



HAL
open science

Dynamic ensemble for target tracking

Ingrid Visentini, Lauro Snidaro, Gian Luca Foresti

► **To cite this version:**

Ingrid Visentini, Lauro Snidaro, Gian Luca Foresti. Dynamic ensemble for target tracking. The Eighth International Workshop on Visual Surveillance - VS2008, Graeme Jones and Tieniu Tan and Steve Maybank and Dimitrios Makris, Oct 2008, Marseille, France. inria-00325617

HAL Id: inria-00325617

<https://inria.hal.science/inria-00325617>

Submitted on 29 Sep 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Dynamic ensemble for target tracking

Ingrid Visentini, Lauro Snidaro, Gian Luca Foresti
University of Udine

Department of Mathematics and Computer Science

Via delle Scienze 206, 33100 Udine, Italy

{ingrid.visentini, lauro.snidaro, gianluca.foresti}@dimi.uniud.it

Abstract

On-line boosting is a recent breakthrough in the machine learning literature that has opened new possibilities in many diverse fields. Instead of generating a static strong classifier off-line, the classifier can be built on-the-fly on incoming samples. This has been successfully exploited in treating computer vision tasks such as tracking as a classification problem thus providing an intriguing new perspective to an old subject. Known solutions to the on-line boosting problem rely on a fixed number of weak classifiers. The first main contribution of this paper removes this limitation and shows how a dynamic ensemble can better address the tracking problem by providing increased robustness. The second proposed novelty consists in a mechanism for detecting scale changes of tracked targets. Promising results are shown on publicly available and our own video sequences.

1 Introduction

Boosting is a well-known method that has been used for a wide spectrum of machine learning tasks. Its robustness and incredible variety of possible applications have made the fortune of this technique. In particular, the Adaboost meta-algorithm has attracted much attention in the field of computer vision after the work of Viola and Jones on the face detection problem (see [17] for a recent tractation).

After almost a decade of research on boosting, which has generated a number of variations of the original off-line algorithm, a new strong impulse to the field of ensemble classifiers was given by the work of Oza [13] where he conceived on-line versions of the bagging and boosting algorithms, and demonstrated their equivalence to their off-line counterparts. Therefore, he formulated a general framework for real-time updating en-

semble classifiers. As shown in the recent work of Grabner and Bischof [7], on-line boosting can be successfully applied to real-time computer vision tasks such as target detection, background updating and tracking. In particular, they continued on the way paved by Avidan (see [3] for a recent tractation) who proposed tracking as a classification problem.

In real world scenarios the appearance of a moving object, as it is captured by a video sensor, is very likely to vary due to changes in perspective, shape, illumination, and the possible occurrence of occlusions. Combining multiple features is a winning strategy in terms of achieved robustness and accuracy [23], however it can prohibitively increase the computational burden to the point of preventing real-time performance. Therefore, it would be more attractive to dynamically select a reduced set of the most expressive features to describe the object of interest for each time instant. This problem is called feature selection and has recently attracted the attention of the researchers in the field of computer vision. As clearly stated in [5], the problem of subset feature selection is to find m features that best complement each other for the current classification task. In particular, the tasks of detecting and tracking an object largely depends on how distinguishable it is from the background. Therefore, among the multitude of features that could be employed to describe an object, the subset including those that best discriminate between object and background should be chosen as the most representative and robust.

In the work of Grabner and Bischof [7] this technique is shown to be real-time and robust to occlusions. However, their implementation uses a fixed number of weak classifiers to form a strong classifier via an architecture that selects, from a limited pool, the best features to describe the object at each time instant. A fixed number of features to be chosen among a limited pool eases the implementation of an on-line algorithm for feature selection, however this could be limiting for describing

a target that is subject to scale and shape variations.

Also in [14] the scale problem is not considered. In this case, no classifiers are discarded from the ensemble when the error is greater than random guessing, but the internal parameters of the boosted weak classifiers are tuned instead.

All these works track a fixed-size object. Small variations in the shape or in the dimension of the target are tolerated, but not explicitly addressed. In [20] a dynamic variation of the ensemble cardinality that supports multiscale tracking is proposed, and the ensemble is rearranged in a classifiers cascade fashion. However, the cascade structure is inherited from a batch boosted step, and the novelty consists in changing the threshold for each level of the cascade according with the error of the features belonging to the level. Moreover, the authors add more base classifiers to the end of the cascade, that is the last level.

