Flea, Do You Remember Me?

Michael Grabner, Helmut Grabner, Joachim Pehserl, Petra Korica-Pehserl, and Horst Bischof

Institute for Computer Graphics and Vision
Graz University of Technology, Austria
{mgrabner,hgrabner,pehserl,korica,bischof}@icg.tugraz.at

Abstract. The ability to detect and recognize individuals is essential for an autonomous robot interacting with humans even if computational resources are usually rather limited. In general a small user group can be assumed for interaction. The robot has to distinguish between multiple users and further on between *known* and *unknown* persons. For solving this problem we propose an approach which integrates detection, recognition and tracking by formulating all tasks as binary classification problems. Because of its efficiency it is well suited for robots or other systems with limited resources but nevertheless demonstrates robustness and comparable results to state-of-the-art approaches. We use a common over-complete representation which is shared by the different modules. By means of the integral data structure an efficient feature computation is performed enabling the usage of this system for real-time applications such as for our autonomous robot *Flea*.

1 Introduction

Autonomous robots guiding blinds, cleaning dishes, delivering mail, laundering, entertaining and handling many other daily tasks belong to the future goals of a competition called RoboCup@Home[1]. The aim is to develop applications that can assist humans in everyday life. One specific task within this challenge is called *Who is Who?* and is thought to focus on enhancement of techniques for natural and social human-computer interaction. In specific, the detection and recognition of *known* vs. *unknown* individuals should enforce robots usability and make them automatically recognize familiar persons. Real-time capability is essential for interaction.

From the computer vision perspective (we do not consider the audio modality in this work) it requires three approaches to successfully handle these tasks, namely (1) Detection (2) Recognition and further on (3) Tracking can be optionally added. For these specific computer vision problems much research has been done and overviews of proposed techniques are given in [1,2,3]. Especially classification techniques have turned out to provide robust results for these tasks and are hard to compete in efficiency. For object detection the probably most widely

[1] www.robocupathome.org

Y. Yagi et al. (Eds.): ACCV 2007, Part I, LNCS 4843, pp. 657–666, 2007.

used technique is AdaBoost introduced by Viola and Jones [4]. For face recognition/identification several classification methods have been applied [5,6] and also tracking recently has been often solved by fromulating it as a classification problem between object and background [7,8,9].

However, only a few approaches exist considering the problem of detection, recognition and tracking as a common problem [10,11]. At most they are combined using consecutive stages and using totally independent techniques for the specific tasks (i.e. motion detection for tracker initialization). Related is also the work of Zisserman et al. [12,13] on face recognition in feature-length films. Their task is to perform person retrieval from movies, and they use face detection and tracking to perform that task. Their system demonstrates that by closely integrating the tasks an impressive performance can be obtained. However their approach is not applicable to our problem because of the high computational costs. To summarize, there does not exist an efficient combination of detection, recognition and tracking however for all specific tasks classification methods have been successfully applied.

In this paper we propose a system which integrates detection, identification and further on tracking. The approach is applied to faces however can be used as well for other objects. All tasks are formulated as binary classification problems allowing to apply well established learning techniques. An integral data structure is shared among all modules allowing very fast and efficient feature extraction. The system is especially suited for any device having limited resources. In specific, we apply the proposed approach to an autonomous robot.

The outline of the paper is as follows. In Section 2 we describe detection, recognition and tracking as binary classification problems. Furthermore the procedure how to share low-level computations among the modules as well as the used learning technique is presented. Section 3 shows experimental evaluations on a public database and in addition illustrative sample sequences captured from our mobile robot *Flea*. Section 4 concludes the paper and gives some outlook of ongoing work.

2 Identifying Familiar Persons and Unknowns

The identification of persons within images requires two steps. First, there is the need of a detection part which is responsible for locating all faces appearing in the image. Second, given the set of faces we want to distinguish between the class of *known persons* and *unknown persons* and further on between the individuals of known persons which is accomplished by the recognition step. The problem of identification is formulated in a coarse to fine manner by applying discriminative classification methods at both stages. In addition we are also interested in tracking of detected individuals since it allows our robot to track even if appearance changes (e.g. due to occlusion, view change) occur.

