

Object Tracking based on Online Learning and Local Features

TAKAYOSHI YAMASHITA^{†1,†2} and HIRONOBU FUJIYOSHI^{†1}

Object detection, object tracking and action recognition are the building blocks for understanding human behavior. Of the three, object tracking plays a vital role in focusing on the person thus enabling action recognition. In this paper, we propose a tracking framework called "Online Real Boosting", an improvement over the popular Online Boosting. Online Boosting is a learning algorithm that selects a discriminative classifiers based on the just arrived samples, in a tracking scenario occlusion or appearance changes results in errors that propagate resulting in drifts. The proposed method reduces drift utilizing Real Adaboost and a probability density function of the object and background. The proposed tracking algorithm was trained for tracking the human head, and the results were compared against an existing method, mean-shift and Online Boosting. As a result, proposed method achieved comparable or better performance, besides improvement on processing speed by reducing the number of weak classifiers by a half compared to Online Boosting. Of the many challenges in object tracking, appearance changes owing to the articulated nature of the object is the biggest. In Online Boosting based tracking, weak classifiers are selected from a pool of classifiers trained offline, and that are sensitive to appearance changes. In this paper, a new feature structure that is robust to appearance change, called the "Soft Decision Feature" is also introduced. The Online Real Boosting and Soft Decision Features were applied to snippets of humans with complex appearance changes. Experimental results show that the proposed combination tracked scenes with variations to human pose successfully while the other methods either drifted or failed.

1. Introduction

Computer based Action recognition is a vital step in automated understanding of Human behavior. Object detection⁽⁶⁾⁽⁷⁾⁽⁸⁾⁽⁹⁾⁽¹⁴⁾⁽¹⁶⁾, object tracking⁽¹¹⁾⁽¹⁰⁾⁽¹⁵⁾ and action recognition are the inter-related building blocks that are individually sufficient, but collectively efficient for action recognition. Many online training methods that bootstrap from the first frame have been proposed⁽¹⁰⁾⁽¹³⁾⁽¹⁷⁾⁽¹⁹⁾⁽²⁰⁾, but

they do not cope with appearance changes owing to the limited prior knowledge. Arbitrary, object tracking is gaining popularity but still class specific tracking finds application in surveillance, digital camera, and behavior analysis. There are many challenges in object tracking, such as appearance changes, partial or full occlusion, rapid object movement, cluttered background and real-time processing. Color histograms based methods like Mean Shift and top down approach based on Particle Filter show promising results in cluttered backgrounds, but drifts in case of similar colored background or low illumination. Methods like "tracking by detection"⁽¹⁰⁾⁽¹⁹⁾ and "tracking by continuous recognition"⁽¹³⁾ that use a discriminative classifier trained online, and is robust against scale and illumination changes. But since the detector is updated based on samples in the prior frame, detection errors propagate resulting in a drift. In this paper we propose the following,

- Online Real boosting

It is an improvement over "Online Boosting" that utilizes Real Adaboost for offline training, and probability densities for the object and background thereby reducing drift. The classifier performance improved, and performance equalling or exceeding the state-of-the-art is achieved with half the number of weak classifiers resulting in processing speed improvement.

- Soft Decision Features

Object tracking deals with articulate and non-rigid objects, where movement is associated with changes to appearance. Features trained for a particular shape would fail, and hence termed as rigid or "Hard Decision Features". An online training algorithm for training pose invariant features termed as "Soft Decision Features" (SDF) is proposed. To track objects like human with pose variations, a online training method that trains features which can adapt to pose variations (SDF) is proposed.

In this paper, Online Real Boosting is discussed in Section 2, while Section 3 discusses about the features flexible features that supports pose changes, and Section 4 summarizes the finding and concludes.

2. Online Real Boosting for Object Tracking

Boosting is a popular ensemble learning method for object detection, where a

^{†1} Chubu University
^{†2} Omron Corporation

set of weak classifiers are trained and linearly combined to form a strong classifier. Oza et al.³⁾ proposed an Online Boosting method that trained classifiers online with the current view of the object to update the weights of the weak classifiers. During each update even small errors would propagate resulting in a drift Grabner et al.¹⁰⁾ proposed semi-supervised Online Boosting algorithm based on Adaboost, which achieved good tracking results against occlusions and cluttered background, but the large number of weak classifier trained for achieving good results is a computational overhead. In this paper, we propose an Online Boosting algorithm similar to Grabner et al.¹⁰⁾, but replace Adaboost with Real Adaboost. A probability density for the object and the background is associated with each of the weak classifier that significantly improves the classification performance. First, we introduce original Online Boosting then a discussion on Grabner's method followed by the discussion on the proposed method.

2.1 Online Boosting

Online Boosting was proposed by Oza et al.³⁾, and improved by Grabner et al.¹⁰⁾. In Online Boosting, the training sample is provided only once to the learner and discarded afterwards. The weak classifiers are updated online each time a new training example arrives. Using the classifier tracking is achieved by exhaustive frame search with sliding window or particle filter.

Offline supervised learning is performed to train weak classifiers with least error ϵ , which is calculated with the weights $D(x_i)$ and sample images x_i as shown in Eqn (1).

$$h_t = \arg \min_{h_j \in H} \epsilon_j = \sum_{i=1}^m D_t(x_i) [y_i \neq h_j(x_i)] \quad (1)$$

Weak classifier weight α_t is based on the classification errors and calculated as shown in Eqn (2),

$$\alpha_t = \frac{1}{2} \log \frac{1 - \epsilon_t}{\epsilon_t} \quad (2)$$

The weights $D(x_i)$ are updated with the weak classifier h_t response,

$$D_{t+1} = \frac{D_t \exp(-\alpha_t y_i h_t(x_i))}{Z_t} \quad (3)$$

The above process is repeated for learning T weak classifiers, and a strong clas-

sifier $H(x)$ is obtained as shown in Eqn (4),

$$H(x) = \sum_{t=1}^T \alpha_t h_t(x) \quad (4)$$

In online learning, there is only one positive sample available at any time. The weight α_t is updated by Eqn.(2) and error rate of positive and negative samples in current frame.

$$\epsilon_t \leftarrow \frac{\lambda_t^{sw}}{\lambda_t^{sc} + \lambda_t^{sw}} \quad (5)$$

Note that λ_t^{sc} and λ_t^{sw} are importance factors of weak classifier h_t for correct classification and misclassification, respectively. On correct classification, λ_t^{sc} is updated to $\lambda_t^{sc} + \lambda$, λ_t^{sw} is updated to $\lambda_t^{sw} + \lambda$, otherwise.

In the original Online Boosting proposed³⁾, weights for all the weak classifiers are updated without any feature selection, however, effective features are selected and their weights were updated for tracking with a selector approach in modified online boosting proposed by Grabner et al¹⁰⁾. This method tracked occluded objects, and to some extent objects that changed its appearance. The performance of this method is proportional to the number of weak classifiers. An increase in number of weak classifiers would improve the tracking performance, but computational costs would also increase. The detector is trained priori and tracking might fail on appearance changes that are not part of the training samples.

