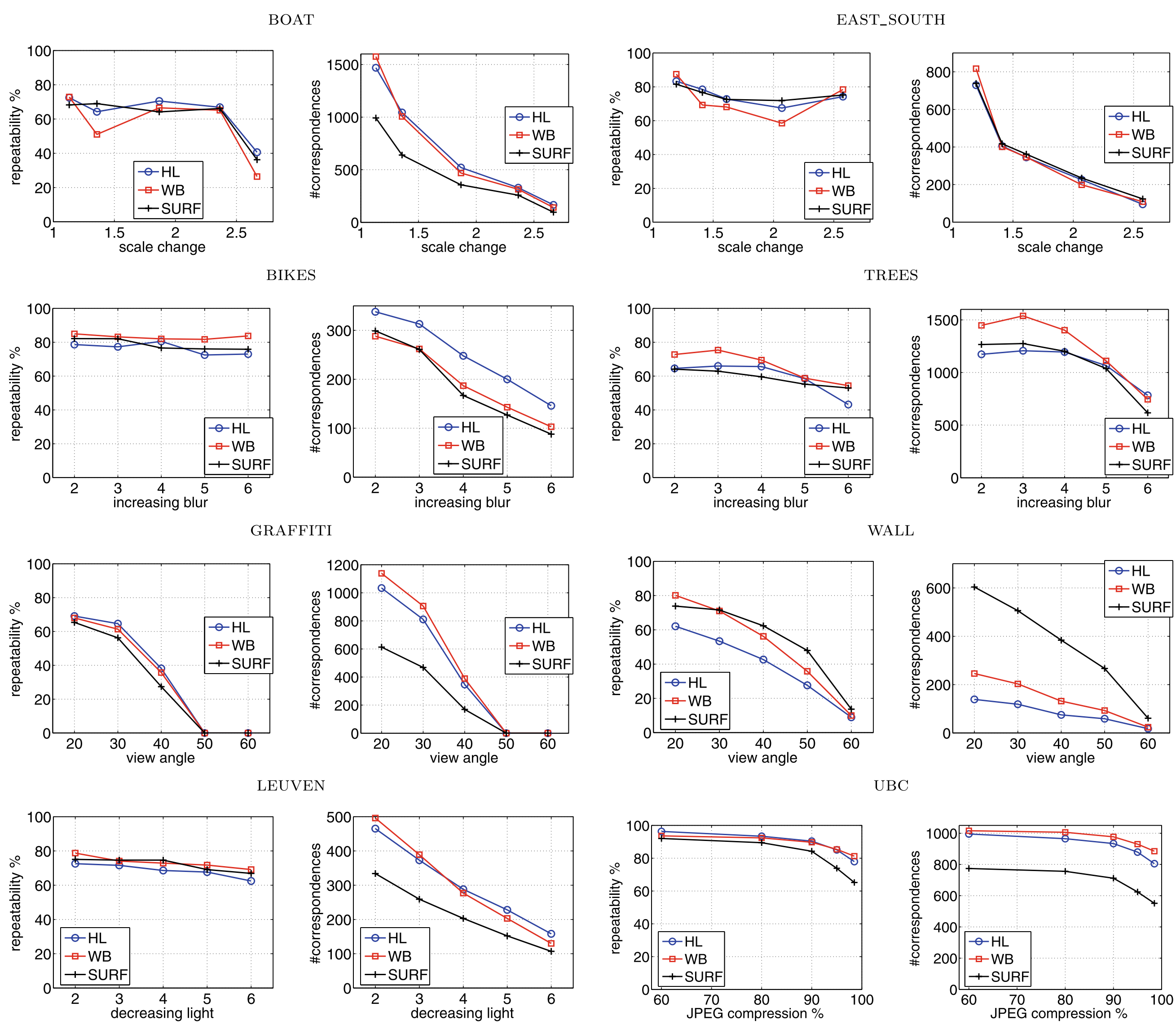


Fig. 5 Repeatability comparison of the Hessian-Laplace detector, its WaldBoost emulation and the SURF detector on Mikolajczyk's dataset