2.1 Detection, Recognition and Tracking as Binary Classification

The key idea of our approach is to formulate the abilities to detect, recognize and to track as classification problems, as depicted in Figure 1. By doing so we can apply the same techniques for all tasks. The major advantage is that low-level computations can be shared and have to be done only once. This will be explained in more detail in Section 2.2.

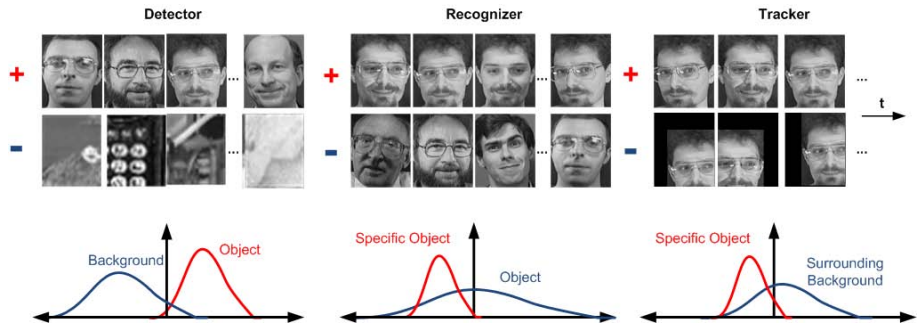


Fig. 1. Detecting, recognizing and tracking persons are considered as independent binary classification problems. A detector is trained off-line given a large set of positive labeled faces against non-face images. For recognition a specific face is trained against all other faces. For tracking an on-line classifier is used which allows continuously updating the model using the surrounding background as negative samples.

Detection. The task of the detector is to distinguish between the class of faces and the background. This can be formulated as a binary classification problem as proposed by Viola and Jones [14]. Given a training set $\mathcal{X}_d = \{\langle \mathbf{x}_{d,1}, y_{d,1} \rangle, \ldots, \langle \mathbf{x}_{d,n}, y_{d,n} \rangle\}$ where $\mathbf{x}_{d,i} \in \mathbb{R}^m$ is an image patch and the class labels $y_{d,i} \in \{+1, -1\}$ for faces and non-faces, respectively. Using this training set a binary classifier is trained by applying an off-line learning algorithm $\mathcal{L}_{off}$. Positive labeled samples (faces) are hand labeled and negative samples are sampled from an image database containing no faces. For evaluation, i.e. the detection of faces in an image, the classifier is applied in an exhaustive way searching over many image patches which are sampled at different locations and scales.

Recognition. Once a detection has been accomplished by the detector it is handed over to the recognizer module. In the first stage it classifies the provided sample as known or unknown and in the second stage verifies the identity in case it is a known face. This can be formulated as a multiclass classification problem. Given a set of samples $\mathcal{X}_r = \{\langle \mathbf{x}_{r,1}, y_{r,1} \rangle, \ldots, \langle \mathbf{x}_{r,n}, y_{r,n} \rangle\}$ where $\mathbf{x}_{r,i} \in \mathbb{R}^m$ represent face images and $y_{r,i} \in \{0, 1, 2, \ldots, M\}$ correspond to the labels each corresponding to one of the M different individuals and 0 to unknowns. This problem can be rewritten and formulated in an one vs. all manner which makes the usage of binary classifiers feasible. Meaning, for each person $j = 1, \ldots, M$

we train a single binary classifier against the other classes. Thus, the training set for the classifier C_j is $\mathcal{X}_{r,j} = \{\langle \mathbf{x}_{r,i}, +1\rangle | y_{r,i} = j\} \cup \{\langle \mathbf{x}_{r,i}, -1\rangle | y_{r,i} \neq j\}$. The classifier C_j is created by applying an off-line learning algorithm $\mathcal{L}_{off}$ on the training set. In fact, a model is learned which best discriminates the current person to the other given identities as shown in Figure 1.

In order to obtain robustness against the class unknown, we add arbitrary faces (e.g. from a face database used for training the face detector) as negative samples for the training. In the evaluation step, each classifier $C_j(\mathbf{x})$ evaluates the face image $\mathbf{x}$ and provides a confidence value. The classifier with the highest response delivers the class label $\hat{y}$.

$$\hat{y} = \arg\max_j C_j(\mathbf{x}); \quad j = 1, \ldots, M \tag{1}$$

The class unknown is recognized if all classifiers responses are below a certain threshold.

