Gradient Feature Selection for Online Boosting

Xiaoming Liu Ting Yu
Visualization and Computer Vision Lab
General Electric Global Research, Niskayuna, NY, 12309, USA
{liux,yut} AT research.ge.com

Abstract

Boosting has been widely applied in computer vision, especially after Viola and Jones's seminal work [23]. The marriage of rectangular features and integral-image-enabled fast computation makes boosting attractive for many vision applications. However, this popular way of applying boosting normally employs an exhaustive feature selection scheme from a very large hypothesis pool, which results in a less-efficient learning process. Furthermore, this poses additional constraint on applying boosting in an online fashion, where feature re-selection is often necessary because of varying data characteristic, but yet impractical due to the huge hypothesis pool. This paper proposes a gradient-based feature selection approach. Assuming a generally trained feature set and labeled samples are given, our approach iteratively updates each feature using the gradient descent, by minimizing the weighted least square error between the estimated feature response and the true label. In addition, we integrate the gradient-based feature selection with an online boosting framework. This new online boosting algorithm not only provides an efficient way of updating the discriminative feature set, but also presents a unified objective for both feature selection and weak classifier updating. Experiments on the person detection and tracking applications demonstrate the effectiveness of our proposal.

1. Introduction

Boosting refers to a simple yet effective method of learning an accurate prediction machine by combining a set of well selected weak classifiers [6]. It has shown greater performance than many traditional machine learning paradigms, when applied to solve challenging tasks in various domains [7]. Given a set of labeled training data, the boosting-based learning algorithm proceeds with the following iterative steps: 1) weak classifier selection from the hypothesis space to strength the already found strong classifier and 2) weight updating of the training data to focus the later learning process on more challenging data samples.

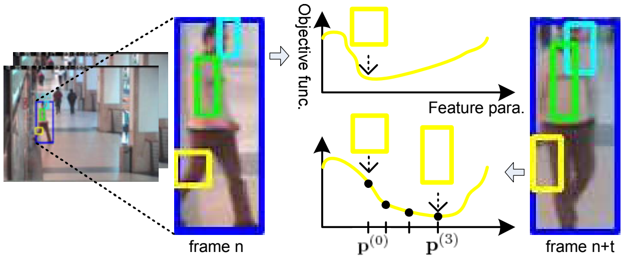

Figure 1. In boosting-based vision applications, the rectangular features/weak classifiers need to be updated due to varying appearance and shape of the object. This paper proposes a gradient-based approach to efficiently perform feature updating.

It is the seminal work of Viola and his colleagues [21,23] that bridges the gap of weak classifier design in boosting and the feature selection step, a critical component in any vision application. The introduced integral image [23] brings the fast computations of visual features, enabling the possibility of performing exhaustive search over a very large feature (hypothesis) space.

Such an exhaustive feature selection might be fine for offline learning tasks. However, it becomes a less satisfactory option for online applications, such as object tracking [2,9], where an adaptive method is needed to continuously update the existing classifiers to handle the appearance/shape variations of the object over time. With the real-time constraint, the exhaustive feature selection over a very large hypothesis space is obviously prohibited. To remedy this problem, Grabner and Bischof [8] propose a novel feature selection method, where a batch of selectors are constructed to take a random guess selection of the features. It might be effective to eventually pick up the discriminative features, this random selection, however, is still far from efficiency. It does not fully take advantage of the nature that for online applications, such as object tracking, the sequential data consumed by the online algorithm usually shows very high correlation over time, while the random selection in [8] does not really capture this time dependency.

To better exploit the correlation of sequential data, as shown in the notional example of Figure 1, this paper pro-

978-1-4244-1631-8/07/$25.00 ©2007 IEEE

poses a gradient-based feature selection approach for online boosting. Assuming a feature set generally trained from offline data is available, and during online learning newly labeled samples are sequentially given, our approach iteratively updates each feature using the gradient descent method. Formulated into the GentleBoost framework [7], the gradient-based feature selection proceeds by continuously minimizing the weighted least square error between the estimated feature response and its true label. When integrated into the online boosting, the proposed approach presents a unified objective for both feature selection and weak classifier updating. As shown in Figure 1, the three colorized rectangular features get updated by minimizing the objective in a gradient descent sense.

Our scientific findings and contributions can be summarized as follows:

◇ We propose a novel non-exhaustive feature selection approach based on the gradient descent method. This is a much more efficient scheme of learning discriminative features compared to the common way of searching the feature hypothesis space exhaustively.

◇ We present an online boosting algorithm by integrating this gradient-based feature selection with the GentleBoost framework. It leads to a unified objective for both feature selection and weak classifier updating.

◇ We pick two popular applications to demonstrate the effectiveness and efficiency of the proposed approach, *i.e.*, online learning of person detector and discriminative tracking of walking person. The quantitative analysis and comparison highlight the advantage of our proposal compared to the offline boosting and previous online boosting methods.

2. Related Work

As a general machine learning algorithm, boosting is well known for several nice properties, such as simple implementation, good generalization to unseen data, and robustness to outliers [16]. Considerable progress has also been made on applying boosting to various computer vision problems, such as image retrieval [21], face detection [24], person detection [11,25], object tracking [2,9], image alignment [15], etc. During boosting learning, how to efficiently select features is an important issue, especially critical to vision applications. To avoid the costly exhaustive feature selection, Treptow and Zell [22] propose a random feature search method via evolutionary algorithm. Dollar *et al.* [5] recently present a notion of feature selection using gradient descent. In above applications, boosting mainly serves as an offline learning machine, *i.e.*, once learned from training data, the boosted classifier is fixed during testing.