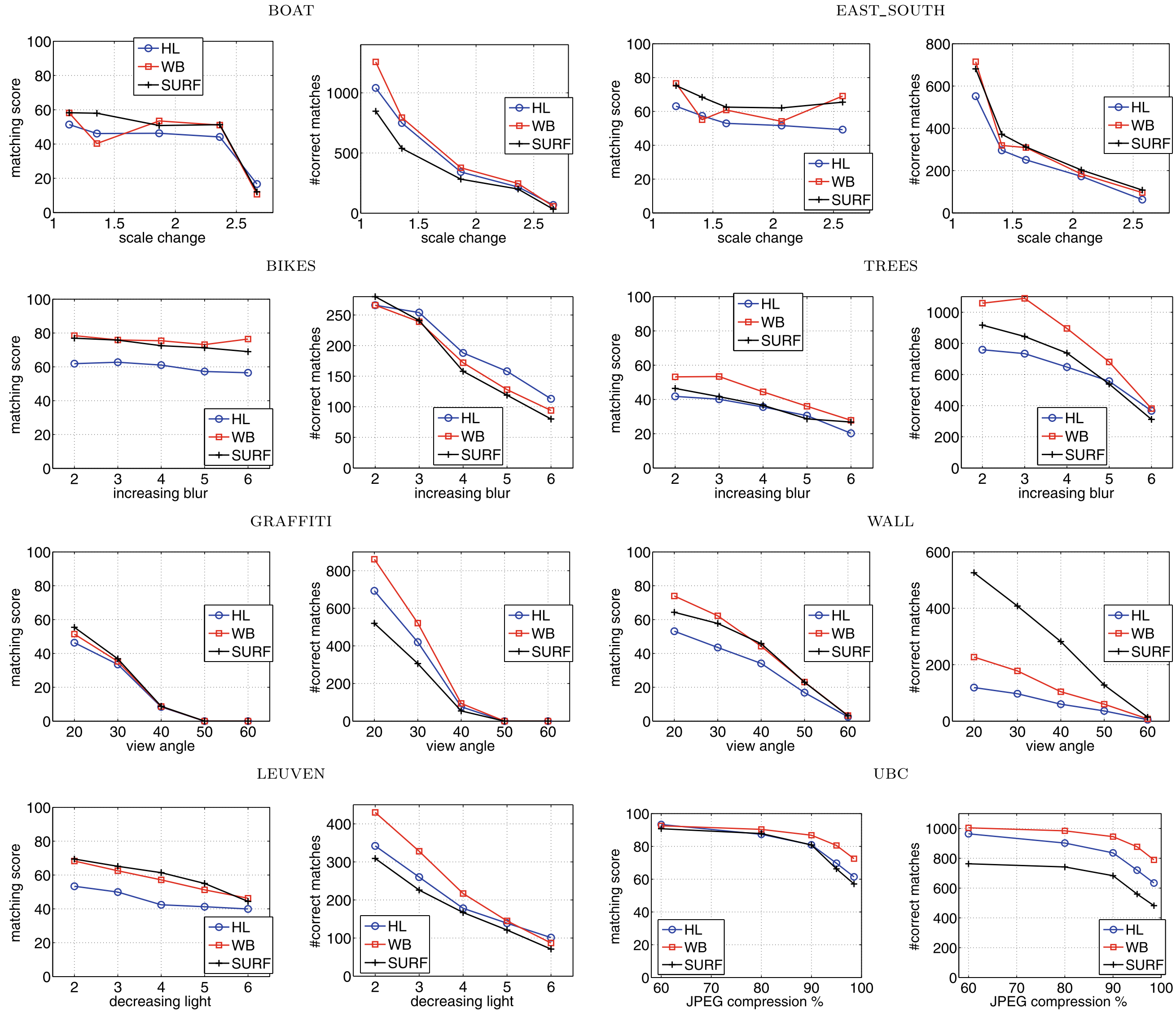


Fig. 6 Matching score comparison of the Hessian-Laplace detector, its WaldBoost emulation and the SURF detector on Mikolajczyk's dataset

tion, light change and JPEG compression from Mikolajczyk (2002). The results are shown in Fig. 5. The WaldBoost detector achieves similar repeatability and number of correspondences as the original Hessian-Laplace detector.

Matching Score For the same sequences, the matching score of the Hessian-Laplace detector ant its WaldBoost emulator is shown in Fig. 6. The WaldBoost detector achieves slightly better matching score than the original algorithm.

Speed The WaldBoost classifier evaluates on average 1.7 features per examined position and scale. Unsurprisingly, this is much less than any reported speed for face detection (Šochman and Matas 2005). The evaluation times are compared in Table 1. The WaldBoost emulator is about nine times faster than the Hessian-Laplace detector with a rather careful design (Mikolajczyk 2008b).

Classifier Structure The Hessian-Laplace detector finds blob-like structures. The structure of the trained WaldBoost emulation should reflect this property. As shown in Fig. 7, the first selected weak classifier is of the centre-surround type and gives high responses to blob-like structures with high contrast between central part and its surrounding (the feature value is average intensity in the central part minus average intensity in the surrounding part).

Coverage The output of the trained WaldBoost emulation of Hessian-Laplace is compared to the original algorithm in Fig. 8(a). As in the repeatability experiment two sets of detections are compared—the original detections and the

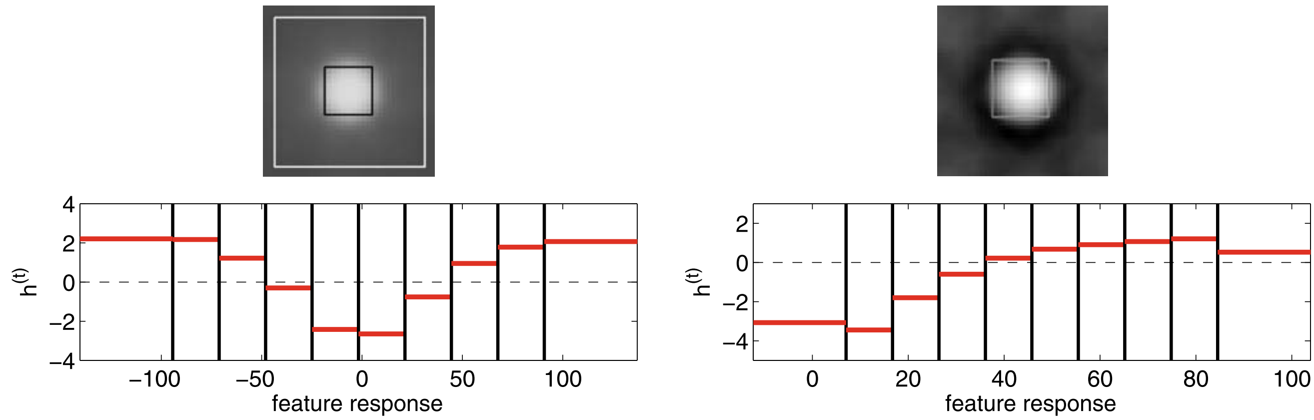


Fig. 7 *Top row.* First centre-surround and variance feature found in WaldBoost Hessian-Laplace (*left*) and Kadir-Brady (*right*) emulated detectors. The background image is visualised as $E(|x_i - 127.5|)$ and $E(x_i)$ respectively, where $E()$ is the average operator and x_i is the i-th positive training example. *Bottom row.* Bin responses in the corresponding domain-partitioning weak classifiers (see Fig. 2)