This approach can be extended by using an on-line learning algorithm $\mathcal{L}_{on}$ for updating the classifiers. It allows to add novel persons by just applying its samples as negative updates to existing classifiers and learning a new classifier as shown above.

Tracking. Tracking allows us to localize the object even if detection fails due to appearance change (e.g. occlusion). Further on it helps to get rid of possible false detections and to increase recognition accuracy.

Following the formulation in [9] we summarize the main steps of the tracking formulation as a binary classification problem. Once the target object has been detected at time t, it is assumed to be a positive image sample $\langle \mathbf{x}, +1\rangle_{t=0}$ for the classifier. At the same time negative examples $\{\langle \mathbf{x}_1, -1\rangle, \ldots, \langle \mathbf{x}_n, -1\rangle\}_{t=0}$ are extracted by taking regions of the same window size from the surrounding background. Given these examples an initial classifier $C_{t=0}$ is trained. The tracking step is based on the classical approach of template tracking. The current classifier C_t is evaluated at the surrounding region of interest and so obtain for each sub-patch a confidence value which implies how well the underlying image patch fits the current model. Afterwards we analyze the obtained confidence map and shift the target window to the new maxima location. Next, the classifier has to be updated in order to adjust to possible changes in appearance of the target object and to become discriminative to a different background. The current target region is used for a positive update of the classifier while surrounding regions again are taken as negative samples. This update policy has proved to allow stable tracking in natural scenes. As new frames arrive, the whole procedure is repeated and the classifier is therefore able to adapt to possible appearance changes and in addition becomes robust against background clutter.

Note, in order to formulate the tracking as a classification task, we need an on-line learning algorithm $\mathcal{L}_{on}$. The binary classifier updates the model (decision boundary) by using the information from a single new sample $\langle \mathbf{x}, y\rangle$, $\mathbf{x} \in \mathbb{R}^m$ and $y \in \{+1, -1\}$.

2.2 Efficient Features and a Single Shared Data Structure

An overview of the proposed system is given in Figure 2. For each frame the integral representation needs to be computed only once which is then used by all three modules for feature computation. Note that each unit selects appropriate features for the specific task however computation time of the features is negligible.

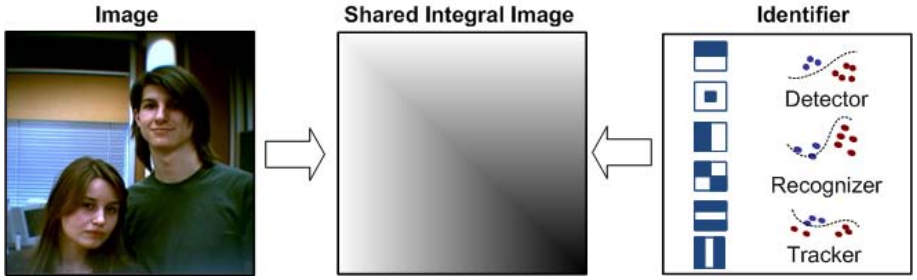


Fig. 2. Each module (detector, tracker and recognizer) is based on the same classification method, allowing the use of same feature types (Haar-like wavelets). These features can be computed very efficiently using a shared integral data structure.

For binary classification of image patches we propose to use the classical approach from Viola and Jones [14]. There main assumption is that a small set of simple image features can separate two classes. The selection of the features is done by a machine learning algorithm. In the following we briefly summarize the applied techniques.

Features. As features we use simple Haar-like features[2]. We spend some time on pre-computation of the efficient data structure, namely the integral image, which can be used for fast feature evaluation. This pre-computation has to be done only once since all three modules share this information.

Boosting for Feature Selection. For training a classifier we apply boosting for feature selection [17,14]. Core of the technique is the machine learning algorithm AdaBoost [18]. Given a set of training samples $\mathcal{X} = \{\langle \mathbf{x_1}, y_1 \rangle, \ldots, \langle \mathbf{x_n}, y_n \rangle\}$, $\mathbf{x}_i \in \mathbb{R}^m$ and $y_i \in \{-1, +1\}$ boosting builds an additive model of weak classifiers in the training stage. At each iteration a weak classifier is trained using a weight distribution over the training samples. In order to perform feature selection, a weak classifier corresponds to a simple image feature. Afterwards a re-weighting of the samples is done. The result is a strong classifier