Recently, research interests are also raised to apply learning methods to online vision applications [2,3,9,10,14,17, 18]. For example, in object detection, Nair *et al.* [17] and Javed *et al.* [10] utilize a co-training approach to classify each incoming data and use it in incrementally online updating of the object classifier. In object tracking, Collins *et al.* [3] and Lim *et al.* [14] continuously update the appearance trackers through online discriminant learning.

Boosting algorithms have also been applied to online vision. Avidan [2] proposes to combine an ensemble of weak classifiers into a strong classifier using AdaBoost to handle object appearance variations, though their classifier updating is still treated in an offline learning mode. Online applications deal with sequential data, thus require the capability of weak classifiers updated in an online fashion. Oza and Russell [19] make the primary efforts on studying the sequential learning of boosted classifiers, called online boosting. They prove that with the same training set, online boosting converges statistically to offline one as the number of iterations goes to infinity. Based on [19], Grabner and Bischof [8,9] propose a novel online boosting framework, which we refer by "OB", and apply it to vision applications. Our online boosting differs from OB in that a gradient-based approach is proposed to perform feature selection and weak classifier updating given the sequential data. Detailed comparisons between these two methods are presented in later sections. Our work also advances the gradient-based feature selection notion of Dollar *et al.* [5] by providing the explicit formula of gradient-based updating.

3. Offline Boosting

We start with the introduction of the conventional boosting in the offline training mode, which we refer by "offline boosting" (OFB). Boosting-based learning iteratively selects weak classifiers to form a strong classifier using summation: $F(x) = \sum_{m=1}^{M} f_m(x)$, where $F(x)$ is the strong classifier and $f_m(x)$ is the weak classifier. There are different variants of boosting proposed in the literature [16]. We use the GentleBoost algorithm [7] based on two considerations. First, unlike the commonly used AdaBoost algorithm [6], the weak classifier in the GentleBoost algorithm is a soft-decision classifier with continuous output. This enables the strong classifier's score to be smoother and favorable for computing derivatives. In contrast, the hard weak classifiers in the AdaBoost algorithm lead to a piecewise constant strong classifier that is difficult to optimize. Second, as shown in [13], for object detection tasks, the GentleBoost algorithm outperforms other boosting methods in that it is more robust to noisy data and more resistant to outliers.

Algorithm 1 summarizes the GentleBoost algorithm. To apply it to vision applications, we firstly define the weak classifier. Given the recent success of Histogram of Oriented Gradient (HOG) feature in object detection [4, 12], we adopt it in our weak classifier design and evaluate HOG by integral histogram computation [20]. As shown in Figure 2(a), the HOG can be parameterized by (x_0, y_0, x_1, y_1), where (x_0, y_0) and (x_1, y_1) are the two corners of a cell. To

```
Input: Training data $\{\mathbf{x}_i; i \in [1, K]\}$ and their
       corresponding class labels $\{y_i; i \in [1, K]\}$.
Output: A strong classifier $F(\mathbf{x})$.
1. Initialize weights $w_i = 1/K$, and $F(\mathbf{x}) = 0$.
2. for $m = 1, 2, ..., M$ do
     (a) Fit the regression function $f_m(\mathbf{x})$ by weighted least
     square of $y_i$ to $\mathbf{x}_i$ with weights $w_i$
     $f_m(\mathbf{x}) = \text{argmin}_{f \in \mathcal{F}} \epsilon(f) = \sum_{i=1}^{K} w_i (f(\mathbf{x}_i) - y_i)^2$.
     (b) Update $F(\mathbf{x}) = F(\mathbf{x}) + f_m(\mathbf{x})$.
     (c) Update the weights by $w_i = w_i e^{-y_i f_m(\mathbf{x}_i)}$ and
     normalize the weights such that $\sum_{i=1}^{K} w_i = 1$.
end
3. Output the classifier $sign[F(\mathbf{x})] = sign[\sum_{m=1}^{M} f_m(\mathbf{x})]$.
```

Algorithm 1: The GentleBoost algorithm.

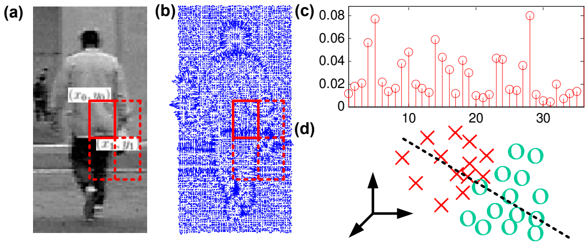

Figure 2. (a) The parametrization of a cell; (b) The gradient map; (c) The HOG of a block; (d) The HOG features of positive and negative samples.

incorporate spatial information into HOG, a 2 × 2 cell array is used to form a block. For each cell, the b-bin histogram of the gradient magnitude at each orientation is computed. The concatenation of the HOG for 4 cells within one block forms a $4b$-dimensional vector, as shown in Figure 2(c). The large hypothesis space $\mathcal{F}$, where (x_0, y_0, x_1, y_1) resides, is obtained via an exhaustive construction within the template coordinate system. For example, there are more than $300,000$ such block features for a template with the size of 30×30. Hence, the exhaustive feature selection process, Step 2(a) in Algorithm 1, is the most computationally intensive step in the GentleBoost algorithm.

When a multi-dimensional feature vector, such as the histogram, is used in the weak classifier, the conventional method of computing the threshold in the decision stump classifier, which often works comfortably with a 1-D feature, can not be directly applied. In this paper, we employ the idea of boosted histogram proposed by Laptev [11]. As shown in Figure 2(d), a weighted Linear Discriminative Analysis (LDA) is applied to the histogram features of positive and negative samples, and results in the optimal projection direction β. Thus, all histograms can be converted to 1-D features by computing the inner product with β.