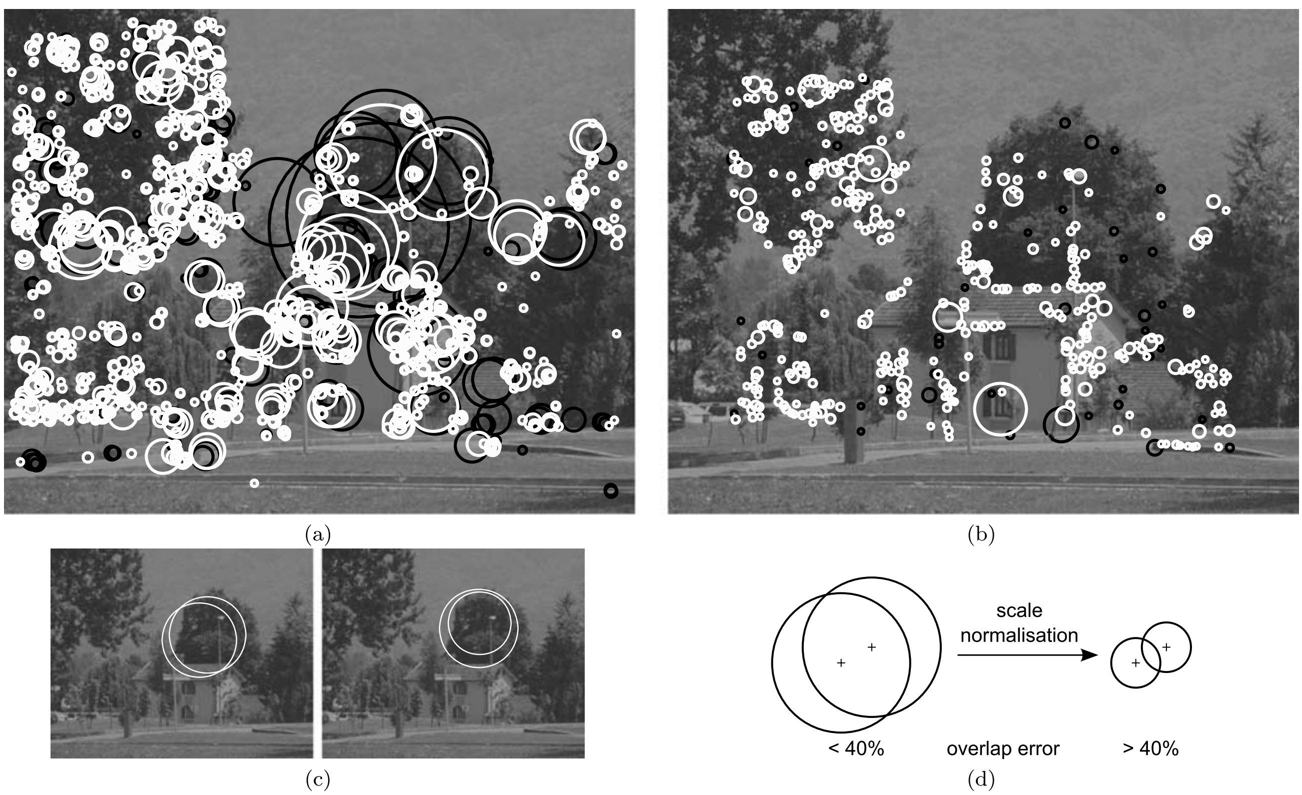


Fig. 8 Comparison of the outputs of the original and WaldBoost-emulated (**a**) Hessian-Laplace and (**b**) Kadir-Brady saliency detectors. The *white circles* show repeated Hessian-Laplace detection. The *black circles* highlight the original detections not found by the WaldBoost detector. Note that for most of missed detections there is a nearby detection on the same image structure. The accuracy of the emulation is 80% for Hessian-Laplace and 90% for Kadir-Brady saliency detector. Note that the publicly available Kadir-Brady algorithm does not detect points close to image edges. (**c**) Missed Hessian-Laplace detections (*left*) and manually found corresponding WaldBoost detections (*right*). (**d**) They are not found as correspondences, because Mikolajczyk's overlap function prefers smaller detections (see the discussion in the text)

WaldBoost emulator detections (with $\gamma = -\infty$). Since the comparison works on a single image, the ground truth transformation matrix is identity.

The white circles show the original detections with a correspondence found among the WaldBoost detections. The black circles show the original detections not found by