[2] Note, also other kind of features like edge orientation histograms can be build using integral data structures [15] or Local Binary Patterns [16].

$$H(\mathbf{x}) = \text{sign}(conf(\mathbf{x}))$$
$$conf(\mathbf{x}) = \frac{1}{\sum_{i=1}^{N} \alpha_i} \sum_{i=1}^{N} \alpha_i \cdot h_i(\mathbf{x}) \tag{2}$$

which consists of a weighted linear combination of N weak classifiers h_i. The value $conf(\mathbf{x})$ bounded in the interval $[-1, +1]$, denotes how confident the classifier is about its decision. This fulfills the requirements of the classifier C from the previous section.

Boosting and especially boosting for feature selection as described above runs off-line, meaning all the training data is given at once. Primarily for tracking we need an on-line learning algorithm. For on-line adaption of the classifier we make use of an on-line version [19]. The key idea is to introduce so called *selectors* which hold a set of weak classifiers and each selector can chose exactly one of them. An on-line boosting algorithm [20] is performed on the selectors and not on the weak classifiers directly. Updating can be done efficiently. After updating the classifier the evaluation is similar to the off-line case, because the selector has chosen one specific weak classifier which again corresponds to a single feature.

3 Results

First we introduce our autonomous robot and give some relevant details regarding the hardware setup. We present a performance evaluation of our proposed system focusing on recognition accuracy as well as considering the ability to distinguish between known and unknown individuals. Finally an illustrative experiment is shown which is also available at `www.flea.at`.

3.1 Robot *Flea*

The used hardware setup consists of an ActiveMedia Peoplebot platform including diverse sensors (e.g. sonar, IR). The robot's head has thirteen degrees of freedom and can move its eyes, mouth, eyebrows, forehead, chin and neck. A Dual Core Centrino with 2 GHz and 1024 MB RAM represents the main

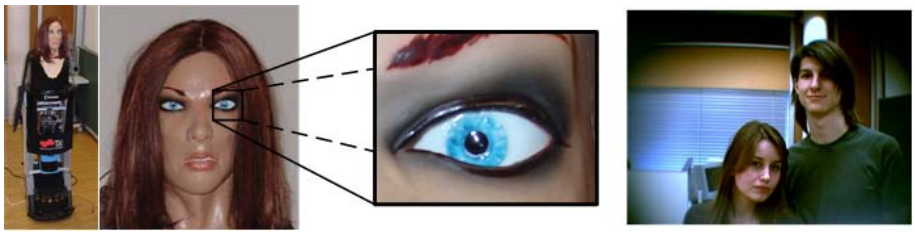
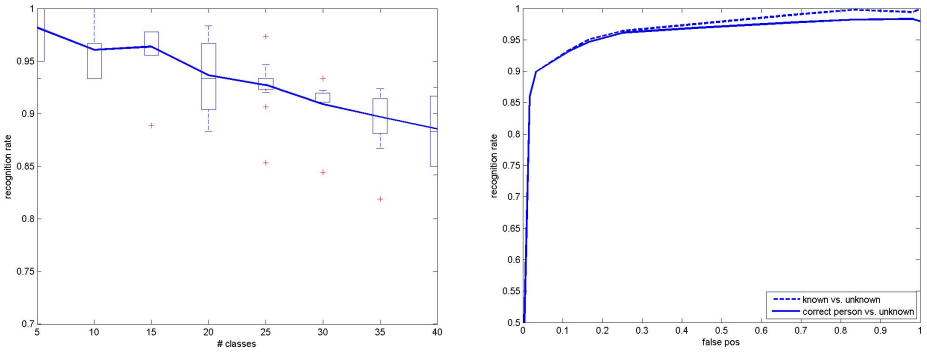


Fig. 3. Our robot *Flea* consists of a humanoid head. Visual information is obtained through a camera which is included in the artificial eye. A captured image from the view of the robot is depicted on the right.



(a) Scalability (b) Performance

Fig. 4. Performance evaluation of the recognition system. (a) shows the recognition rate when increasing the number of persons, (b) depicts the trade off of recognizing persons vs. unknowns.

processing unit. A stereo camera from Videre Design STH-MDCS2-VAR (max. 1280×960 used: 640×480) is used for capturing images and about 12 frames per seconds are processed with a non optimized C++ implementation.