In summary, we use the following weak classifier:

$$f(\mathbf{x}; \mathbf{p}) = \frac{2}{\pi} atan(\beta^{\mathbf{T}} \mathbf{h}(x_0, y_0, x_1, y_1) - t), \qquad (1)$$

where β is the LDA projection direction, t is the threshold and $\mathbf{p}$ are the parameters of the weak classifier $\mathbf{p} = [x_0, y_0, x_1, y_1, \beta, t]^{\mathbf{T}}$. Given a cell location (x_0, y_0, x_1, y_1), the histogram features $\mathbf{h}$ are computed from all training data via the integral histogram. Weighted LDA is applied to compute β, and finally t is obtained through binary search along the span of LDA projections of all training data, such that the weighted least square error (WLSE) is minimal. Similar to [15], we use the $atan()$ function in the weak classifier, instead of the commonly used decision stump, because of its derivability with respect to the parameters $\mathbf{p}$.

4. Gradient Feature Selection

4.1. Problem definition

We define the feature selection as a process of updating the parameters of each weak classifier $\mathbf{p}_m$. As shown in Step 2(a) of Algorithm 1, the WLSE is used in selecting the weak classifier from the hypothesis space during the boosting iteration. Hence, it is natural to use the WLSE as the objective function for the feature selection (updating). This leads to the following problem we are trying to solve

$$\min_{\mathbf{p}} \epsilon(f(\mathbf{x}; \mathbf{p})) = \min_{\mathbf{p}} \sum_{i=1}^{K} w_i (f(\mathbf{x}_i; \mathbf{p}) - y_i)^2. \qquad (2)$$

In the context of feature selection, solving this problem means that given the initial parameters $\mathbf{p}^{(0)}$, we look for the new parameters of the weak classifier that can lead to smaller WLSE on the dataset $\{\mathbf{x}_i\}$ with K samples in total. We choose to use the gradient descent method to solve this problem iteratively.

4.2. Algorithm derivation

Plugging Eq. 1 into Eq. 2, the function to be minimized is

$$\epsilon = \sum_{i=1}^{K} w_i (\frac{2}{\pi} atan(\beta^{\mathbf{T}} \mathbf{h}_i(x_0, y_0, x_1, y_1) - t) - y_i)^2. \qquad (3)$$

Taking the derivative with respect to $\mathbf{p}$ gives

$$\frac{d\epsilon}{d\mathbf{p}} = \sum_{i=1}^{K} 2 w_i (f(\mathbf{x}_i) - y_i) \frac{df_i}{d\mathbf{p}}, \qquad (4)$$

where $\frac{df_i}{d\mathbf{p}} = [\frac{\partial f_i}{\partial x_0} \frac{\partial f_i}{\partial y_0} \frac{\partial f_i}{\partial x_1} \frac{\partial f_i}{\partial y_1} \frac{\partial f_i}{\partial \beta} \frac{\partial f_i}{\partial t}]^{\mathbf{T}}$. Based on Eq. 1, we have

$$\begin{aligned} \frac{\partial f_i}{\partial z} &= \frac{2}{\pi} \frac{\beta^{\mathbf{T}} \frac{\partial \mathbf{h}_i}{\partial z}}{1 + (\beta^{\mathbf{T}} \mathbf{h}_i - t)^2}, \quad z = x_0, y_0, x_1, y_1, \\ \frac{\partial f_i}{\partial \beta} &= \frac{2}{\pi} \frac{\mathbf{h}_i}{1 + (\beta^{\mathbf{T}} \mathbf{h}_i - t)^2}, \\ \frac{\partial f_i}{\partial t} &= \frac{2}{\pi} \frac{-1}{1 + (\beta^{\mathbf{T}} \mathbf{h}_i - t)^2}. \end{aligned} \qquad (5)$$

As an example, we show how to compute the derivative of the histogram feature with respect to one of the cell location parameters x_0, *i.e.*, $\frac{\partial \mathbf{h}_i}{\partial x_0}$. The remaining partial derivatives, $\frac{\partial \mathbf{h}_i}{\partial x_1}$,$\frac{\partial \mathbf{h}_i}{\partial y_0}$ and $\frac{\partial \mathbf{h}_i}{\partial y_1}$, can be computed similarly. As mentioned in Section 3, $\mathbf{h}_i$ is a $4b$-dimensional vector $\mathbf{h}_i(x_0, y_0, x_1, y_1) = [h_{i,1}, h_{i,2}, \cdots, h_{i,4b}]^{\mathbf{T}}$ computed from the 4 cells of a block. Given a labeled image $\mathbf{x}_i$, the gradient map is computed and a set of integral images of the magnitude of the gradient at each orientation $\{\bar{\mathbf{x}}_{i,1}, \bar{\mathbf{x}}_{i,2}, \cdots, \bar{\mathbf{x}}_{i,b}\}$ is obtained. Since all 9 corners of the 2×2 cells can be fully described by the cell location (x_0, y_0, x_1, y_1), each bin of the histogram can be computed via accessing the integral image $\bar{\mathbf{x}}_{i,j}$ at the corresponding pixel location defined as follows