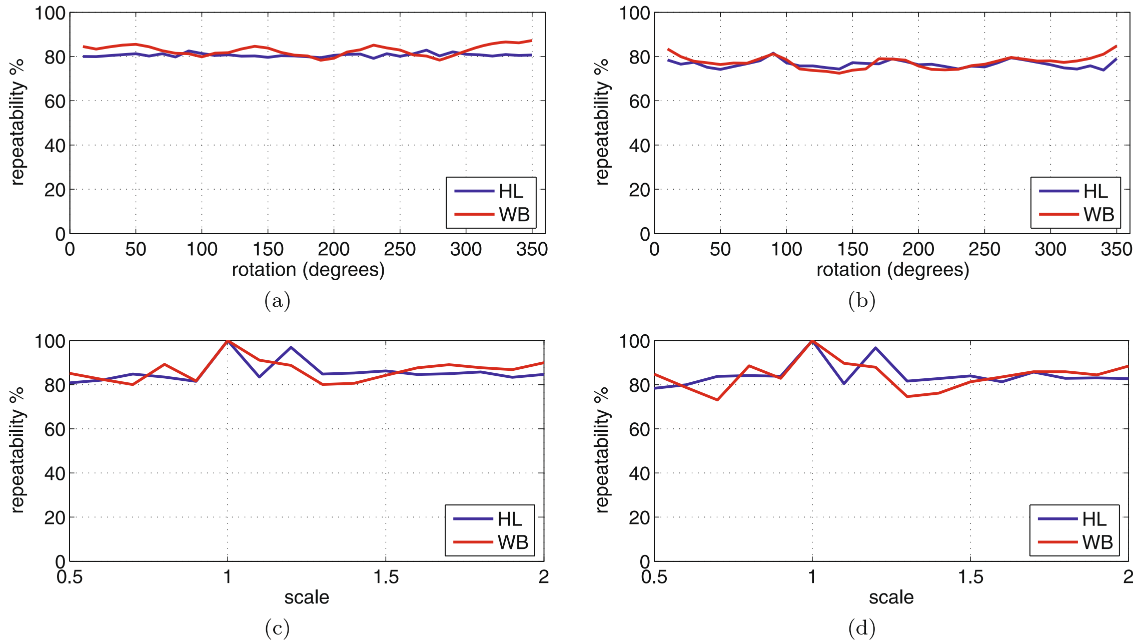


Fig. 9 Rotation and scale invariance of the WaldBoost Hessian-Laplace emulator. *Top row:* Repeatability on rotated first images from (**a**) BOAT, and (**b**) EAST_SOUTH sequences for the Hessian-Laplace detector (HL) and its WaldBoost emulator (WB). *Bottom row:* Repeatability on *scaled* first images from (**c**) BOAT, and (**d**) EAST_SOUTH sequences

Table 1 Speed comparison on the first image (850 × 680) from the BOAT sequence. The speed-up on another images is similar

	Hessian-Laplace	Kadir-Brady
Original	0.9 s	1 m 48 s
SURF	0.09 s	–
Speed-up	10×	–
$\bar{T}_{s^*}$	3	–
WaldBoost	0.10 s	0.76 s
Speed-up	9×	142×
$\bar{T}_{s^*}$	1.7	2.2

WaldBoost. Note that most of the missed detections have a correct detection nearby, so the corresponding image structure is actually found. The percentage of repeated detections of the original algorithm is 80%.

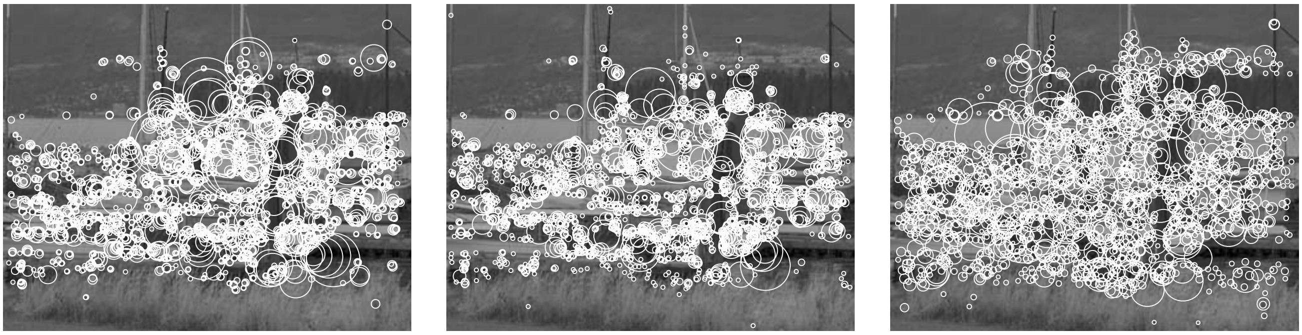
The WaldBoost detector may seem to miss consistently the large regions. Figure 8(c) shows manually selected WaldBoost regions close to the original detections—the "tree blob" is in fact detected. The real problem is in the way the correspondence overlap is computed. To compute the overlap of two detected points, Mikolajczyk et al. (2005) first normalise their scale to 30 pixels. This way, the problem of unnecessary large regions which would almost always have large overlaps is avoided. However, as shown in Fig. 8(d), this normalisation returns small overlap when large regions are only slightly misplaced. This problem is general and *appears in all region detection papers which use the Mikolajczyk's repeatability measure*. To conclude, the real emulation accuracy is in fact higher than 80%.

Rotational Invariance One of the properties of the emulated Hessian-Laplace detector which should be preserved is its rotational invariance. A learning approach can achieve rotational invariance with non-rotationally invariant features by introducing synthetically rotated positive samples into the training set. The results in Fig. 9 (top row) show that the rotational invariance is preserved even without introducing synthetic training samples. This is probably a consequence of the large training pool which is available. Instead of introducing rotated samples synthetically, the statistics are covered by collecting huge number of samples.

Scale Invariance Similarly, the detector invariance to scale changes has been tested. The emulated detector achieves similar scale invariance as the original algorithm as shown in Fig. 9 (bottom row).

Comparison to SURF The WaldBoost emulator has been compared with the SURF detector (Bay et al. 2008) which is a simplification of the Hessian-Laplace detector, manually designed for maximum speed. The SURF is commonly used as a good compromise between speed, accuracy and repeatability.