2.2 Proposed method

The proposed method focuses on the training method and the feature type. For example, to track frontal faces, Haar-like features are effective, but for side facing faces, contour extracting edge features are effective. To track scenes with facial pose changes, Haar-like, ABS Haar-like, EOH (Edge Orientation Histograms) and Edgelet features are effective, and feature selection is performed from a pool of features trained offline. To select weak classifier with high discriminative ability, Real Adaboost is employed to select features, which outputs real value as likelihood. The proposed method is shown in Fig. 1. In the proposed method the detector is trained offline with Real Adaboost. The weak classifiers trained offline are grouped into N selectors, based on the training sample a weak classifier is trained online from each of the selector group, and linearly combined into a

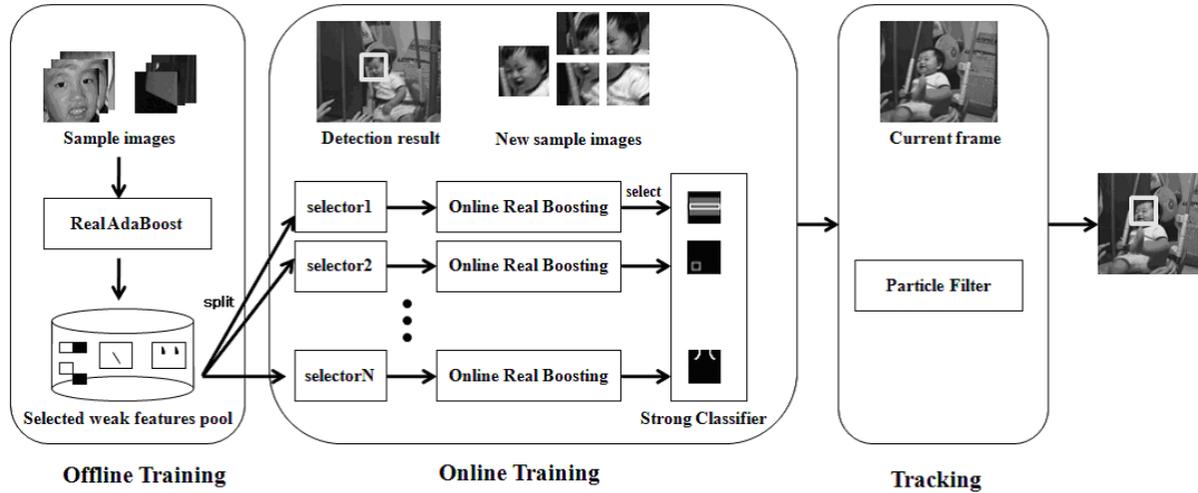


Fig. 1 Online Real Boosting.

strong classifier. The strong classifier is used as a detector the next frame. Object is tracked either with particle filter approach.

2.3 Online Real Boosting

Online Real Boosting has the following three steps, preparation for online learning, weak classifier selection, and weak classifier update. The online learning flow for the proposed method is captured in Fig 2. The following sections details the steps.

2.3.1 Preparation for Online Learning

The strong classifier $H_{offline}$ include weak classifiers trained offline, for objects such as face, human body, etc. As shown in Fig.2, the weak classifiers are grouped into the N subsets and termed as a selector. One weak classifier is trained online from each selector group, based on the current training sample. As shown in Fig.2, L samples labeled as positive and negative samples are trained for online learning. As shown in Fig.2, the importance factor λ_n which determines the classification performance of each of the selector is initialized in Eqn.6.

2.3.2 Selection of Weak Classifiers

As shown in Step 4 of Fig.2, one weak classifier from each of the selector is

selected. First, in step 4.1.1, probability density function $W_{n,m}^y$ of weak classifier $h_{n,m}$ is updated with importance factor λ_n as shown in Eqn.(7). The probability density function $W_{n,m}^y$ is obtained from offline learning, representing the positive and negative probability distributions. Positive sample update $W_{n,m,j}^{+1}$, while negative sample update $W_{n,m,j}^{-1}$. The probability density function $W_{n,m}^y$ is implemented as a histogram, and the histogram index bin_j decided from the feature value. After updating the probability density function with all the samples, the hypothesis of each weak classifier is updated as shown in Eqn.(8), hence, if the distribution in index bin_j is to be updated based on a sample, then the neighboring indices are also updated based on a Gaussian smoothing function. β is added to prevent divide-by-zero. In step 4.1.3, Eqn. (9), weak classifier are evaluated by the Bhattacharya distance $Z_{n,m}$, and the classifier with the least $Z_{n,m}$ is selected from the selector.

2.3.3 Weak Classifier update

The error rate of selected weak classifier is calculated in step 4.3 of Fig.2. In case of correct classification $y \cdot h_n(x)$ is positive and as shown in Eqn.(11), ϵ_n^{+1} is updated, and in case of negative value and misclassification, ϵ_n^{-1} is updated

0. Train strong classifier in offline learning.
 Strong classifier $H_{offline}(x)$ (weak classifier: M)
 Weak classifiers $h_{offline,m}(x)$ $\{m = 1, \dots, M\}$
 1. New sample in current frame $\{(x_1, y_1), \dots, (x_L, y_L), y_i = \pm 1\}$, L : sample No.
 2. Divide M weak classifiers to N (same as selector No.)
 3. Initialize importance factor λ_n

$$\lambda_n = 1/N \quad (6)$$
 Initialize classification error $\epsilon = 0$
 4. For $n=1, 2, \dots, N$ // Selector No.
 4.1. For $m=1, 2, \dots, M/N$ // weak classification No.
 4.1.1. Update probability function
 If $h_{n,m}(x) \in bin_j$

$$W_{n,m,j}^y = W_{n,m,j}^y + \lambda_n \quad (7)$$

 4.1.2. Update hypothesis of weak classifier
 If $h_{n,m}(x) \in bin_j$

$$h_{n,m}(x) = \frac{1}{2} \ln \frac{G(W_{n,m,j}^+) + \beta}{G(W_{n,m,j}^-) + \beta} \quad (8)$$
 G is a Gaussian smoothing function
 4.1.3. Evaluation value for weak classifier selection

$$Z_{n,m} = 2 \sum_{j=1}^T \sqrt{W_{n,m,j}^+ W_{n,m,j}^-} \quad (9)$$
 T : bin No. of histogram
 4.2. Select weak classifier with least Z

$$m^* = \arg \min_m Z_{n,m} \quad (10)$$

$$h_n = h_{n,m^*}$$

as shown in Eqn.(12). From the classification error ϵ_n calculated by Eqn.(13), effectiveness of the weak classifier α_n is calculated by Eqn.(14). In Real Adaboost, each weak classifier outputs a real value as confidence, however, in our method, each of the weak classifier is associated with an effectiveness coefficient α_n , and confidence is calculated as shown in Eqn.(18). The importance factor λ_n for