In this paper, we propose a dynamic ensemble to address the problem of tracking as classification. The cardinality of the ensemble can automatically grow to address scale changes and to respond to difficult conditions such as occlusions and shape transformations (Section 3.1). The number of classifiers does not grow indefinitely, as the ensemble is automatically reduced to save computational requirements when no particular condition occurs and the ensemble is “overqualified”. We also dynamically organize the classifiers in a cascade to speed-up the probing phase in each frame (Section 3.2). This computational saving allows us to apply the classifier at different dimensions on the image to detect possible scale changes of tracked objects (the mechanism is proposed in Section 3.3). The pool of possible features is updated depending on the target size; this allows to keep tracking the target as it undergoes shape and scale changes. Finally, the presented work is suited for fixed and mobile camera, and it does not require any previous knowledge of the target.

2 Background

Tracking is a key processing step for many computer vision applications (see [22] for an extensive account) in which moving objects have to be located in consecutive video frames. The task is not an easy one since sensor noise, illumination changes, and occlusions are typical issues that have to be dealt with to keep each target associated with its label.

An interesting new approach is to treat tracking as a classification problem as proposed by Avidan in [2] using support vector machines. A binary classifier is trained to distinguish a given object in the scene from the background. At each time instant t the classifier

locates the most probable position in the image of the model developed up to the previous time instant $t - 1$. The region located at time t is used to update the classifier and the process is repeated. This approach is particularly appealing as the classifier is continuously updated by learning the current appearance of the target.

Real-time performance is a quality always appreciated in tracking algorithms. This is quality becomes a particularly stringent requirement for applications such as video-surveillance [9]. Boosting was already an attractive technique for off-line classification as demonstrated by the vast literature on it. Recently, the seminal work of Oza [13] gave further thrust to this topic by proposing an on-line and fast implementation that has opened a whole new world of possibilities.

In the following sections, after a brief account of the key concepts of boosting, we address some of the issues of the current implementations of on-line boosting for tracking.

2.1 Boosting

The aim of boosting is to combine many weak (base) classifiers in order to form an ensemble whose performance is better than that of the base classifiers [6]. The major concepts and the biggest difference between on-line and off-line boosting are here briefly explained.

2.1.1 Off-line boosting

The off-line boosting technique is employed to generate from an overcomplete set of features a final classifier of the form

$$H(x) = \text{sign} \left(\sum_{t=1}^T \alpha_t h_t(x) \right) \quad (1)$$

where each hypothesis h_t is slightly better than random guessing and α_t are the coefficients used by linear combination. At each epoch t , the choice of the best classifier h_t depends on its training error ϵ_t with respect to the probability distribution D_t on the learning samples \mathbf{x} and their correspondent labels \mathbf{y} , such that

$$\epsilon_t = Pr_{i \sim D_t} [h_t(x_i) \neq y_i] = \sum_{i: h_t(x_i) \neq y_i} D_t(i) < \Theta \quad (2)$$

In this way, the best hypothesis is chosen and added to the strong classifier in a greedy fashion according to the error minimization.

2.1.2 On-line boosting

The on-line boosting technique, firstly introduced by Oza [13], is used to model a new strong classifier or to

update a previously built one, in order to adapt the ensemble to better recognize the target. The strong classifier becomes

$$H(x) = \text{sign} \left(\sum_{m=1}^M \log \left(\frac{1 - \epsilon_m}{\epsilon_m} \right) h_m(x) \right) \quad (3)$$

where M is the number of weak classifiers fixed a priori. In this case, the training samples are processed one-by-one and, since no distribution on the samples is available, a distribution on the weak classifiers is maintained at each step. It is proved that for a number of training samples growing to infinity (or for $M \rightarrow \infty$) the on-line and the batch boosting algorithms converge to the same classifier [13].

The learning parameter λ is used to estimate the importance of the sample that is propagated through the set of hypotheses. It modifies the correctly recognized samples weight λ_m^{sc} and the misunderstood samples weight λ_m^{sw} , determining the accuracy of each hypotheses h_m in terms of its error

$$\epsilon_m = \frac{\lambda_m^{sw}}{\lambda_m^{sc} + \lambda_m^{sw}} \quad (4)$$

Notice that, while in the off-line method a distribution of the samples is always available to all classifiers, in on-line boosting it is approximated by the learning weights for each selected classifier.