The comparison of the repeatability and the matching score of all three detectors is shown in Figs. 5 and 6. All the detectors has been set to produce similar number of de-



(a) (b) (c)

Fig. 10 Comparison of Hessian-Laplace (**a**), its WaldBoost emulator (**b**) and SURF detector (**c**) outputs on the first image from the BOAT sequence. WaldBoost returns similar distribution of points as the emulated Hessian-Laplace. The SURF points are distributed differently

tections on the first image of the EAST_SOUTH sequence. Neither of the fast detectors approximates the original detector perfectly. Yet, both could be said to achieve similar statistics as the original Hessian-Laplace detector, deviating slightly at different sequences.

The evaluation speeds of the detectors are compared in Table 1. The WaldBoost detector achieves similar evaluation speed as the manually tuned SURF detector. However, since most of the computational components are the same in both detectors, the average evaluation time $\bar{T}_{S*} = 1.7$ for WaldBoost and $\bar{T}_{S*} = 3$ for SURF suggests that further code optimisation of the WaldBoost detector could lead to even faster implementation.

An important difference between the SURF detector and the Hessian-Laplace WaldBoost emulator is that the first one is *a simplification* while the other is *an emulation*. The SURF produces different set of regions compared to the Hessian-Laplace detector. This could be verified by computing the coverage as in Fig. 8. For the SURF detector only 49.7% coverage is reached compared to 80% of the WaldBoost detector. The difference in detectors outputs is shown in Fig. 10.

Summary To conclude, the WaldBoost emulator of the Hessian-Laplace detector is able to detect points with similar repeatability and slightly higher matching score while keeping the rotational and scale invariance of the original detector. Moreover, the WaldBoost emulator was able to increase nine times the speed of detection compared to the original detector. When compared to the manually tuned SURF detector, similar repeatability, matching score and evaluation speed characteristics are reached. However the WaldBoost detector emulates the Hessian-Laplace detector significantly more closely.

5.2 Fast Saliency Detector

The emulation of the Kadir-Brady saliency detector (Kadir and Brady 2001) uses the same image pool for training as the WaldBoost Hessian-Laplace emulator. The saliency threshold of the original detector was set empirically to 2 to collect a sample pool of a reasonable size. Higher value of threshold also helps to limit the positive examples only to those with higher saliency. 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 since its response corresponds to the importance of the feature.

The Haar-like feature set was extended by the "variance" feature described in Sect. 4. The training was run for $T = 20$ (training steps) with $\alpha = 0.2$ and $\beta = 0$ as in the Hessian-Laplace experiment.

Publicly available version of Kadir-Brady detector has several drawbacks which need to be considered in the experimental evaluation. Due to relatively wide search for local maximum in the scale space, no detections near the image border are found. This results in a strip around image border where no detections are returned (see Fig. 8(b)). Also the scale range of detections is limited. In all following experiments, WaldBoost emulator detections are filtered by the same restrictions for the comparison reasons. However, the WaldBoost emulator of the Kadir-Brady detector *does not have these restrictions* inherently.

Repeatability and Matching Score 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 Fig. 11. Similar performance to the Kadir-Brady detector is reached for similar number of correspondences and correct matches on both sequences.

Speed The main advantage of the emulated saliency detector is its speed. The classifier evaluates on average 2.2 features per examined location and scale. Table 1 shows that the emulated detector is about 142× faster than the original detector.