$$\begin{aligned}
h_{i,j} &= \bar{\mathbf{x}}_{i,j}(x_0, y_0) + \bar{\mathbf{x}}_{i,j}(x_1, y_1) - \bar{\mathbf{x}}_{i,j}(x_0, y_1) \\
&\quad - \bar{\mathbf{x}}_{i,j}(x_1, y_0), \\
h_{i,j+b} &= \bar{\mathbf{x}}_{i,j}(x_1, y_0) + \bar{\mathbf{x}}_{i,j}(2x_1 - x_0, y_1) - \bar{\mathbf{x}}_{i,j}(x_1, y_1) \\
&\quad - \bar{\mathbf{x}}_{i,j}(2x_1 - x_0, y_0), \\
h_{i,j+2b} &= \bar{\mathbf{x}}_{i,j}(x_0, y_1) + \bar{\mathbf{x}}_{i,j}(x_1, 2y_1 - y_0) - \bar{\mathbf{x}}_{i,j}(x_1, y_1) \\
&\quad - \bar{\mathbf{x}}_{i,j}(x_0, 2y_1 - y_0), \\
h_{i,j+3b} &= -\bar{\mathbf{x}}_{i,j}(2x_1 - x_0, y_1) - \bar{\mathbf{x}}_{i,j}(x_1, 2y_1 - y_0) \\
&\quad + \bar{\mathbf{x}}_{i,j}(x_1, y_1) + \bar{\mathbf{x}}_{i,j}(2x_1 - x_0, 2y_1 - y_0).
\end{aligned} \tag{6}$$

where $i \in [1, K]$ and $j \in [1, b]$. The Eq. 6 defines total $4b$ equations to compute the $4b$-dimensional orientation histogram $\mathbf{h}_i$. Given above $\mathbf{h}_i$ definition, the derivative of $\mathbf{h}_i$ with respect to x_0 can be computed by

$$\begin{aligned}
\frac{\partial h_{i,j}}{\partial x_0} &= \frac{\partial \bar{\mathbf{x}}_{i,j}}{\partial x}|_{(x_0,y_0)} - \frac{\partial \bar{\mathbf{x}}_{i,j}}{\partial x}|_{(x_0,y_1)}, \\
\frac{\partial h_{i,j+b}}{\partial x_0} &= -\frac{\partial \bar{\mathbf{x}}_{i,j}}{\partial x}|_{(2x_1-x_0,y_1)} + \frac{\partial \bar{\mathbf{x}}_{i,j}}{\partial x}|_{(2x_1-x_0,y_0)}, \\
\frac{\partial h_{i,j+2b}}{\partial x_0} &= \frac{\partial \bar{\mathbf{x}}_{i,j}}{\partial x}|_{(x_0,y_1)} - \frac{\partial \bar{\mathbf{x}}_{i,j}}{\partial x}|_{(x_0,2y_1-y_0)}, \\
\frac{\partial h_{i,j+3b}}{\partial x_0} &= \frac{\partial \bar{\mathbf{x}}_{i,j}}{\partial x}|_{(2x_1-x_0,y_1)} - \frac{\partial \bar{\mathbf{x}}_{i,j}}{\partial x}|_{(2x_1-x_0,2y_1-y_0)},
\end{aligned} \tag{7}$$

where $\frac{\partial \bar{\mathbf{x}}_{i,j}}{\partial x}|_{(x_0,y_0)}$ is the partial derivative of $\bar{\mathbf{x}}_{i,j}$ with respect to the horizontal axe x and evaluated at (x_0, y_0), and can be easily computed via discrete differentiation such as $\frac{\partial \bar{\mathbf{x}}_{i,j}}{\partial x}|_{(x_0,y_0)} = \frac{1}{2}[\bar{\mathbf{x}}_{i,j}(x_0 + 1, y_0) - \bar{\mathbf{x}}_{i,j}(x_0 - 1, y_0)]$.

Please note that the above feature gradient derivations are not necessarily only suitable to the HOG feature. In fact, all integral-image-enabled features can have similar formulas to compute the gradients, thus can all be used in the proposed framework.

5. Online Boosting

5.1. Online boosting with gradient feature selection

Understanding that feature gradients, such as $[\frac{\partial \mathbf{h}_i}{\partial x_0}, \frac{\partial \mathbf{h}_i}{\partial x_1}, \frac{\partial \mathbf{h}_i}{\partial y_0}, \frac{\partial \mathbf{h}_i}{\partial y_1}]$, bring the connection between minimizing the WLSE of the weak classifier and feature updating, now we introduce our online boosting algorithm. Our online boosting follows the basic scheme of the conventional offline boosting. That is, weak classifier selection and weights updates are performed in each iteration of the boosting. However, these two approaches differ in the weak classifier selection step. In the offline boosting, this step is achieved by exhaustively evaluating all classifier candidates in a hypothesis space, which is often huge due to the over-complete feature set. Hence, this is a very computationally demanding operation. In contrast, our online boosting treats the weak classifier selection as a gradient descent optimization problem. Our approach takes an initial set of weak classifiers and a number of labeled training samples as inputs. For each weak classifier $f_m(\mathbf{p}_m)$, the online boosting iteratively updates the feature parameters $\mathbf{p}_m$ according to the above computed gradient $\frac{d\epsilon}{d\mathbf{p}}$. The objective function $\epsilon(\mathbf{p}_m)$ is computed at each iteration and expected to keep decreasing. The iteration will cease if the objective arrives at a minimum, or the magnitude of the gradient $|\frac{d\epsilon}{d\mathbf{p}}|$ is smaller than a threshold. Algorithm 2 summarizes our algorithm, which we refer as "Gradient-based Online Boosting" (GOB).

```
Input: Training data $\{\mathbf{x}_i; i \in [1, K]\}$, their corresponding
       class labels $\{y_i; i \in [1, K]\}$, and an initial set of
       weak classifiers $\{f_m(\mathbf{p}_m); m \in [1, M]\}$.
Output: An updated strong classifier $F(\mathbf{x})$.
1. Initialize weights $w_i = 1/K$, and $F(\mathbf{x}) = 0$.
2. for $m = 1, 2, ..., M$ do
    (a) Compute $\epsilon(\mathbf{p}_m)$ and $\frac{d\epsilon}{d\mathbf{p}}|_{\mathbf{p}_m}$ using Eq. 3 and Eq. 4.
    (b) if $\epsilon(\mathbf{p}_m)$ is decreasing then
        Update $\mathbf{p}_m = \mathbf{p}_m - \frac{d\epsilon}{d\mathbf{p}}|_{\mathbf{p}_m}$.
        Jump to (a).
    end
    (c) Update $F(\mathbf{x}) = F(\mathbf{x}) + f_m(\mathbf{x}; \mathbf{p}_m)$.
    (d) Update the weights by $w_i = w_i e^{-y_i f_m(\mathbf{x}_i)}$ and
    normalize the weights such that $\sum_{i=1}^{K} w_i = 1$.
end
3. Output the classifier
$sign[F(\mathbf{x})] = sign[\sum_{m=1}^{M} f_m(\mathbf{x}; \mathbf{p}_m)]$.
```