In our case, we follow the approach suggested in [15], where the on-line approach is modified to balance the skewness of the labels presented to each of the weak classifiers. The skewness between the (two) classes corresponds to a different importance weight that is involved in the boosting process. As in [19] a parameter k penalizes false negatives with respect to false positives. The error of a single classifier becomes

$$\epsilon_m \approx \frac{\lambda_m^{fn} \sqrt{k_m} + \frac{\lambda_m^{fp}}{\sqrt{k_m}}}{(\lambda_m^{tp} + \lambda_m^{fn}) \sqrt{k_m} + \frac{(\lambda_m^{fp} + \lambda_m^{tn})}{\sqrt{k_m}}} \quad (5)$$

where λ_m^{tp} , λ_m^{tn} , λ_m^{fp} , λ_m^{fn} are, respectively, the weights of true positive, true negative, false positive and false negative examples.

In the work of Grabner and Bischof [7] and Grabner *et al.* [8], an architecture based on boosted selectors is presented. A selector is defined as a function that returns the best classifier among a limited pool of base classifiers $\mathcal{H}^{weak} = \{h_1^{weak}, \dots, h_M^{weak}\}$, that is, $h^{sel}(\mathbf{x}) = h_m^{weak}(\mathbf{x})$ where m is chosen according an optimization criterion such as minimum error e_i among the classifiers so that $m = \arg \min_i e_i$. A fixed number of selectors (50) is mentioned in the experiments, each of them picking a classifier out of a set of 250.

A fixed number of selectors, and most of all, a limited number of features actually probed on each frame are a guarantee for real-time performance. However, such an architecture artificially constraints the classifier as the number of selectors and the cardinality of the pools must be chosen a priori.

In this paper, we start again from the on-line boosting algorithm invented by Oza [13], and we give our own adaptation for the task of feature selection. As it will be explained in the following sections, in this work, both the number of classifiers forming the ensemble and the cardinality of the pool are dynamic to better adapt the ensemble to the tracking problem and to better support scale and shape changes of the targets.

3 Proposed solution

A previous batch step that generates a finite classifier is a good way to exploit the size M of the final strong classifier, but this is not always suitable, i.e., in the case of the tracking of an unknown object. Moreover, the fixed number of features M can be limiting for describing the object or, on the contrary, the ensemble could be overqualified and, thus, waste computational resources.

The main idea presented in this paper is to build and update the set of weak classifiers \mathcal{H}^{weak} depending on value of the ensemble error ϵ_H , and, then, to construct a fast classifiers cascade similar to [18]. This process allows to save time that can be used in new evaluations at multiple scales in multiple locations for an exhaustive search.

3.1 Dynamic ensemble

At each frame t , after finishing the on-line learning procedure, erroneous base classifiers can be removed from the pool \mathcal{H}^{weak} . The weak classifiers with error greater than random guessing (0.5) are taken out of the set. The remaining K contribute to calculate the ensemble error ϵ_H

$$\epsilon_H = \frac{1}{K} \sum_{k:\epsilon_k < 0.5} \epsilon_k \quad (6)$$

The weak learners not contributing to Eq. (6) are replaced in the classifiers pool \mathcal{H}^{weak} at time $t + 1$ with new hypotheses. This procedure helps to regenerate the diversity of the weak classifiers set, and, then, of the strong classifier. When the target changes size, the pool is modified to include the new possible features or to discard the features that are not compatible with the new dimensions.

The ensemble error ϵ_H is useful to understand the difficulty of the sample in relation to the ensemble. If this measure grows, the ensemble could have difficulties in recognizing the target and more (or different) classifiers are required. Otherwise, when the error decreases, the ensemble could be redundant and, thus, some features could be discarded. This last part can be easily done by removing weak learners according to a predefined policy. For example, one could choose to randomly remove features so that $\Theta_{low} < \epsilon_H < \Theta_{high}$.