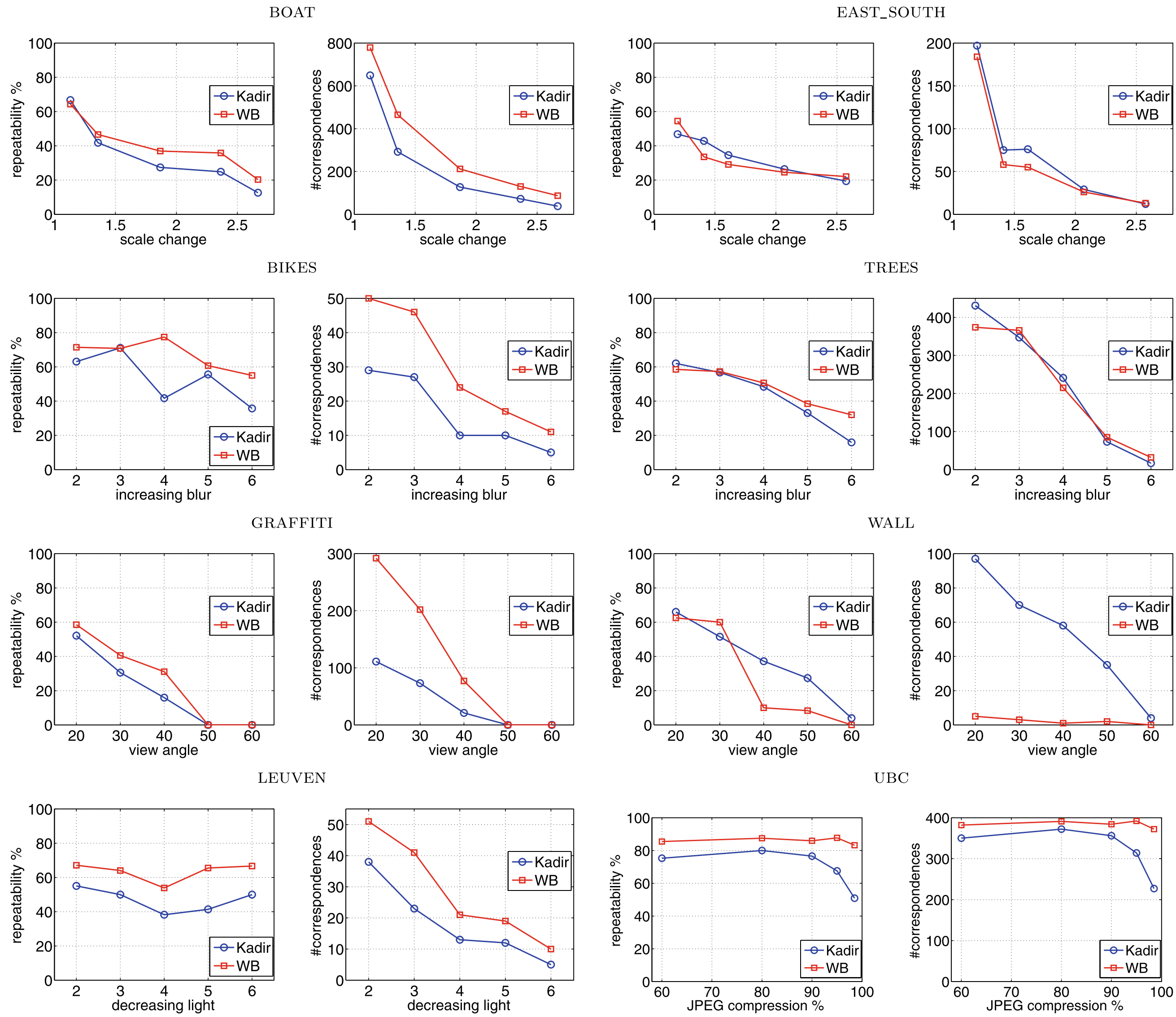


Fig. 11 Repeatability comparison of the Kadir-Brady detector and its WaldBoost emulation on Mikolajczyk's dataset

Classifier Structure Our early experiments showed that the Haar-like features are not suitable to emulate the entropy-based saliency detector. With the variance features, the training was able to converge to a reasonable classifier. In fact, the variance feature is chosen for the first weak classifier in the WaldBoost ensemble (see Fig. 7). The bin responses of the weak classifier show that higher variations are preferred.

Coverage The outputs of the WaldBoost saliency detector and the original algorithm are compared in Fig. 8(b). The coverage of original detections is 90%.

Rotational and Scale Invariance Invariance to rotation and scale changes of the WaldBoost emulator and the Kadir-Brady detector are compared in Fig. 13. Due to very different approaches in computing the detectors responses (Haar-like features vs. entropy), the WaldBoost emulator is not able to reach perfect rotation invariance on images rotated by 90 degrees but is able to keep similar rotational invariance otherwise. Moreover, the feature-based approach of the WaldBoost emulator results in slightly better scale invariance of the detector. This can be probably explained by the instability of the entropy based Kadir-Brady detector especially at small scales where the probabilities are difficult to estimate. It is shown also in Hare and Lewis (2004) that their difference-of-Gaussian detector is more robust to a range of transformations than the Kadir-Brady detector.

Summary To conclude, the WaldBoost training is able to emulate Kadir-Brady detector generally with similar repeatability, matching score and robustness to rotation changes, while improving slightly its scale invariance. But,