4.3. Calc classification error ϵ_n of $h_m(x)$
 If $y \cdot h_n(x) \geq 0$

$$\epsilon_n^+ = \epsilon_n^+ + \lambda_n \cdot |h_n| \quad (11)$$
 Else

$$\epsilon_n^- = \epsilon_n^- + \lambda_n \cdot |h_n| \quad (12)$$

$$\epsilon_n = \frac{\epsilon_n^+}{\epsilon_n^+ + \epsilon_n^-} \quad (13)$$
 4.4. Set weight of weak classifier

$$\alpha_n = \frac{1}{2} \log \frac{1 - \epsilon_n}{\epsilon_n} \quad (14)$$
 4.5. Update importance factor of selector
 If $y \cdot h_n(x) \geq 0$

$$\lambda_n = \frac{\lambda_n}{2(1 - \epsilon_n)} \quad (15)$$
 Else

$$\lambda_n = \frac{\lambda_n}{2\epsilon_n} \quad (16)$$
 5. Strong classifier

$$H(x) = \text{sign}(\text{conf}(x)) \quad (17)$$

$$\text{conf}(x) = \sum_{n=1}^n \alpha_n h_n(x) \quad (18)$$

Fig. 2 Online learning of Online Real Boosting.

each of the selector is updated based on the classification error ϵ_n , as shown in Eqn.(15) and Eqn.(16). The importance factor λ_n , determines the classification performance of the selector n . A large value of λ_n , means that the selector has selected an effective classifier for tracking, while a small value of λ_n means otherwise.

2.3.4 Tracking based on Particle Filter using Weak Classifiers

Tracking is achieved by Particle Filter using the strong classifier trained online. To predict the sampling points in Particle Filter, the confidence of strong classifier $H(x)$ is calculated. The particle with the maximum confidence of $H(x)$ is output as tracking result. Weights for each of sampling points is updated based on

Table 1 Comparison Online Boosting and Online Real Boosting.

	Online Boosting	Online Real Boosting
weak classifier	local feature	local feature Probability function of positive and negative
Hypothesis	threshold	Probability function
training method	AdaBoost	Real AdaBoost

confidence of the strong classifier result $H(x)$.

2.4 Proposed Method, compared with Online Boosting

The major differences of the proposed method, Online Real Boosting to that of Online Boosting¹⁰⁾ is captured in Table 1. The major difference is that each of the weak classifier has a probability density function $W_{n,m,j}^y$, and trained by Real Adaboost. Online Boosting select the weak classifiers by re-creating the probability density function of target and background based on the current samples. The weights are updated based on the just arrived sample, and any large changes to appearance will lead to a large difference between the training sample and the target frame, resulting in tracking failure. In the proposed method, the Probability density function is derived from offline training, and updated online based on the just arrived training samples. The probability density function for the positive samples is obtained by offline learning with a sample set containing various facial poses and illumination condition, hence tracking faces with large changes to facial pose and illumination condition is possible.

2.5 Experiments

2.5.1 Comparison of Tracking Performance

The performance of the proposed method compare with Online Boosting. For the experiments, human head is selected as the target object, as it has a wide range of appearance and pose variations. The evaluation dataset from Jepson⁵⁾ was selected. It contains 1145 frames, with various face poses and cluttered background. The detector was trained offline with 5000 positive and 9000 negative samples, normalized to 24x24 pixels size. Both Online Real Boosting, and Online Boosting trained 300 weak classifiers offline. Some examples from the offline training dataset are shown in Fig.3. The training dataset includes various face poses like frontal, side view, etc. The face size is normalized to 20x20 pixels of



Fig. 3 Example of training samples.

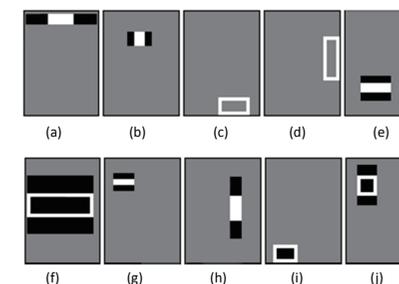


Fig. 4 Weak classifiers trained in offline learning.

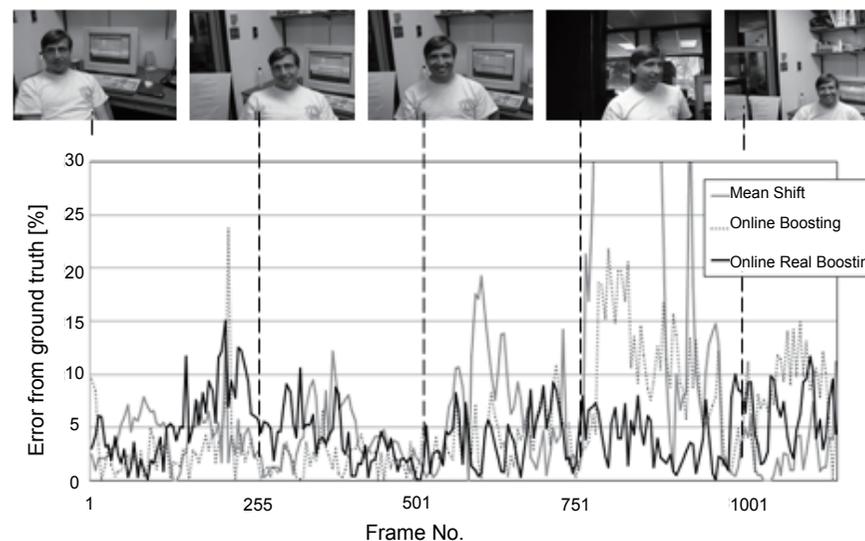


Fig. 5 Tracking Errors in each frame.

the 24x24 pixels, and the rest contains the background. The top 10 weak classifiers learned from offline training is shown in Fig.4. (a), (b), (e), (g), and (h) are Haar-like features, (c) and (d) are EOH features, and (f), (i) and (j) are ABS Haar-like features.