Algorithm 2: The proposed gradient-based online boost algorithm (GOB).

5.2. Comparison between OB and GOB

We briefly describe the conventional OB [8] in Algorithm 3, in order to compare it with the proposed GOB. In OB, the strong boosted classifier is composed of a batch of selectors, and each selector contains a random subset of weak classifiers. During the updating, given a new training sample, each selector is responsible for selecting the best weak classifier from its subset, by updating all classifiers in its subset simultaneously and choosing the one with the

```
for each selector do
    (a) for each weak classifier in the selector do
        Update the weak classifier and compute error.
    end
    (b) Choose the weak classifier with lowest error.
    (c) Update sample importance weights.
    (c) Replace the worst weak classifier with a randomly
    selected feature.
end
```

Algorithm 3: The simplified description of the conventional online boost algorithm (OB).

lowest error. All selectors will conduct the same updating sequentially. Note that, unlike GOB that both the feature (block location) and the weak classifier (LDA direction and threshold) get updated, OB only updates the weak classifier (threshold) [8]. The selector will also update its feature set by replacing the worst weak classifier in its subset with a randomly selected feature.

OB depends on random selections of the discriminative features, which may show good performance when the object being tracked is rigid, and camouflage features are around. However, this method might suffer from the large object deformation. Once object deformation appears, most of the local rectangular/block features may become invalid due to the local mis-alignment. This consequently will result in OB continuously looking for new features to add into the selector. However, such a process is blindly random without considering the nature that a local deformation strongly implies that a local shifting of the originally found rectangular feature is still very likely to be a good discriminative feature. In essence, the proposed gradient-based feature selection method is to train each rectangular feature doing exactly this local object deformation tracking in a discriminative sense. Therefore, the overall capacity of our algorithm to handle the object deformation is much better than OB, which we will demonstrate in Section 6.2.

We can also compare the computational costs of OB and GOB to see the efficiency of our method. Assuming M weak classifiers (GOB) or selectors (OB) are trained in total, each one of the M selector has N weak classifiers and the cost of updating each weak classifier is C, the average cost of updating all weak classifiers is $O(NCM)$ for OB. Similarly, in our proposal, assuming the average number of iterations for updating one weak classifier is n, the cost of updating all weak classifiers is $O(nCM)$. For OB, it is reported that $N = 250$ [8]. However, we experimentally learn that $n = 2$. Thus, a huge computational gain is observed using our approach. [8] also mentions an efficient version of the algorithm by using the same N weak classifiers for all selectors. Hence, the computation cost will reduce to $O(NC + M)$ where our algorithm is still more efficient than OB in this case.

Figure 3. Positive samples of the person detection database.

6. Experiments

We demonstrate the proposed gradient-based online boosting idea with two popular applications: person detection and person tracking.

6.1. Person detection

Person detection is often achieved by learning a dedicated classification-based detector from labeled data. Given large amount of positive training samples and infinite number of negative samples in theory, the computation and storage burden can make the training impractical, especially when new samples are received sequentially and detector updating is needed. There are three ways to handle this issue via boosting. First, if the computation and storage allow, we can re-train the detector with all available data using OFB. Second, when the computation burden exceeds the limit, we can test the detector on all available data first, and only re-train on a challenging subset of the original data. We call this approach as "Offline Selective Boosting" (OFSB). Third, GOB can be used to update the detector with the newly received samples.

We use a labeled person database with 915 positive and 6361 negative samples. Some of the typical samples are shown in Figure 3. To imitate the real-world scenario of continuously receiving new data in sequential learning, we partition this database into three distinct sets: the training set s_1, the updating set s_2, and the test set s_3, where each has $\frac{2}{5}$, $\frac{2}{5}$, and $\frac{1}{5}$ positive and negative samples of the database. Furthermore, s_2 is evenly partitioned into 6 subsets, $\{s_{2,1}, \cdots, s_{2,6}\}$, which are assumed to arrive sequentially. Figure 4 illustrates our experimental design. To begin the experiments, an initial classifier is trained from s_1 using OFB. When a new data set arrives, such as $s_{2,1}$ or $s_{2,2}$, OFB is re-trained on all available data up to the current time. However, when $s_{2,3}$ arrives, it is assumed that re-training on all data starts to become impractical due to the demanding computation cost and memory requirement. Hence, OFSB is used in this case, and the present classifier is tested on the set of $\{s_1, s_{2,1}, s_{2,2}, s_{2,3}\}$. The sample, which is closer to the classification boundary (measured by the strong classifier response), is added into a subset, until the size of this subset reaches the allowed limit, *i.e.*, the size of $\{s_1, s_{2,1}, s_{2,2}\}$. Finally, offline boosting is used to re-train a classifier based on this chosen subset. In contrast, when each subset of s_2 sequentially arrives, we can also use

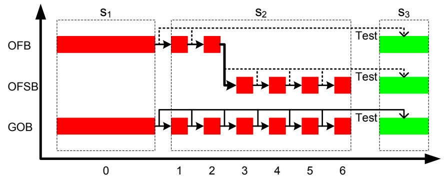

Figure 4. Experimental design. Three learning schemes, offline boosting (OFB), offline selective boosting (OFSB) and gradient-based online boosting (GOB), are used to train and update a detector from s_1 and s_2. After each update, s_3 is used to test the detection performance.