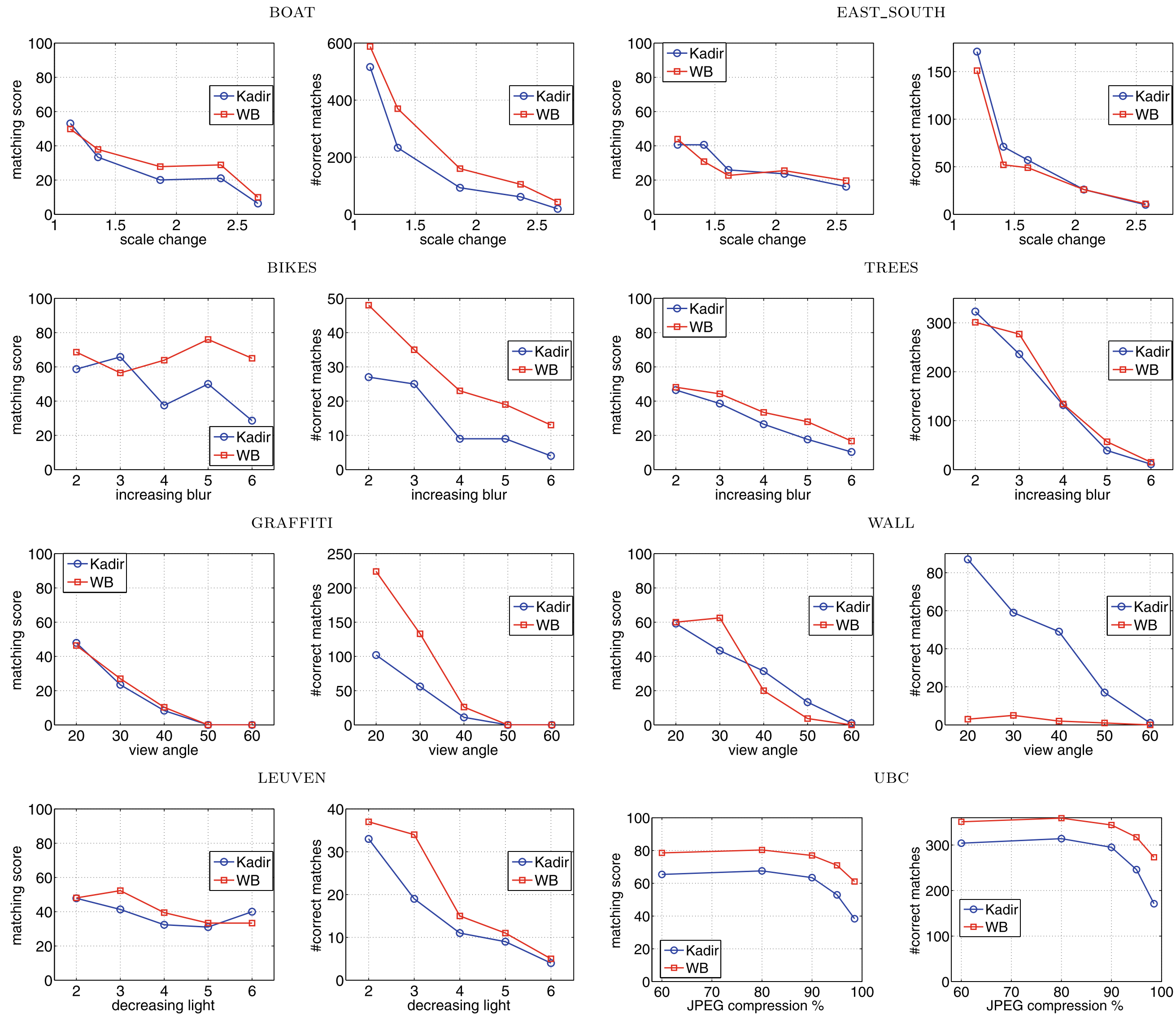


Fig. 12 Matching score comparison of the Kadir-Brady detector and its WaldBoost emulation on Mikolajczyk's dataset

most importantly, the decision times of the emulated detector are about 142 times lower than that of the original algorithm. That opens new possibilities for using the Kadir-Brady detector in time sensitive applications.

6 Conclusions and Future Work

In this paper, a general learning framework has been proposed for speeding up existing binary decision processes by a sequential classifier which is learnt by the WaldBoost algorithm.

Two interest point detectors, the Hessian-Laplace and the Kadir-Brady saliency detector, served as examples of emulated algorithms. The experiments show similar repeatability and matching scores of the original and emulating algorithms. For both, the Hessian-Laplace and the Kadir-Brady detectors, the WaldBoost emulation improved significantly the speed. The emulator was nine times faster for the Hessian-Laplace detector and about 142 times faster for the Kadir-Brady detector. In the case of the Kadir-Brady detector this speed-up opens new possibilities for using the detector in time sensitive applications. For the Hessian-Laplace detector, the achieved speed is similar to SURF, a commonly used Hessian-like fast detector; the WaldBoost emulator approximates the output of the Hessian-Laplace detector more precisely.

The proposed approach is general and can be applied to other algorithms as well. For future research, an interesting extension of the methodology would be to train an emulator which not only guarantees output similar to an existing algorithm but which also possesses some additional quality like

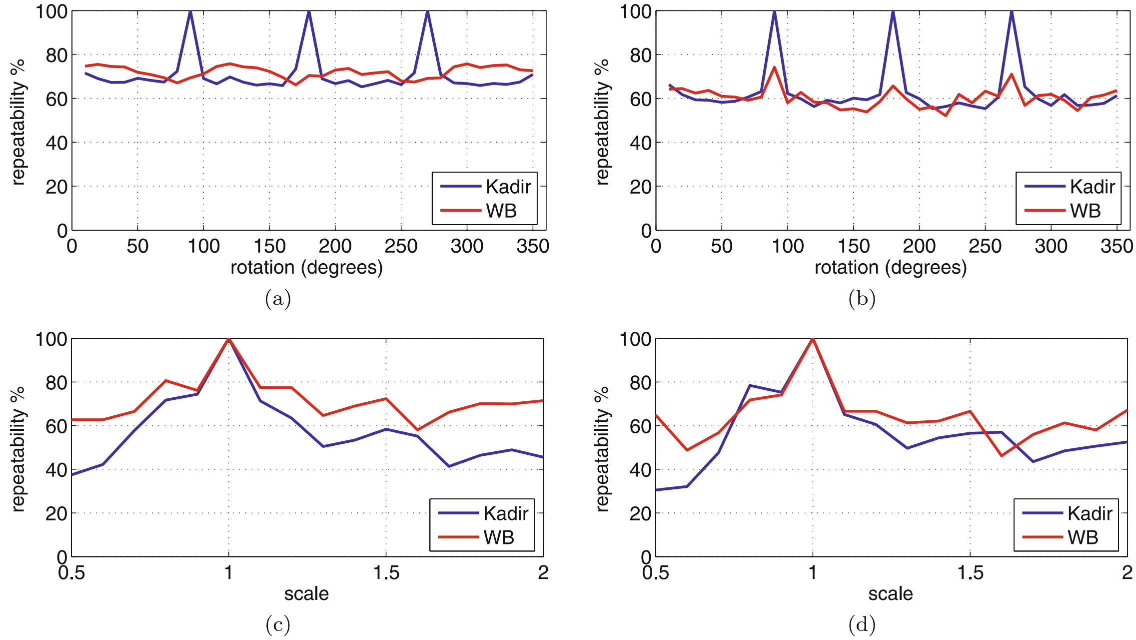


Fig. 13 Rotation and scale invariance of the WaldBoost Kadir-Brady emulator. *Top row:* Repeatability on *rotated* first images from (**a**) BOAT, and (**b**) EAST_SOUTH sequences for the Kadir-Brady detector (Kadir) and its WaldBoost emulator (WB). *Bottom row:* Repeatability on *scaled* first images from (**c**) BOAT, and (**d**) EAST_SOUTH sequences

insensitivity to certain acquisition conditions (e.g. motion blur) or maximum performance in a particular environment or task.