The process of adding features is more complex as it requires an update-and-train step. The proposed framework considers a threshold Θ_{high} for the ensemble error. When it is crossed, a random feature is drawn from the abstract pool and the corresponding base classifier h_r is trained on the previously collected samples. If $\epsilon_r < 0.5$ holds, it is added to the ensemble. The loop is repeated until the Eq. (6) is satisfied. The entire process is described by Algorithm 1.

3.2 Classifiers cascade

In [18] Viola and Jones proposed to build a cascaded classifier to speed-up the application of the ensemble. They exploited the fact described in [1] that more accurate detectors produce the narrowing of the focus of attention on specific regions of the image where the object is likely to be present. In [18, 4, 21], the cascade was trained on a bulky amount of previously collected data, with a so called ‘‘off-line’’ learning procedure. Our idea, detailed in [16], is to build on-the-fly at every round a fast classifiers structure, without inheriting any structure, and without requiring any previous knowledge of the target class.

3.3 Scale changes

The cascade process reduces the time of application of the classifier on the region of interest \mathbf{x} . The time saving cascade allows to evaluate the detector at n different scales at different locations to test if the target has increased or decreased its size. At each frame, the strong classifier H generates a matrix of confidence values Map^{sc} for each search scale sc . The values of Map^{sc} represent the level of confidence of the classifier in estimating the presence of the searched target for each possible location.

The most probable location in Map^{sc} is the pixel with the maximum confidence. Let p be a generic pixel of Map_H^{sc} , then the most probable location is given by

$$\arg \max_{sc} \left(\arg \max_p (Map^{sc}(p)) \right) \quad (7)$$

Algorithm 1: Dynamic online boosted ensemble

Require: Strong classifier H previously initialized
Require: Base learners set \mathcal{H}^{weak}
Require: Base learner model L_0

```

for  $t = 1, 2, \dots, T$  do
  // Apply classifiers cascade
   $s \leftarrow 0$  // Number of samples
  for every subwindow  $x_{t,k}$  in  $Frame_t$  do
    if  $H_t(x_{t,k}) > \max(H_t(x_t))$  then
       $y_{t,k} \leftarrow 1$  // Positive sample
    else
       $y_{t,k} \leftarrow -1$  // Negative sample
    end if
     $s \leftarrow s + 1$ 
  end for
  for every discarded learner at  $t - 1$  do
     $\mathcal{H}^{weak} \leftarrow h_{random}$  // Add random base learners
  end for
  // Train the set  $\mathcal{H}^{weak}$ 
  for each training sample  $(x_{t,k}, y_{t,k}), k = 1, \dots, s$  do
     $\lambda \leftarrow 1$ 
    OnlineBoosting( $\mathcal{H}^{weak}, L_0, (x_{t,k}, y_{t,k}), \lambda$ ) [15]
  end for
  // Remove erroneous hypotheses and get ensemble error
   $\epsilon_H \leftarrow 1$ 
  for  $h_m \in \mathcal{H}^{weak}, m = 1, 2, \dots, M$  do
    if  $\epsilon_m \geq 0.5$  then
      remove  $\epsilon_m$ 
    else
       $\epsilon_H \leftarrow \epsilon_H + \epsilon_m$ 
    end if
  end for
  /* While the global error is greater than expected, train on-the-fly new hypotheses */
  while  $\epsilon_H \geq \Theta_{high}$  do
    // Train random feature  $h_r$ 
    OnlineBoosting( $h_r, L_0, (x_{t,k}, y_{t,k}), \lambda$ ) [15]
    if  $\epsilon_r \leq 0.5$  then
       $\mathcal{H}^{weak} \leftarrow h_r; \epsilon_H \leftarrow \epsilon_H + \epsilon_r$ 
    end if
  end while
  // Build the cascade with the learners in  $\mathcal{H}^{weak}$ 
  for  $l = 1, 2, \dots, L$  do
    for  $h_m \in \mathcal{H}^{weak}, m = 1, 2, \dots, M$  do
      if  $\epsilon_m \leq \theta_l$  then
         $\mathcal{H}_l^{weak} \leftarrow h_m$ 
      end if
    end for
  end for
end for