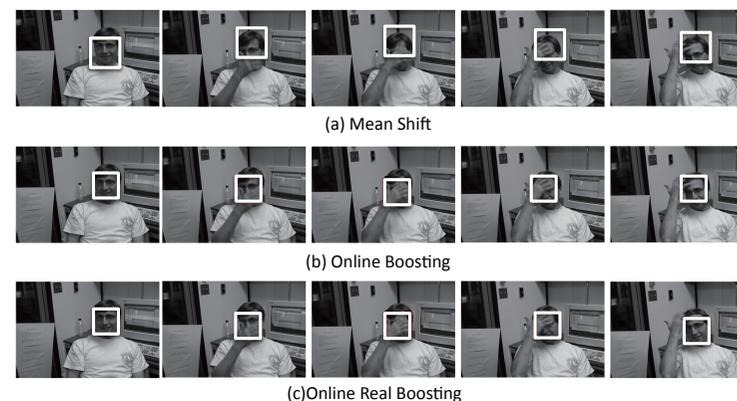
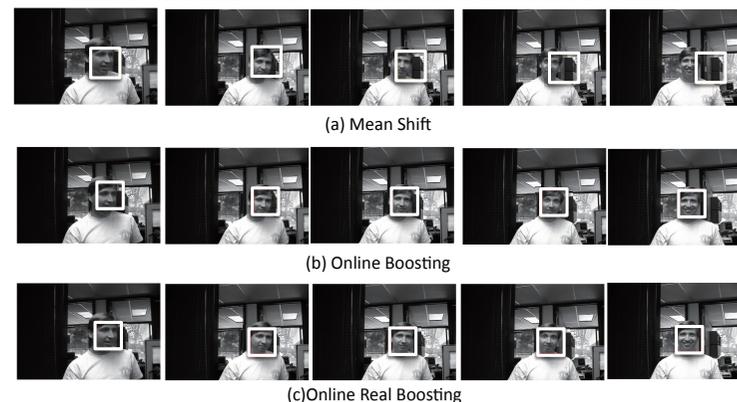
Table 2 Comparison of tracking performance.

	Average Error	SD
Mean Shift	11.2%	20.1
Online Boosting	9.8%	6.1
Online Real Boosting	6.8%	5.7

The 300 weak classifiers trained offline are grouped into 100 selectors, and one weak classifier is selected online from each selector during training. The number of particles in the Particle Filter is set to 200, and as shown in Fig.2, Eqn.(18), the locations with the highest confidence are selected. Both the methods were initialized with the same set of locations. Table 2 captures the tracking performance for both the methods and Mean Shift. Error is defined as the distance between the actual object center to that of the tracked result. The average and standard deviation on errors are captured and the figures indicates that the proposed method outperforms the other two. A very small standard deviation for the proposed method indicates that during occlusion and changes to facial expression, the tracker would not drift. Fig.5 captures the tracking errors on a frame-by-frame basis indicating that the proposed method outperforms the other two for almost all the frames. Tracking results for all the 3 methods are shown for few frames around the 200th and 800th frame for all 3 methods in Fig.6 and Fig.7, respectively. Fig.6 and Fig.7, (a) is for Mean Shift, (b) for Online Boosting and (c) for Online Real Boosting. In the 200th frame, face is occluded with the hand, in this case, Mean Shift exhibits a large drift, while online boosting and the proposed tracks correctly. In the 800th frame, the object to be tracked is seen in front of a cluttered background. In this case, Mean Shift drifts to similar objects in the background, classification errors of the weak classifiers increase in Online Boosting, and hence the tracking position drifts right. On the other hand in the proposed method, the tracking coordinates almost matches the correct coordinates, owing to the reduced classification errors proving the effectiveness of the proposed method.

2.5.2 Selector - Performance Comparison

The performance of the proposed method and Online Boosting was compared against the number of selectors and weak classifiers. Fig.8 captures the effect of changing the number of selectors and weak classifiers on tracking errors. The

**Fig. 6** Tracking result of each methods in around 200 frame.**Fig. 7** Tracking result of each methods in around 800 frame.

tracking errors decrease with the increase in number of selectors. As observed from Fig.8, a consistent difference in performance can be observed, where the proposed method outperforms Online Boosting. For example, average error rate from the actual location for the proposed method is a little over 8% for the proposed method, but the error rate falls just below 10% for Online boosting with 90 selectors, a 3 fold increase. This plot indicates that the proposed method

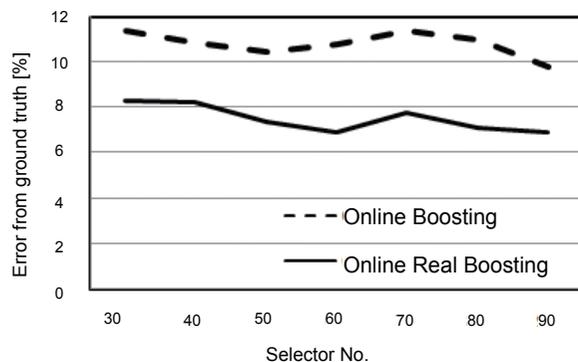


Fig. 8 Error comparison of number of selector.

Table 3 Performance comparison by number of selectors.

	Average Error	SD
Mean Shift	12.0%	8.6
Online Boosting	9.5%	6.3
Online Real Boosting (Selector : 100)	8.2%	5.1
Online Real Boosting (Selector : 50)	8.8%	5.5
Online Real Boosting (Selector : 30)	10.0%	6.4

outperforms Online boosting for various the number of selectors.

2.5.3 Performance in a Scene with Pose Variation

Human face pose changes to a large extent when capturing a scene with a hand-held camera, especially when capturing a movie of a kid. A scene with such a scenario was selected for comparing the performance for various number of selectors. The evaluation dataset is a 500 frame movie with face pose variation of a kid. Table 3 captures the average error rate and the standard deviation. Fig.9 and Fig.10 show the tracking results for Online Boosting and the proposed method respectively. In both Fig.9 and Fig.10, (a) shows the results for 30 selectors, (b), (c), (d) and (e) are for 50, 70, 90 and 100 selectors respectively.

From table 3, tracking error rate of the proposed method is less compared to

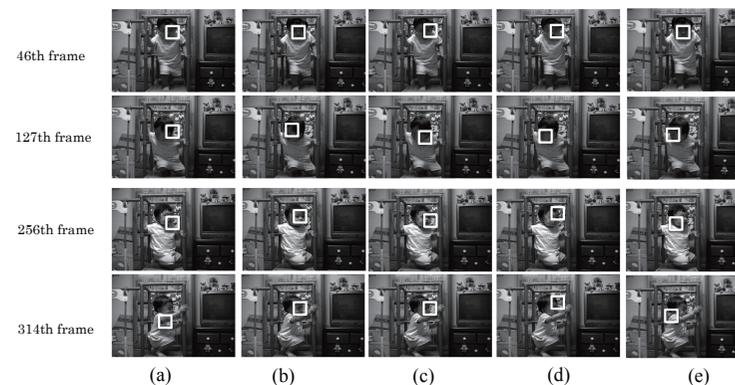


Fig. 9 Tracking performance of Online Boosting in face pose variation scenario.

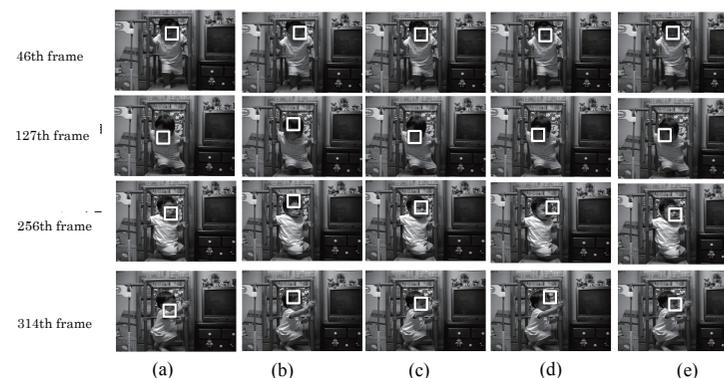


Fig. 10 Tracking performance of Online Real Boosting in face pose variation scenario.