Training of the robot is done in a fully autonomous way. In case Flea meets an unknown person, she focuses on it and asks for the name and other relevant information. During the conversation it starts collecting training samples of the person and trains a classifier for identification. When meeting Flea somewhere and asking the robot *Flea, do you remember me?*, she will reply *Sure,...* adding your name in case she knows you and otherwise asking you for your name. This is exactly the task that has to be fulfilled within the *Who is Who?* competition from RoboCup@Home.

3.2 Recognition Performance

For evaluation of the recognition accuracy and in specific the recognition of class unknown we use the AT&T database[3] which includes 40 different persons with 10 samples per class. This dataset is well suited since in our application it is not important to distinguish a huge number of individuals however it is important to accurately recognize the class of unknown individuals.

In the first experiment we want to demonstrate recognition accuracy with respect of the number of classes. For training the dataset has been randomly split into training and test set (70% trainig and 30% testing). The result, depicted in Figure 4, has been obtained by running the experiment 10 times. The second evaluation illustrates on the one hand the performance of distinguishing between unknown and known faces and on the other hand shows also the trade off of recognizing the correct class in case of a known person. For training we use the same training procedure as in the previous evaluation. We randomly selected 20

[3] `www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html`

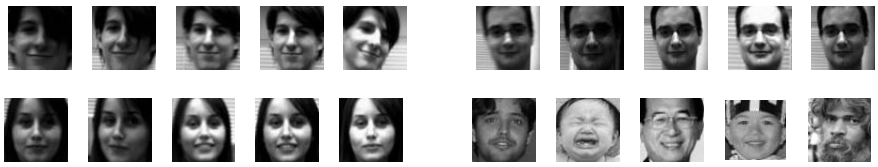
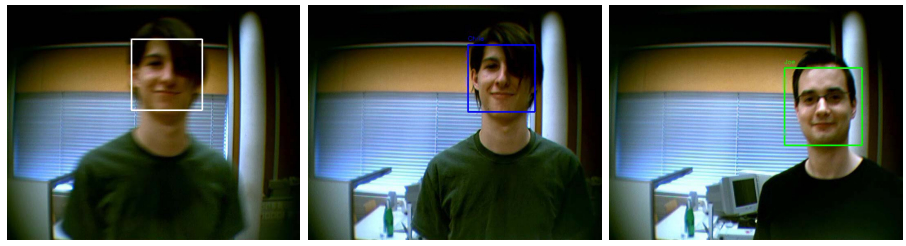
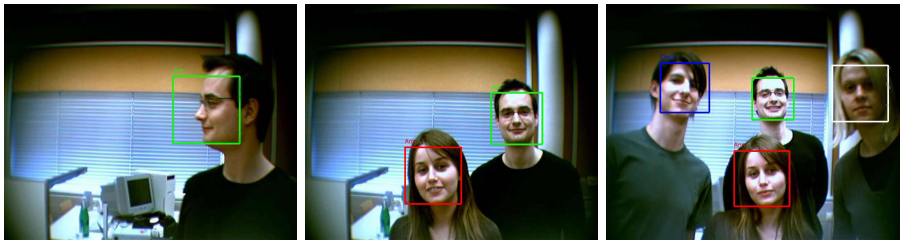


Fig. 5. Subset of training samples for three different persons (from upper left to bottom right: Chris, Joe, Ann) and samples taken from the training database (bottom right set) as additional negative samples



(a) Frame 215 (b) Frame 220 (c) Frame 355



(d) Frame 357 (e) Frame 431 (f) Frame 839

Fig. 6. Sample sequence from the perspective of *Flea*. Learned faces are detected, robustly tracked (d) and correctly identified if they are known (e-f). Different individuals are marked by different colored rectangles. The approach is running on the robot with about 12 frames per second.

individuals from the dataset and applied cross-validation to achieve statistically significant results.

As depicted in Figure 4 the overall performance for distinguishing between unknown and known faces is handled in a proper way. The difference in performance of recognizing the correct identity compared to recognizing just the class known is marginal.

3.3 Sequences

We also want to demonstrate a typical *Who is who?* scenario. Three persons introduce themselves to the robot whereas the robot collects samples of each

individual as depicted in Figure 5. Training each individual is done within a few seconds including the capturing of the faces.