```

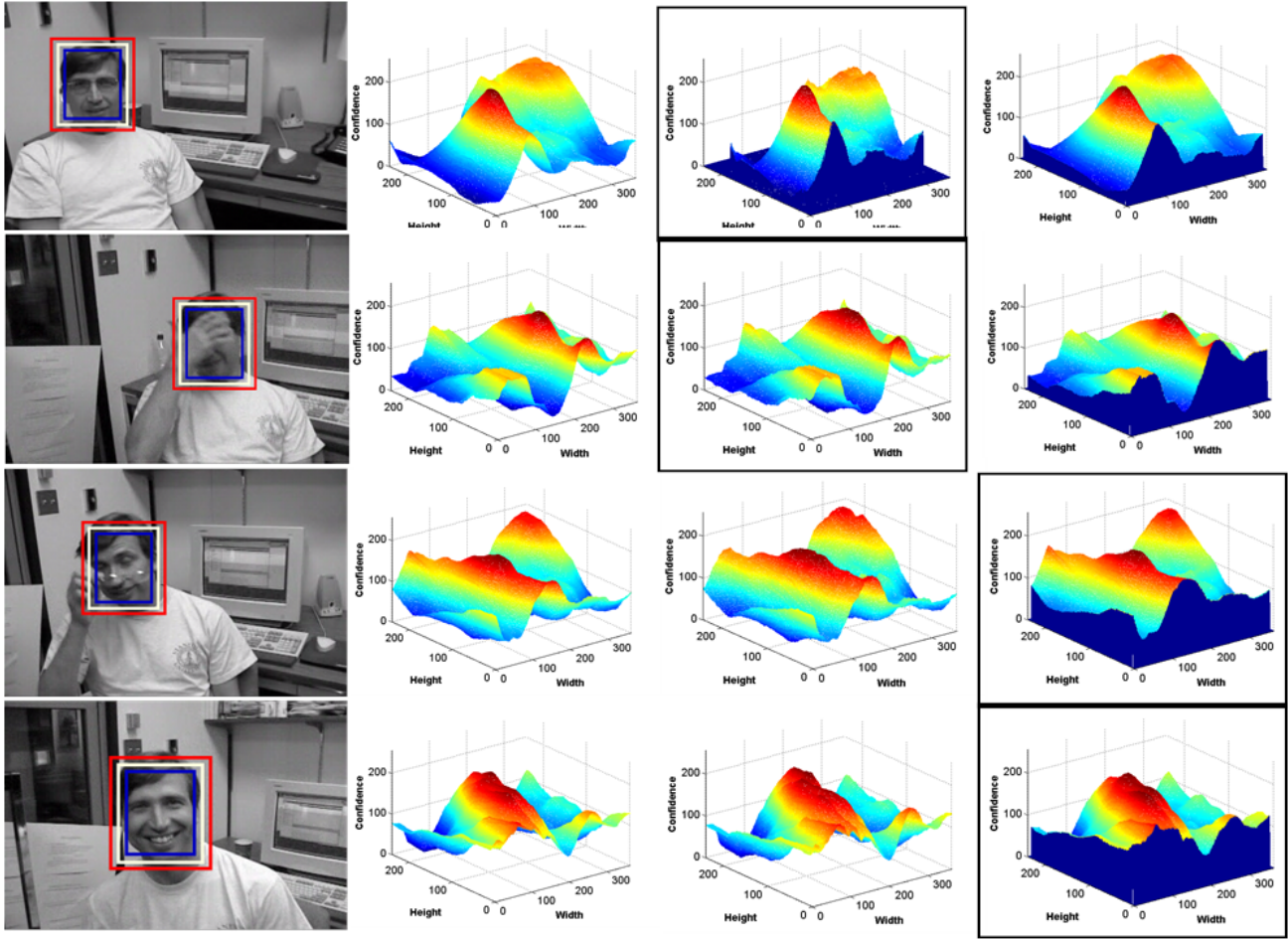


Figure 1. Experimental results on a public sequence [10] to test scale changes. First column: original frame. Second, third and fourth columns: confidence maps for scale variations (-10%, 0, +10% respectively). The highest confidence map is framed with a black box.

where $Map^{sc}(p) = H_t^{sc}(p) = conf^{sc}(p)$ If the maximum is found at a scale sc of the search region different from the current one, then the size of the search region is updated accordingly.