Mean Shift and Online Boosting. The evaluation dataset has upright face around the 46th frame, and left-downward facing face around the 127th frame, and right-upward facing face at the around 265th frame. On a evaluation data with these face poses, Online Boosting tracked successfully with a large number of selectors. The proposed method achieved matching performance with 50% of the selectors which proves that the proposed method achieves matching performance with just half the number of selectors.

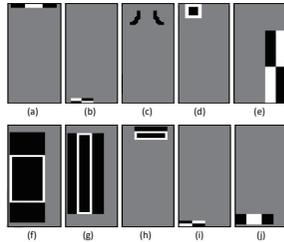


Fig. 11 Weak Classifier of human data in offline learning.

2.5.4 Human Tracking Performance

The proposed method can track an arbitrary object trained offline. Human body with wide appearance and pose variation was set as the target object. A detector with 300 weak classifiers was trained offline with 6700 positive samples and 10000 negative samples. Fig. 11 shows the top 10 weak classifiers, and (c) shows the Edgelet feature. Edgelets were not selected for face detector, but selected for Human detection as it captures the the shoulder area. Fig. 12 and Fig. 13 shows human tracking result for the proposed method. Fig.12 shows a human changing pose indoors, and the resulting tracking bounding box show that the proposed method can track human body with change in appearance. Fig. 13 shows a scene with low illumination condition where the proposed tracks successfully.

2.6 Summary

In this section, we proposed a new Online Boosting method that trained weak classifiers with Real Adaboost and a probability density function of positive and negative associated with each classifier to obtain better classification performance. The method also employs various feature types to reduce the number of weak classifiers. The method was compared with Online Boosting and Mean Shift, for tracking a human head with a dataset of facial images in a cluttered background, and the results indicate that the proposed method outperforms the other methods. The proposed method was also applied for tracking human body and it showed robust performance for both changes in appearance and illumination.



Fig. 12 Tracking performance in a scene with human pose variation scenario.



Fig. 13 Tracking performance in a low illumination scenario.

3. Soft Decision Features for Object Tracking

In Object tracking, appearance change associated with an object pose is one of the critical issues. In Online Real Boosting, features are trained offline and grouped into selectors. In the case of human tracking, weak classifiers should be trained for an array of possible human poses like sitting, standing, bending, etc. which is not a simple task. In this section, we describe a new structure of weak classifier that are flexible to appearance changes of target object.

3.1 Problem of Online Boosting

Online Real Boosting selects a set of weak classifiers and linearly combine them for detecting the target object from a candidate pool trained offline. To track non-rigid objects like human body, which can assume various poses, the candidates should be trained offline for all the poses. In the case of human tracking, the detector shall detect sitting and standing (upright) poses. It is practically impossible to exhaustively select weak classifiers for all the poses that the human body can assume. An attempt to exhaustively detect all the human poses by brute force is limited by the number of classifiers. An increase in the number of supported poses would increase the computational costs. Hence most of the methods limit the number of poses supported. Tracking fails for the poses that are not learned offline. The weak classifier is hard-coded for a particular

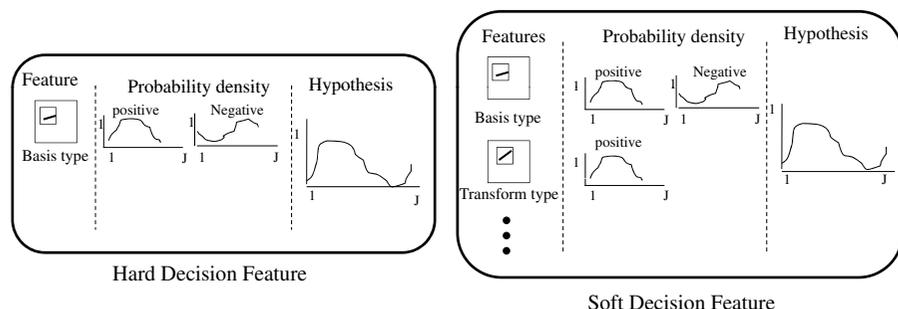


Fig. 14 SDF Structure.

object pose, and is termed as a Hard Decision Feature (HDF).

3.2 Object tracking applied pose variation

To track human with Online Real Boosting, a framework of weak classifiers that adapt flexibly to change in appearance is proposed. The weak classifiers trained online flexibly adapts to the appearance changes in the target object, and these features are termed Soft Decision Features (SDF), this section discusses the offline and the online training method of SDF.

Fig.14 shows the proposed SDF. The Online Real Boosting that was proposed in the previous section, trains a weak classifier with feature for a particular object shape, a hypothesis, and a Probability density function. The proposed method trains a weak classifier with a set of transformable features, probability density function for the object and background and a hypothesis function. Transformable features are divided into 2 types based on the supported object shape. "Basis type" is the feature that supports the original object shape, and "Transform type", that supports the transformed shape. The shapes for the "Transform type" are obtained during offline training by transforming the Basic type using a transform G as shown in Eqn.19,

$$F_{trans} = G(F_{org}; p, s) \quad (19)$$

where F_{trans} represents the transformed shape, and F_{org} is the Basis type, p and s represent position and scale respectively. The transformed shapes is grouped with the basis for offline training of the SDF feature. During online

Input: Training samples $X = (x_n, y_n); n \in [1, N], y_n \in [-1, +1]$
 Output: Strong classifier $H_{off}(X)$

1. Initialize weight of training samples

$$\lambda_n = 1/N \quad (20)$$
2. For $t = 1, \dots, T$ // Training No.
 - 2.1. For $m = 1, \dots, M$ // Weak classification candidate No.
 - 2.1.1. Probability density function W_{org}^y

$$W_{org,m,j}^y = W_{org,m,j}^y + \lambda_n \quad (21)$$
 - 2.1.2. Transform weak classifier

$$F_{trans} = G(F_{org}; p, s) \quad (22)$$
 - 2.1.3. Probability density function W_{trans}^y

$$W_{trans,m,j}^y = W_{trans,m,j}^y + \lambda_n \quad (23)$$
 - 2.1.4. Calc similarity

$$S_m = 2 \sum_{j=1}^J \sqrt{W_{org,m,j}^{+1} W_{trans,m,j}^{+1}} \quad (24)$$
 - 2.1.5. Merge probability density functions
 if $S_m > \theta$

$$W_{m,j}^{+1} = W_{org,m,j}^{+1} + W_{trans,m,j}^{+1} \quad (25)$$
 - Repeat 2.1.2 - 2.1.5 to grouping
 - 2.1.6. Calc Evaluation value Z

$$Z_m = 2 \sum_{j=1}^J \sqrt{W_{m,j}^{+1} W_{m,j}^{-1}} \quad (26)$$

training an optimal shape is selected from a set of shapes for tracking.