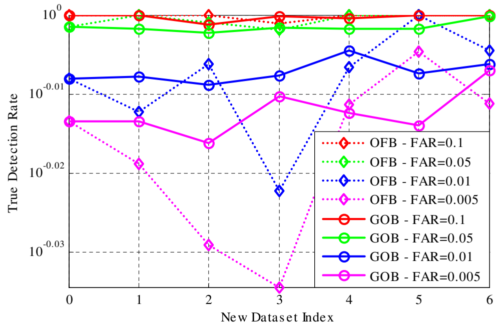

Figure 5. Detection performance of continuously learned boosted classifier from new data sets. The horizontal axis represents the index of the new data set while the vertical axis represents the detection rate of the updated classifier tested on s_3. Stable and increasing performance can be observed from our gradient-based online boosting (GOB) algorithm.

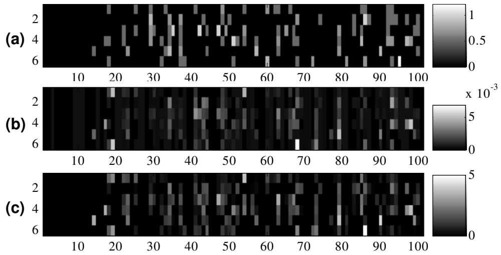

Figure 6. Updating of the weak classifiers in GOB. The vertical axes is the new data set index, from 1 to 6. The horizontal axes is the index of weak classifier. The brightness of each grid indicates the amount of changes for the parameter of each weak classifier. (a) MSE of the box location change; (b) Angle change of the LDA projection; (c) Change of the threshold.

GOB to continuously update the present classifier based on the new subset only.

After each re-training or updating, the performance of the resulting boosted classifier is tested on s_3, as shown in Figure 5. For clear illustration purpose, the detection rate-false alarm curve is not shown. Instead we sample the curve at four particular false alarm rates (FAR), namely 0.1, 0.05, 0.01, 0.005. Note that the dash line represents OFB when the new data set index is less than 3. Otherwise, it is OFSB. Figure 6 shows the amount of changes in the weak classifiers during GOB learning. It can be seen that majority of the weak classifiers actually being updated when receiving new data.

A number of observations can be made. First, as expected, overall both online and offline boosting increase their performance as taking more data for training. Second, since the re-training process of OFB/OFSB throws away previously trained classifier, it is more likely to obtain a classifier that has unstable performance. While GOB always starts with the prior knowledge, *i.e.*, the present classifier, to improve its capacity on the new data, meanwhile still maintain its discriminative power on the old data. Hence, the online boosting demonstrates a stable and increasing performance on the unseen data set s_3.

GOB also has a huge computational advantage over OFB/OFSB. For the experiments in Figure 5, GOB takes around 15 minutes to finish updating when receiving each new data set. However, OFB/OFSB take more than 7 hours to finish re-training once. Both time costs are based on a Matlab$^{\text{TM}}$ implementation and exclude the computation of feature extraction (computing integral images of oriented gradients).

6.2. Person tracking

Following the same weak classifier design as in person detection, the proposed GOB is also applied to person tracking application. The main idea of applying online boosting in this domain is to formulate the tracking problem as a discriminative training process, which continuously updates the boosted classifier to discriminate the human region from its nearby backgrounds.

We take a similar tracking-and-updating process cycle as [8]. Once an optimal human region is tracked with the previously trained boosted classifier, a set of new training data is cropped from this location (positive data) and nearby neighborhoods (negative data). The weak classifiers are updated with this new data set. Essentially, a boosted classifier trained in this way manages to learn a drifting detector of the tracker. As long as the tracker keeps following the person with accurate alignments, the drifting detector will report higher classification scores; on the contrary, if the tracker deviates from its optimal location, negative responses will be returned.

One example of applying the proposed GOB algorithm to person tracking in surveillance videos is shown in Figure 7(a), where as the person walks towards the camera, his body pose also gradually changes from a side view to the front view. GOB is capable of learning to follow this turning action, and continuously track the person. The colorized small rectangles illustrate the weak classifiers that have changed their box location parameters (x_0, y_0, x_1, y_1) by GOB at the corresponding frame. Due to the balance between the classification performance and efficiency, we select 50 weak classifiers in total for all tracking experiments.