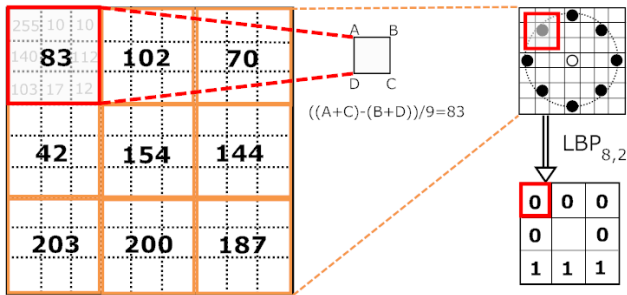


Figure 2. LBP application scheme.

4 Experimental Results

In this section several experiments are presented, in order to demonstrate the effectiveness of the proposed approaches for what concerns the adaptivity given by the dynamic replacement of the features combined with the flexibility of the scale changing.

4.1 Features

As weak learners, we employed bayesian classifiers based on Haar features (described in [11]) and on Local Binary Patterns [12], computed as shown in Fig. 2. While Haar features encode the generic shape of the object, LBP features capture local and small texture details, thus having more discriminative capability.

For each hypothesis h_k two normal distributions

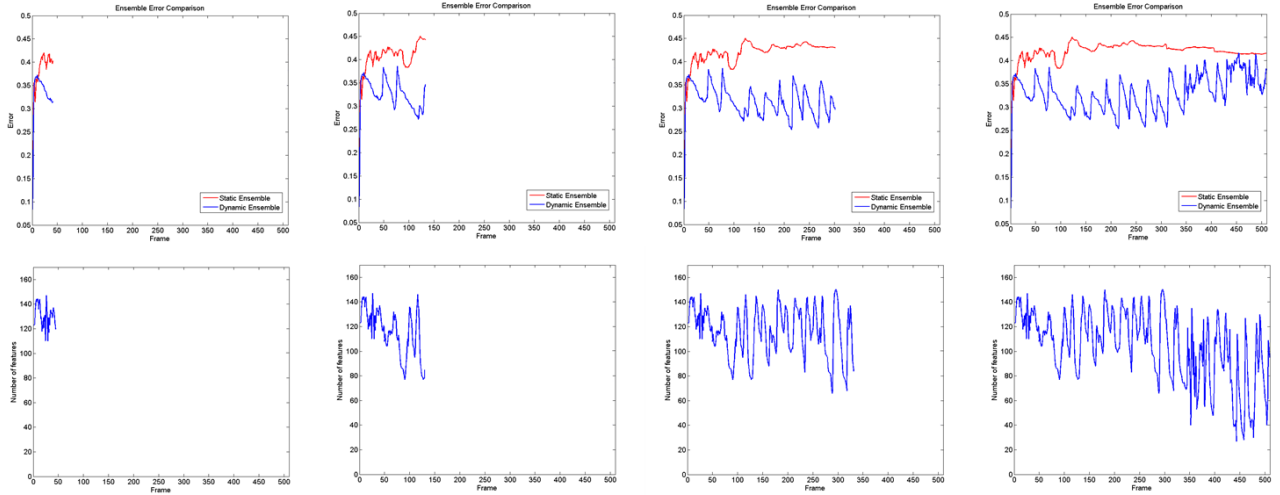


Figure 4. First row: dynamic (blue) and static (red) ensemble error comparison on the sequence of Figure 3. Second row: number of weak learners in the dynamic process.



Figure 3. Experimental results on a PTZ camera. Top to bottom: dynamic ensemble (cascade type 2) and fixed size classifier results.

$\mathcal{N}(\mu_k^+, \sigma_k^{2+})$ and $\mathcal{N}(\mu_k^-, \sigma_k^{2-})$ are maintained over observed positive and negative samples respectively [7]. This simple but effective classifier gives an interpretation to the non-metric features. When a scale change is applied, the Haar features dimensions are resized coherently with the new scale and the shape of the feature. Regarding the LBPs, the dimensions of the nine areas that compose the feature are increased or reduced equally.

4.2 Scale changes

A public video sequence [10] has been employed for this experiment. Our aim is to show how the dynamic ensemble classifier automatically selects the best scale and location for the object that is tracking. The ensemble is the cascade of type 2 based on Haar features,

that has been proposed and exhaustively tested in [16].