3.3 Offline training for SDF

Offline training for SDF is shown in Fig.fig2. The following section discusses each one of the steps involved in the training, and its difference to Online Real Boosting.

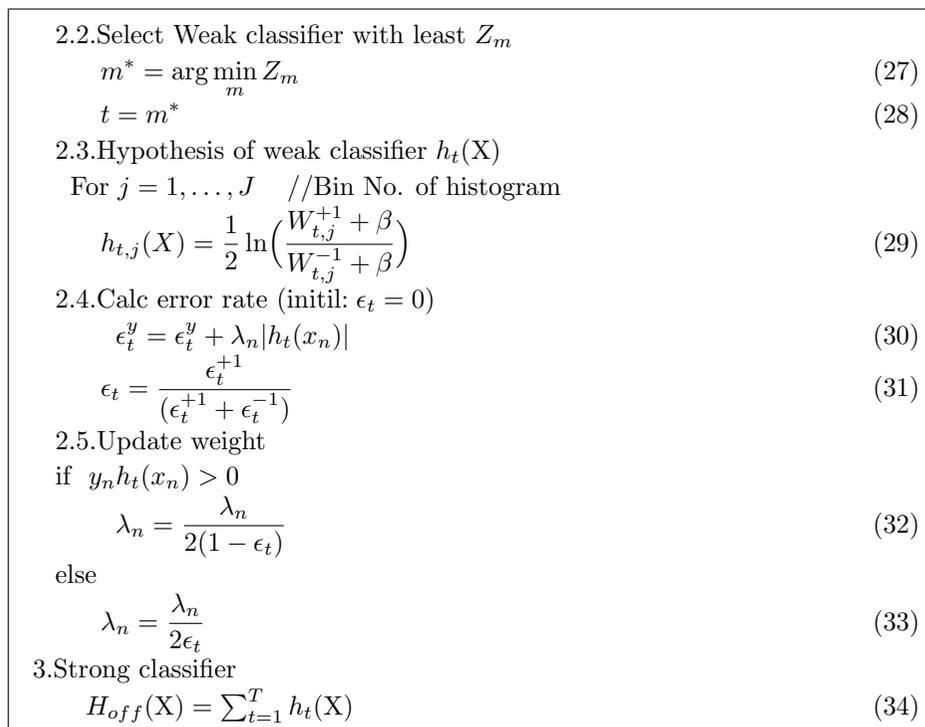


Fig. 15 Offline training using SDF.

3.3.1 Preparation for Offline training

Training samples are labeled, $X = (x_n, y_n; n \in [1, \dots, N], y \in [-1, +1])$ as positive ($y = +1$) and negative ($y = -1$) samples, and weight for all the samples are initialized to $\lambda_n = 1/N$. In offline training from M candidates, T weak classifiers are selected. Weak classifiers are selected as shown Fig. 15 steps 2.1 to 2.5 and the above steps are repeated in a loop till a predefined number of weak classifier is reached.

3.3.2 Weak classifier candidates Generation

First, for each weak classifier, the probability density function, W_{org}^y for positive and negative classes are determined, where the probability density function

differentiates the object from the background, and defined by histogram. As shown in Fig.15, step 2.1.1, the feature value obtained for a training sample x_n is calculated as the index for adding λ_n into the histogram, and the probability density function is obtained.

3.3.3 Grouping of transformed shapes

After generating the candidates, the transformed shapes F_{trans} are generated from the F_{org} as shown in Step 2.1.2. The transformation parameters are adjusted to deduce the transformed shapes and a probability density function is defined for each one of them, similar to that of the basis type as shown in 2.1.1. The similarity of the probability distribution for the basis and the transform types is defined by the Bhattacharya distance, as shown in Step 2.1.4. Transformed shapes with similarity values greater than the threshold θ is selected as SDF in step 2.1.5. The parameters for the transformation function G are adjusted to deduce the transformed shapes F_{trans} and steps 2.1.2 through 2.1.5 are repeated for all F_{trans} . Similarly grouping is performed for all the weak classifier candidates. The similarity of positive and negative classes Z is calculated as shown in equation 2.1.6.

3.3.4 Weak classifiers selection

As shown in Step 2.2, the weak classifier with a minimum Z score is selected from the candidates. For the selected weak classifier, the weak hypothesis $h_t(X)$ is calculated with the probability density function W_t^y for the positive and negative class.

3.3.5 Updating weight and Strong classifier

As shown in Step 2.4, the error rate for a weak classifier is calculated using the all training samples, and the weights λ_n are adjust as shown in Step 2.5. Based on this, for the selected classifier, weight of misclassified samples are incremented. The steps are repeated and T weak classifiers are selected and linearly combined to form the strong classifier $H_{off}(X)$.

3.3.6 Difference from Online Real Boosting

The difference of the proposed method to that Online Real Boosting is in Steps 2.1.2 through 2.1.5 shown in Fig.15. In the proposed method, during offline training, shapes with similar probability density function are grouped into a single weak classifier. The similarity of the probability density function implies

that the hypothesis function of the weak classifier is also similar. Transformation types can be obtained by locally constraining the parameters in transformation process in Step 2.1.2, and transforming the basis type.

3.4 Online learning of SDF

Online training flow for SDF is shown in Fig.16. The proposed method consists of steps 3.1 and 3.2 where the probability distribution for the the basis type is updated and optimal SDF classifier is selected, steps 3.3 and 3.4, where the weak classifiers are optimized and shapes are selected. Online Real Boosting is made only of steps 3.1 and 3.2, and in the proposed steps 3.3 and 3.4 were added for shape optimization. The following sections details the steps involved in online training.

3.4.1 Preprocessing of online learning

The target object in the just arrived frame is taken as the positive sample ($y = +1$) and the background as negative sample ($y = -1$). The T number of classifiers selected offline are divided into R random groups, similar to the earlier method. Selecting SDF from each group might be computationally intensive if there are many candidate features in each of the groups. To improve on processing speed, Online Real Boosting divided the groups randomly and constrains the number of SDF in each group.

3.4.2 Selection of SDF

The weights for the training samples λ_k are initialized as show in Step 2, Fig.16. Step 3 is executed for all the divided groups, and a weak classifier from each group is selected. At first, the weights for the samples are used to update the probability distribution function $W_{r,m}^y$ weak classifier candidate m , in group r , as shown in Step 3.1.1. The probability distribution is updated similar to offline training, i.e., the feature values of the samples are used to update the corresponding histogram bin by adding the weights λ_k . The histogram is normalized as shown such that the area under the curve is 1. The variance Z for the weak classifier candidate is calculated as shown in Step 3.1.2. and as in Step 3.2, the classifier with the least value of variance, Z is selected.