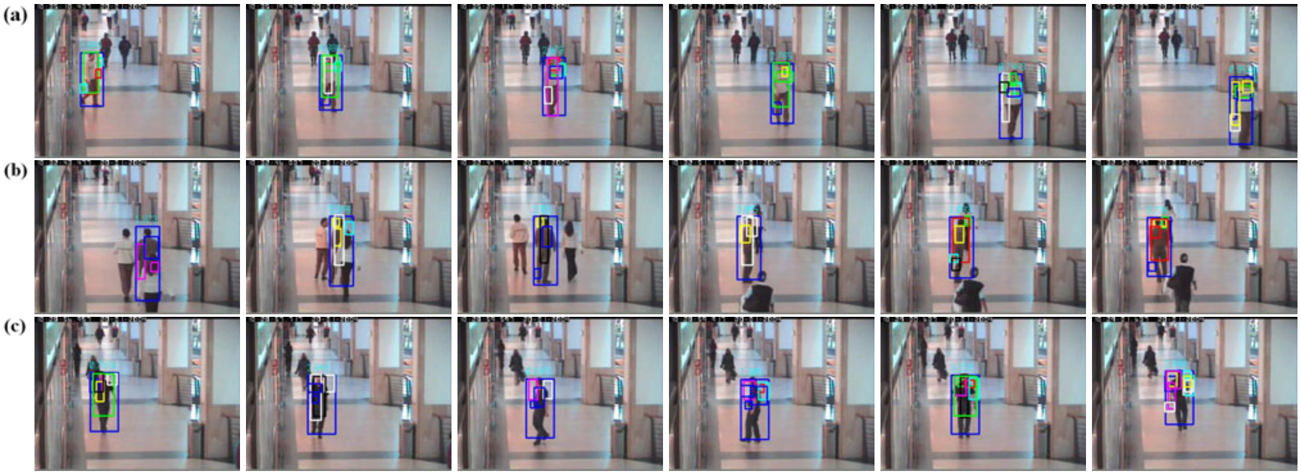

Figure 7. Tracking results by the proposed GOB algorithm. (a) a person changing pose; (b) a person under severe occlusion and continuously turning over; (c) a person making a full turn.

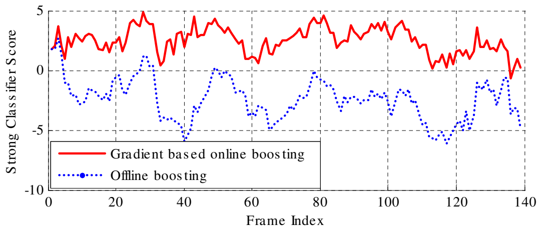

Figure 8. Strong classifier scores (GOB vs OFB) evaluated at the optimal location returned by GOB.

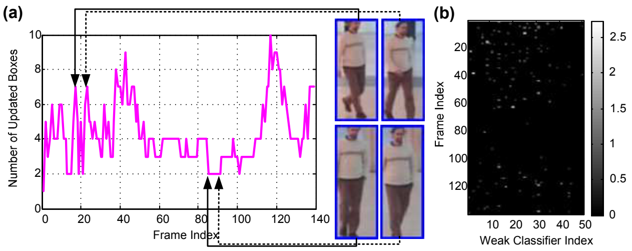

Figure 9. (a) The number of weak classifiers that change their feature box locations over time. Higher number indicates relative larger shape and appearance deformation; (b) MSE of the box location changes over time. The intensity reflects the magnitude of the feature location changes.

Without surprise, an offline trained frontal view person classifier can not reliably track the subject, and loses tracking at the very early stage of the sequence. From Figure 8, it is easy to see the benefits of online classifier updating. The strong classifier responses with the gradient updating are much higher than the one without updating. It verifies the power of the gradient-based boosting to keep learning better discriminant features (weak classifiers) to separate the person from backgrounds.

The proposed GOB framework is particularly suitable for the tracking application, since, during the successive frames, the object of interest will generally only show minor changes in terms of appearance and shape. These minor but non-ignorable changes could be best captured by the gradient learning framework. During the updating within two continuous frames, most of the weak classifiers will not change their feature locations, reflecting the fact that the object parts corresponding to these feature locations show no difference between two frames, while only a few feature locations will get updated, implying those regions are undergoing local shape deformation and/or appearance change.

This property is demonstrated in Figure 7(a), where the number of colorized rectangles (weak classifiers with changed locations) only represents a small portion of total weak classifiers (50) we applied to the sequence. Furthermore, in Figure 9(a), we show the plot of the number of weak classifiers with changed locations through updating for the sequence in Figure 7(a). It is observed that during the whole sequence at most one-fifth (10/50) of the weak classifiers update their locations between successive frames. It validates that GOB is very efficient, yet still effective to capture the minor appearance changes using discriminative training. Also, the higher number of updating indicates relative larger shape and appearance deformation. In Figure 9(b), we illustrate a similar figure as in Figure 6(a), while here the updating is for online tracking application. Again, the relative less updating reflects the fact that the unseen new data received over time is actually showing strong correlations with the previous ones. Hence, only minor updating is needed, which is particularly true for tracking application.

In Figure 7(b), we show a challenging sequence that our GOB can tackle. The object being tracked is under very severe occlusion, when a girl wearing the white clothes is approaching and passing the object of interest. GOB shows great performance on learning the relatively stable yet dis-

Sequence ID	1	2	3	4	5	6	7	8	9	10
Conventional online boosting (OB)	15.1	Lost	14.1	Lost	Lost	Lost	5.0	Lost	4.6	Lost
Gradient-based online boosting (GOB)	6.2	10.9	7.1	Lost	7.7	11.0	5.5	5.4	4.4	8.8

Table 1. Average tracking errors of two online boosting-based trackers. The average size of the tracking window is 140×46 pixels.

criminant features (the head and shoulder in this example), thus managing to follow the person even under such a challenging situation. Figure 7(c) is another example where the person makes a full turn and walks back.