In Figure 1, we can see the original frame placed on the left, and the confidence maps for minus 10%, normal, and plus 10% scale respectively reported in the second, third, and fourth column respectively. The scale with the maximum confidence is framed with a black box. The first image from the top is the starting frame; the classifier has been initialized with a patch of the face, and the different search scales are marked in white, blue and red. In the second frame the subject is occluded more than 50%. As it can be seen, the classifier is robust to occlusions. In the third frame a scale change has been detected; the +10% search area has in fact a higher confidence peak. The same applies for the last frame.

4.3 Dynamic ensemble

In this section, some experiments performed on real-world video sequences are presented to validate the dynamic approach.

At every frame, we employ the cascade framework to speed up the application of the ensemble. The thresholds are fixed at $\theta_0 = 0.1$, $\theta_1 = 0.25$ and $\theta_2 = 0.4$ respectively, while the ensemble error threshold is $\Theta_{high} = 0.4$.

In Figure 3 a comparison between a classifier with a fixed number of features and a dynamic ensemble is presented. The video is the PTZ camera sequence cited before. On a AMD Athlon64™ 3500+ with 1GB of RAM the system processed the output at 20fps. The



Figure 5. CAVIAR sequence with two pedestrians.

first row shows the dynamic classifier results, while the second row represents the target tracked by the fixed size classifier, built using 150 features. In Figure 4, the comparison of the two ensemble errors (red for the fixed size and blue for the dynamic classifier) is plotted, while in the last row is drawn the number of features that represent the whole cascade in the dynamic set. As we can see, while the error increases, the number of classifiers employed grows, but also the performances are improved. The first row shows that the dynamic tracker is more accurate. In this sequence the object starts with a lateral perspective position, and moves toward the entrance of the building. The point of view is, then, from above, and the shape changes slowly. The error of the cascaded ensemble decreases as epochs proceed, but it increases in the last part of the sequence, reflecting the shape variation. As we can see from the second row of Figure 4, the rising difficulty to detect the target corresponds to a progressive augmentation of the amount of features in the ensemble.

In the second experiment, a public sequence belonging to the CAVIAR¹ dataset has been employed. In this case, a simple background subtraction technique could not be applied during all the frames of the sequence because of the sudden shadows on the floor and because of illumination changes. On the contrary, the detection via classification is more robust and can be used at each frame. In Figure 5 the results of the type 2 cascaded ensemble built with a maximum of 100 features are shown. The two pedestrians are tracked by two different ensemble instances. As we can see from Figure 6, the error of the first ensemble that tracks the first pedestrian is generally higher than the second one, due probably to the similarity between the upper part of the man and the floor, and it raises when the man disappears exiting the scene. The second ensemble error is high at the beginning, when a partial occlusion of the subject occurs, and it decreases when the second object walks unoccluded in the corridor. On the

¹<http://homepages.inf.ed.ac.uk/rbf/CAVIAR/>

contrary, when the subject crosses an area where shadows are marked and its shape is not well delimited, the error grows and more features are added in the ensemble, as we can see from the first figure. In both cases, when the error increases, the amount of features in the ensemble increases. The static ensemble error is lower when the subjects enter the scene, but does not decrease rapidly due to the slow features interchange of the ensemble.

5 Conclusions

In this work, we have proposed two main novelties over existing implementations of on-line boosting for feature selection in tracking applications. The first one consists in a dynamic ensemble of classifiers that automatically adapts its complexity to the difficulty of the classification problem. The second improvement consists in a scale change detection mechanism that exploits the fast cascaded ensemble to detect dimensions variations of tracked targets and thus automatically update the pool of possible features. Experimental results conducted on standard and our own video sequences have shown the potential of the proposed approach.

References

- [1] Y. Amit and D. Geman. A computational model for visual selection. *Neural Computation*, 11(7):1691–1715, 1999.
- [2] S. Avidan. Support vector tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(8):1064–1072, August 2004.
- [3] S. Avidan. Ensemble tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(2):261–271, 2007.
- [4] L. Bourdev and J. Brandt. Robust object detection via soft cascade. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR’05)*, volume 2, pages 236–243, Washington, DC, USA, 2005. IEEE Computer Society.
- [5] R. T. Collins, Y. Liu, and M. Leordeanu. Online selection of discriminative tracking features. *Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1631–1643, October 2005.
- [6] Y. Freund and R. E. Schapire. Experiments with a new boosting algorithm. In *Thirteen International Conference on Machine Learning*, pages 148–156, 1996.
- [7] H. Grabner and H. Bischof. On-line boosting and vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 260–267, Los Alamitos, CA, USA, 2006.
- [8] H. Grabner, M. Grabner, and H. Bischof. Real-time tracking via on-line boosting. In *Proceedings of*