3.4.3 Shape optimization

In Section 3.3, Shape optimization is performed for the the weak classifier selected. In Steps 3.3.1, and 3.3.2, the probability density function of the weak

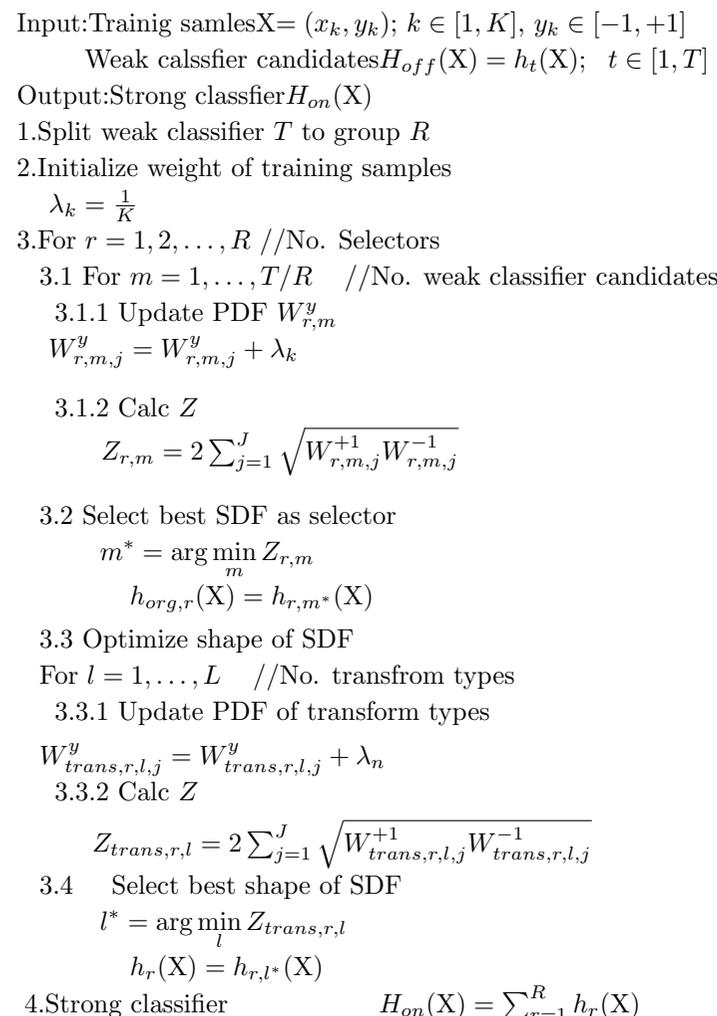


Fig. 16 Online training for SDF.

classifier's transformed shape is updated, and Z is calculated. As in Section 3.4, the model with minimum Z is selected as the selector. Step 3 is executed for all groups, and the selectors are linearly combined into a strong classifier $H_{on}(X)$. In Online Real Boosting, the weak classifiers got from Step 3.2 are used as the selector. In the proposed method, the candidate weak classifiers are made up of various shapes, and the shape that suits the target object are selected ensuring a shape invariant robust tracking method.

3.4.4 Feature type of SDF

An SDF is a weak classifier that consists of features with similar probability density function. As the weak classifiers are made up of similar shaped features, it can track objects with little appearance change. Offline training for SDF and optimal shape selected during online training was discussed. Features form the basis for implementation, and in this paper Edgelet and EHOg are selected as they perform very well for Human detection.

3.4.5 SDF based Particle Filter Tracking

In object tracking, searching methods includes the sliding window that exhaustively slides through the given region, and Particle Filter with a probability distribution for arriving at the sampling points. Performance of sliding window can be improved by applying the classifier entire image, which might be computationally expensive and would impact performance. Methods based on sampling assigns weights to each of the sampling points based on the likelihood previous frame, thus limiting the sampling region. Sampling based methods outperform sliding window based approaches. The tracking performance is based on the prior probability, and for most of the cases it achieves reasonable performance. The proposed method utilizes Particle Filter to achieve high processing speed. Object location (x, y) and scale s are the state vectors for the particle filter. The state model is a random walk where each particles are randomly distributed based on a 2D normal distribution. The sampling region width is taken as the 2 times the width of the target object with respect to the object center in the previous frame, and scale factor is set to ± 0.1 of the original size. The weights for each of the particle is calculated based on the weights of the classifier output got from online training.



Fig. 17 Training samples in online learning.

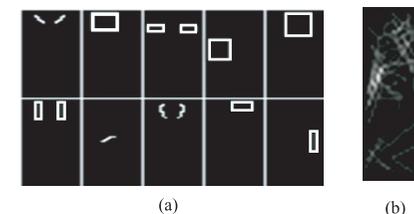


Fig. 18 Example of selected features in online learning: (a) Top 10 weak classifiers, (b) Feature distribution of all weak classifiers.

3.5 Experiments

3.5.1 Experiment Overview

SDF and HDF based Online Real Boosting was compared for tracking performance. Errors in tracking were compared for around 800 Frames that included appearance and pose variations.

Initially, 500 weak classifiers were trained offline with 5000 training samples, normalized to 24×60 pixels. Fig. 17 shows some of the samples from offline training set. The sample set contains standing humans, while other poses like sitting present in the test data set is absent. Parameters in the transformation function are changed, Basis type is adjusted by ± 1 pixel, and the size s is changed between -0.8 to 1.2 times of the Basis type's width with 0.1 increments. The oriented gradient g changes the HOG elements by ± 1 . The parameters are changed for experimental purposes, and by large changes to the parameters shapes that are further apart would be grouped which might lead to a grouping of shapes that vary to a large extent from that basis type. The grouping threshold θ was experimentally set to 0.8 .

Fig. 18 shows the top 10 weak classifiers trained offline and the classifier pattern distribution. In Fig. 18(a) the features with rectangles are the HOG, and the features with lines are the Edgelet. Fig 18(b) shows the overlapped feature distribution of all the features both like Edgelet and EHOg. The distribution shows that the features are concentrated around the head and along the body contour. The average number of shapes in an SDF is 3.5 . For both SDF and HDF based Online Real Boosting, 500 weak classifiers were trained offline, and