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

We thank the anonymous reviewers for their valuable comments that led to an improvement of the manuscript.

References

Baker, S., & Nayar, S. K. (1996). Algorithms for pattern rejection. In *International conference on pattern recognition* (Vol. 2, pp. 869–874).

Bay, H., Ess, A., Tuytelaars, T., & Gool, L. V. (2008). Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, *110*(3), 346–359.

Bourdev, L., & Brandt, J. (2005). Robust object detection via soft cascade. In *IEEE conference on computer vision and pattern recognition* (pp. 236–243). Washington: IEEE Computer Society.

Brubaker, S. C., Wu, J., Sun, J., Mullin, M. D., & Rehg, J. M. (2008). On the design of cascades of boosted ensembles for face detection. *International Journal of Computer Vision*, *77*(1–3), 65–86.

Dollár, P., Tu, Z., & Belongie, S. (2006). Supervised learning of edges and object boundaries. In *IEEE conference on computer vision and pattern recognition* (Vol. 2, pp. 1964–1971).

Fergus, R., Perona, P., & Zisserman, A. (2005). A sparse object category model for efficient learning and exhaustive recognition. In *IEEE conference on computer vision and pattern recognition* (Vol. 1, pp. 380–387).

Friedman, J., Hastie, T., & Tibshirani, R. (1998). *Additive logistic regression: a statistical view of boosting*. Technical report, Department of Statistics, Stanford University, Sequoia Hall.

Froba, B., & Ernst, A. (2004). Face detection with the modified census transform. In *International conference on automatic face and gesture recognition* (pp. 91–96).

Grabner, M., Grabner, H., & Bischof, H. (2006). Fast approximated SIFT. In *Asian conference on computer vision* (Vol. I, pp. 918–927).

Hare, J. S., & Lewis, P. H. (2004). Salient regions for query by image content. In: *Image and video retrieval: third international conference* (pp. 317–325).

Huang, C., Ai, H., Lao, S., & Li, Y. (2007). High-performance rotation invariant multiview face detection. *Pattern Analysis and Machine Intelligence*, *29*(4), 671–686.

Jones, M. J., & Rehg, J. M. (2002). Statistical color models with application to skin detection. *International Journal of Computer Vision*, *46*(1), 81–96.

Kadir, T., & Brady, M. (2001). Saliency, scale and image description. *International Journal of Computer Vision*, *45*(2), 83–105.

Kálal, Z., Matas, J., & Mikolajczyk, K. (2008). Weighted sampling for large-scale boosting. In *British machine vision conference*.

Lepetit, V., Lagger, P., & Fua, P. (2005). Randomized trees for real-time keypoint recognition. In *IEEE conference on computer vision and pattern recognition* (Vol. II, pp. 775–781).

Lienhart, R., & Maydt, J. (2002). An extended set of Haar-like features for rapid object detection. In *International conference on image processing* (Vol. 1, pp. 900–903).

Martin, D., Fowlkes, C., & Malik, J. (2004). Learning to detect natural image boundaries using local brightness, color, and texture cues. *Pattern Analysis and Machine Intelligence*, *26*(5), 530–549.

Matas, J., & Šochman, J. (2007). Wald's sequential analysis for time-constrained vision problems. In *International conference on robotics and automation*.

Mikolajczyk, K. (2002). *Detection of local features invariant to affine transformations*. Ph.D. thesis, INPG, Grenoble.

Mikolajczyk, K. (2008a). http://www.robots.ox.ac.uk/~vgg/research/affine.

Mikolajczyk, K. (2008b) Personal communication.

Mikolajczyk, K., & Schmid, C. (2004). Scale and affine invariant interest point detectors. *International Journal of Computer Vision*, *60*(1), 63–86.

Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., & Van Gool, L. (2005). A comparison of affine region detectors. *International Journal of Computer Vision*, *65*(1/2), 43–72.

Ojala, T., Pietikäinen, M., & Mäenpää, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Pattern Analysis and Machine Intelligence*, *24*(7), 971–987.

Rivest, R. L. (1987). Learning decision lists. In *Machine learning* (pp. 229–246).

Rosten, E., & Drummond, T. (2006). Machine learning for high-speed corner detection. In *European conference on computer vision* (Vol. 1, pp. 430–443).

Schapire, R. E., & Singer, Y. (1999). Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, *37*(3), 297–336.

Siegmund, D. (1985). *Sequential analysis. Test and confidence intervals*. *Springer series in statistics*. New York: Springer.

Šochman, J., & Matas, J. (2005). WaldBoost—learning for time constrained sequential detection. In *IEEE conference on computer vision and pattern recognition* (Vol. 2, pp. 150–157).

Šochman, J., & Matas, J. (2007). Learning a fast emulator of a binary decision process. In *Asian conference on computer vision* (Vol. II, pp. 236–245).

Sung, K. K., & Poggio, T. (1998). Example-based learning for view-based human face detection. *Pattern Analysis and Machine Intelligence*, *20*(1), 39–51.

Viola, P., & Jones, M. (2001). Robust real time object detection. In *International workshop on statistical and computational theories of vision*.

Wald, A. (1947). *Sequential analysis*. New York: Dover.

Xiao, R., Zhu, L., & Zhang, H. (2003). Boosting chain learning for object detection. In *International conference on computer vision* (pp. 709–715).

Zhu, L., Chen, Y., & Yuille, A. L. (2006). Unsupervised learning of a probabilistic grammar for object detection and parsing. In *Advances in neural information processing systems* (pp. 1617–1624).