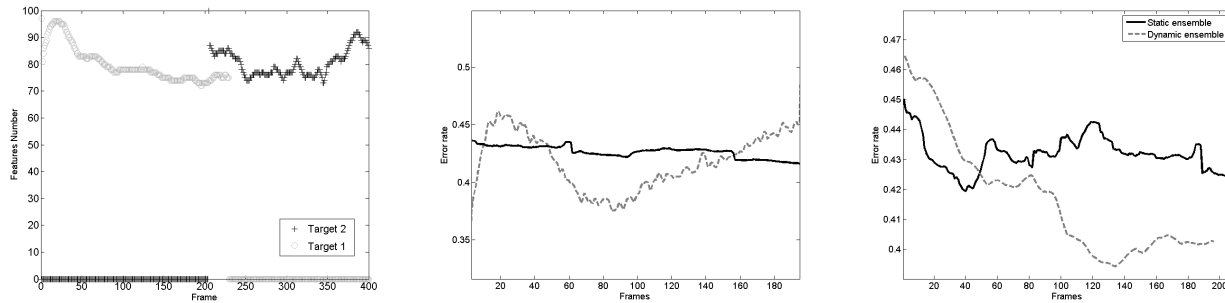


Figure 6. Left: number of features included in the cascaded ensembles for first (dotted gray line) and second (black line) object. Center: error of the first object's ensemble with respect to the static ensemble error. Right: error of the second ensemble.

- the *British Machine Vision Conference (BMVC)*, volume 1, page 4756, September 2006.
- [9] A. Hampapur, L. Brown, J. Connell, A. Ekin, N. Haas, M. Lu, H. Merkl, and S. Pankanti. Smart video surveillance: exploring the concept of multiscale spatiotemporal tracking. *Signal Processing Magazine, IEEE*, 22(2):38–51, March 2005.
- [10] A. Jepson, D. Fleet, and T. El-Maraghi. Robust online appearance models for vision tracking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 415–422, Kauai, 2001.
- [11] R. Lienhart and J. Maydt. An extended set of haar-like features for rapid object detection. In *Proceedings of the IEEE International Conference on Image Processing*, volume 1, pages 900–903, September 2002.
- [12] T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(7):971–987, 2002.
- [13] N. C. Oza. Online bagging and boosting. In *2005 IEEE International Conference on Systems, Man and Cybernetics*, volume 3, pages 2340–2345, New Jersey, October 2005.
- [14] T. Parag, F. Porikli, and A. Elgammal. Boosting adaptive linear weak classifiers for online learning and tracking. In *International Conference on Computer Vision and Pattern Recognition*, 2008.
- [15] M.-T. Pham and T.-J. Cham. Online learning asymmetric boosted classifiers for object detection. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.
- [16] L. Snidaro and I. Visentini. Fusion of heterogeneous features via cascaded on-line boosting. In *Proceedings of the Eleventh International Conference on Information Fusion*, pages 1340–1345, Cologne, Germany, June 30th–July 3rd 2008.
- [17] K. Tieu and P. Viola. Boosting image retrieval. *International Journal on Computer Vision*, 56(1-2):17–36, 2004.
- [18] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 511–518, Kauai, Hawaii, December 2001.
- [19] P. Viola and M. Jones. Robust real-time face detection. *International Journal of Computer Vision (IJCV)*, 57(2):137–154, 2004.
- [20] B. Wu and R. Nevatia. Improving part based object detection by unsupervised, online boosting. In *International Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.
- [21] R. Xiao, H. Zhu, H. Sun, and X. Tang. Dynamic cascades for face detection. In *International Conference on Computer Vision*, pages 1–8, 2007.
- [22] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Computing Surveys*, 38(4):1–45, 2006.
- [23] W. Zhang, B. Yu, G. J. Zelinsky, and D. Samaras. Object class recognition using multiple layer boosting with heterogeneous features. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 323–330, Washington, DC, USA, 2005. IEEE Computer Society.