Figure 6 illustrates the applied system on a sequence of our autonomous robot. As can be seen, the approach handles the recognition of known and unknown faces and further on shows the benefit of combining detection, recognition and tracking.

4 Conclusion

We have combined detection, recognition and tracking by formulating all tasks as binary classification problems. As a result low-level computations can be shared among all modules. Due to efficient feature computation the approach can be used within real-time applications such as autonomous robots. Note that the approach is not limited to faces since all modules are generic and therefore the proposed approach can be applied to any other type of object. The common formulation opens several new venues such as improving (specializing) detectors and recognizers for images taken from a static camera as it is the case in video surveillance applications.

Acknowledgement

This work has been sponsored by the Austrian Joint Research Project Cognitive Vision under projects S9103-N04 and S9104-N04, the EC funded NOE MUSCLE IST 507572.

References

1. Yang, M.H., Kriegman, D.J., Ahuja, N.: Detecting faces in images: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(1), 34–58 (2002)
2. Tan, X., Chen, S., Zhou, Z.H., Zhang, F.: Face recognition from a single image per person: A survey. Pattern Recognition 39(9), 1725–1745 (2006)
3. Yilmaz, A., Javed, O., Shah, M.: Object tracking: A survey. ACM Computing Surveys 38(4) (2006)
4. Viola, P., Jones, M.: Robust real-time object detection. International Journal of Computer Vision (2002)
5. Jonsson, K., Kittler, J., Li, Y.P., Matas, J.: Learning support vectors for face verification and recognition. In: Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition, pp. 208–213. IEEE Computer Society Press, Los Alamitos (2000)
6. Yang, P., Shan, S., Gao, W., Li, S.: Face recognition using Ada-boosted Gabor features. In: Proceedings Conference on Automatic Face and Gesture Recognition, pp. 356–361 (2004)
7. Avidan, S.: Ensemble tracking. In: Proceedings IEEE Conference Computer Vision and Pattern Recognition, vol. 2, pp. 494–501 (2005)
8. Avidan, S.: Support vector tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence 26, 1064–1072 (2004)

9. Grabner, H., Grabner, M., Bischof, H.: Real-time tracking via on-line boosting. In: Proceedings British Machine Vision Conference, vol. 1, pp. 47–56 (2006)
10. Ebbecke, M., Ali, M., Dengel, A.: Real time object detection, tracking and classification in monocular image sequences of road traffic scenes. In: Proceedings International Conference on Image Processing, vol. 2, pp. 402–405 (1997)
11. Hernández, M., Cabrera, J., Dominguez, A., Santana, M.C., Guerra, C., Hernández, D., Isern, J.: Deseo: An active vision system for detection, tracking and recognition, pp. 376–391 (1999)
12. Arandjelovic, O., Zisserman, A.: Automatic face recognition for film character retrieval in feature-length films. In: Proceedings IEEE Conference Computer Vision and Pattern Recognition, vol. 1, pp. 860–867 (2005)
13. Sivic, J., Everingham, M., Zisserman, A.: Person spotting: Video shot retrieval for face sets. In: Proceedings International Conference on Image and Video Retrieval, pp. 226–236 (2005)
14. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proceedings IEEE Conference Computer Vision and Pattern Recognition. vol. 1, pp. 511–518 (2001)
15. Porikli, F.: Integral histogram: A fast way to extract histograms in cartesian spaces. In: Proceedings IEEE Conference Computer Vision and Pattern Recognition, vol. 1, pp. 829–836 (2005)
16. Ojala, T., Pietikäinen, M., Mäenpää, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(7), 971–987 (2002)
17. Tieu, K., Viola, P.: Boosting image retrieval. In: Proceedings IEEE Conference Computer Vision and Pattern Recognition, vol. 1, pp. 228–235 (2000)
18. Freund, Y., Schapire, R.: A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences 55(1), 119–139 (1997)
19. Grabner, H., Bischof, H.: On-line boosting and vision. In: Proceedings IEEE Conference Computer Vision and Pattern Recognition. vol. 1, pp. 260–267 (2006)
20. Oza, N., Russell, S.: Online bagging and boosting. In: Proceedings Artificial Intelligence and Statistics, pp. 105–112 (2001)