Finally, we also carry out a quantitative study on tracking using two methods: the GOB tracker and the OB tracker [8,9]. We strictly follow the OB algorithm presented in [8] to repeat their implementation, where each selector has a different subset of the weak classifiers. We test on ten video sequences (4000+ frames in total) from the CAVIAR database [1]. The ground truth locations of these data are given. The average tracking errors against the ground truths are reported in Table 1. The tracking error is defined as the pixel distance between the center of the bounding box returned by the trackers and that of the ground truth. Consistent with the discussion in Section 5.2, our GOB tracker outperforms the OB tracker. The gained performance is attributed to the local gradient search capability in each weak classifier updating, so as to keep following its discriminative yet shifted/deformed features.

7. Conclusions

We have introduced a novel gradient-based feature selection approach for online boosting. Assuming a generally trained feature set and labeled samples are given, our approach iteratively updates each feature using the gradient descent method, by minimizing the weighted least square error between the estimated feature response and the true label. Furthermore, we integrate the gradient-based feature selection with an online boosting framework. The proposed online boosting is applied to the person detection and tracking applications. Extensive experiments demonstrate the effectiveness and efficiency of our proposal. Future directions of this work include extending online learning for other variants of boosting and applying online boosting into other vision applications.

Acknowledgement

This project was supported by awards #2005-IJ-CX-K060 and #2006-IJ-CX-K045 awarded by the National Institute of Justice, Office of Justice Programs, US Department of Justice. The opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the Department of Justice.

References

[1] http://homepages.inf.ed.ac.uk/rbf/CAVIAR/.

[2] S. Avidan. Ensemble tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 29(2):261–271, 2007.

[3] R. Collins, Y. Liu, and M. Leordeanu. On-line selection of discriminative tracking features. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 27(10):1631–1643, 2005.

[4] N. Dalal and W. Triggs. Histograms of oriented gradients for human detection. In *Proc. IEEE Computer Vision and Pattern Recognition, San Diego, California*, volume 1, pages 886–893, 2005.

[5] P. Dollar, Z. Tu, H. Tao, and S. Belongie. Feature mining for image classification. In *Proc. IEEE Computer Vision and Pattern Recognition, Minneapolis, Minnesota*, 2007.

[6] Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[7] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: A statistical view of boosting. *The Annals of Statistics*, 38(2):337–374, 2000.

[8] H. Grabner and H. Bischof. On-line boosting and vision. In *Proc. IEEE Computer Vision and Pattern Recognition, New York, NY*, volume 1, pages 260–267, 2006.

[9] H. Grabner, M. Grabner, and H. Bischof. Real-time tracking via on-line boosting. In *Proc. 17th British Machine Vision Conference, Edinburgh, UK*, volume 1, pages 47–56, 2006.

[10] O. Javed, S. Ali, and M. Shah. Online detection and classification of moving objects using progressively improving detectors. In *Proc. IEEE Computer Vision and Pattern Recognition, San Diego, California*, volume 1, pages 696–701, 2005.

[11] I. Laptev. Improvements of object detection using boosted histograms. In *Proc. 17th British Machine Vision Conference, Edinburgh, UK*, volume 3, pages 949–958, 2006.

[12] K. Levi and Y. Weiss. Learning object detection from a small number of examples: The importance of good features. In *Proc. IEEE Computer Vision and Pattern Recognition, Washington, DC*, volume 2, pages 53–60, 2004.

[13] R. Lienhart, A. Kuranov, and V. Pisarevsky. Empirical analysis of detection cascades of boosted classifiers for rapid object detection. In *Proc. 25th Pattern Recognition Symposium, Madgeburg, Germany*, pages 297–304, 2003.

[14] J. Lim, D. A. Ross, R.-S. Lin, and M.-H. Yang. Incremental learning for visual tracking. In *Advances in Neural Information Processing Systems 17, NIPS, Vancouver, Canada*, pages 793–800, 2004.

[15] X. Liu. Generic face alignment using boosted appearance model. In *Proc. IEEE Computer Vision and Pattern Recognition, Minneapolis, Minnesota*, 2007.

[16] R. Meir and G. Raetsch. *An introduction to boosting and leveraging*. S. Mendelson and A. Smola, Editors, Advanced Lectures on Machine Learning, LNAI 2600. Springer, 2003.

[17] V. Nair and J. Clark. An unsupervised, online learning framework for moving object detection. In *Proc. IEEE Computer Vision and Pattern Recognition, Washington, DC*, volume 2, pages 317–324, 2002.

[18] A. Opelt, M. Fussenegger, A. Pinz, and P. Auer. Weak hypotheses and boosting for generic object detection and recognition. In *Proc. 8th European Conf. Computer Vision, Prague, Czech Republic*, volume 2, pages 71–84, 2004.

[19] N. Oza and S. Russell. Experimental comparisons of online and batch versions of bagging and boosting. In *Proc. 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 359–364, 2001.

[20] F. Porikli. Integral histogram: A fast way to extract histograms in cartesian spaces. In *Proc. IEEE Computer Vision and Pattern Recognition, San Diego, California*, volume 1, pages 829–836, 2005.

[21] K. Tieu and P. Viola. Boosting image retrieval. *Int. J. Computer Vision*, 56(1-2):17–36, 2004.

[22] A. Treptow and A. Zell. Combining adaboost learning and evolutionary search to select features for real-time object detection. In *IEEE Congress on Evolutionary Computation*, volume 2, pages 2107–2113, 2004.

[23] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. IEEE Computer Vision and Pattern Recognition, Hawaii*, volume 1, pages 511–518, 2001.

[24] P. Viola and M. Jones. Robust real-time face detection. *Int. J. Computer Vision*, 57(2):137–154, 2004.

[25] P. Viola, M. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. *Int. J. Computer Vision*, 63(2):153–161, 2005.