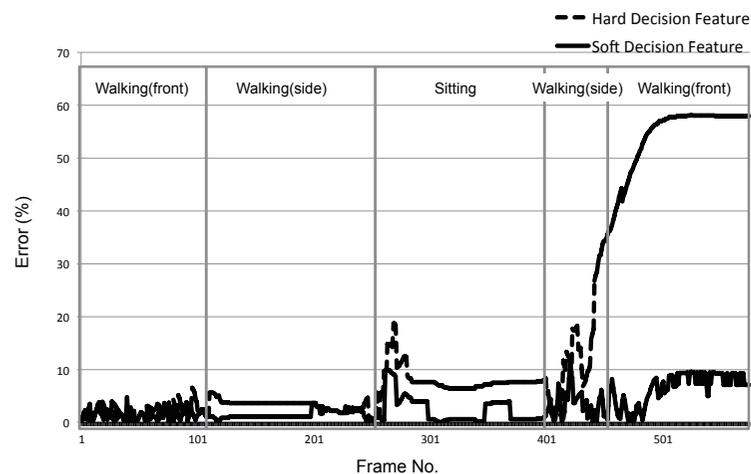


Fig. 19 Tracking performance for each group.

the number of selectors was set to 50. Each selector selects a weak classifier from the random grouping of 500 weak classifiers. The number of groups is same as the selector, 50, and each group contains 10 weak classifiers. Both sliding window and Particle Filter based approach were adopted for verifying the tracking performance of the proposed method against pose and positional changes. Sliding window was employed over a window 2 times the width of the object detected in the previous frame. The Particle filters set the object center detected from the previous frame as the center. The number of particles was set to 100. Tracking performance was compared as the accuracy of tracking location. Humans in the evaluation image set were marked manually with a bounded rectangle beforehand. Tracking performance is defined as the distance between the centers of the tracking result and the correct rectangle. Positional performance Pa is calculated as shown in Eqn (35),

$$Pa = \text{Dist}(\text{Tracking}, \text{GroundTruth}) / \text{Width}[\%] \quad (35)$$

3.5.2 Tracking performance comparison

Tracking performance was compared with snippets of human body that included pose and direction changes. The snippets contained human body in a

Table 4 Tracking Performance comparison (Sliding Window).

Pose variation	SDF		HDF	
	Error[%]	SD	Error[%]	SD
(a)standing(front)	1.17	0.81	1.13	1.03
(b)standing(side)	1.62	1.45	2.46	2.35
(c)sitting	1.33	2.69	4.24	7.22
(d)standing(front)	1.91	4.74	Miss	-
(e)standing(side)	1.90	3.89	Miss	-
total	1.58	2.71	6.46	38.97

Table 5 Tracking Performance comparison (Particle Filter).

Pose variation	SDF		HDF	
	Error[%]	SD	Error[%]	SD
(a)standing(front)	1.73	1.17	1.84	1.91
(b)standing(side)	1.83	4.26	3.95	9.72
(c)sitting	2.10	4.12	7.34	4.08
(d)standing(front)	3.07	4.64	Miss	-
(e)standing(side)	2.70	4.19	Miss	-
total	2.28	3.76	7.52	39.99

frontal standing pose, and also sideways standing, sideways sitting, and a sequences like standing, sitting and standing up again.

Fig. 19 captures the tracking performance of frames with a sliding window in chronological order. The performance of SDF and HDF matches for the first 200 frames. SDF performs better for sideways standing and for sitting pose. In the later part of the snippet, for sitting, walking and then eventually frontal walking, HDF fails. For the same scenario, tracking errors in SDF also increases to some extent, but tracking succeeds.

Tables 4 and 5, captures the average and Standard deviation on errors for tracking location for each of the pose in both Sliding window and Particle Filter based methods. As seen in Table 4, standard deviation(SD) changes for human pose changes, while the average errors do not vary much. For HDF, for changes in pose the average errors vary to a large extent and while sitting, tracking fails. As shown in Table 5, the positional errors for Particle Filter do not vary much compared to the sliding window approach. This shows that Particle Filters with higher processing can be adopted over sliding windows.

Tracking result of SDF and HDF with Particle Filter is shown in Fig.21 and



Fig. 20 HDF based tracking: Results for (a)Walking(Frontal), (b)Walking(Side), (c)Sitting, (d)Walking(Side), (e)Walking(Frontal).



Fig. 21 SDF based tracking: Results for (a)Walking(Frontal), (b)Walking(Side), (c)Sitting, (d)Walking(Side), (e)Walking(Frontal).

Fig. 21, respectively. As seen in Fig.21(c), SDF tracks correctly even for a sitting pose. In Fig.20(c) a large deviation from the target is seen for HDF tracking and Fig.20(d) and (e) shows that following the failure, tracking also fails for walking. This error is a result of the probability density function not being updated correctly, due to a large deviation in the tracking position. In the proposed method, tracking succeeds even for sitting poses indicating that the probability density function was updated correctly. As shown in Fig. 21(d) and (e) which is also proven by the fact that tracking succeeds for the subsequent walking pose.

This shows that the proposed method that trained features offline only for upright poses, also flexibly tracks sitting pose, which proves that the method is pose invariant to some extent.

3.5.3 Generalization of Pose changes

The previous sections showed that the tracking performance of the proposed method, is robust even for snippets with sitting pose. This section explains the tracking performance results with other poses. To test pose variations, a snippet with a person jumping, and to test occlusions, a snippet from a mogul competition were used. The performance results are shown in Fig. 22 and Fig. 23. The results in Fig. 22 and Fig. 23 shows that proposed method is robust



Fig. 22 Pose variation with sudden movements.

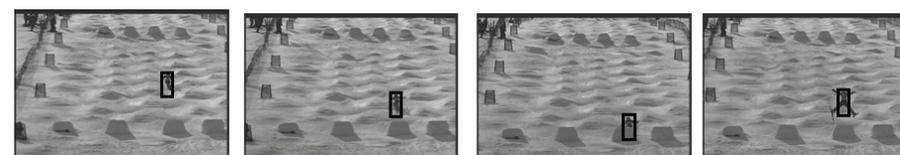


Fig. 23 Pose variation with occlusion.

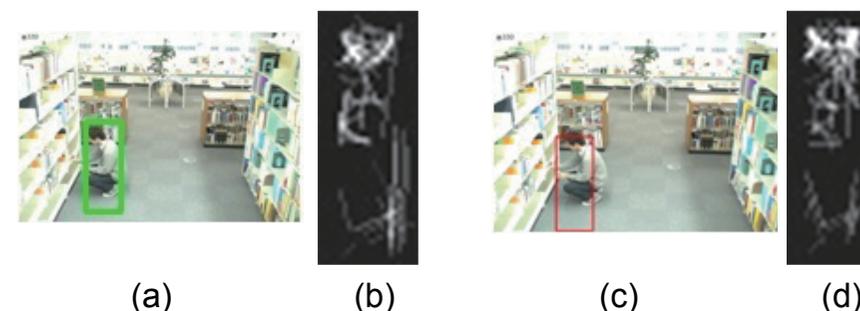


Fig. 24 Selected weak classifiers: (a) and (c) Tracking (sitting) for SDF and HDF (b) and (d) Feature distribution (sitting) for SDF and HDF.

against pose variations and occlusions.

3.6 Comparison with earlier method

3.6.1 Comparison of features selected

Tracking result for SDF and HDF, and the corresponding feature distribution for sitting is shown in Fig. 24. Fig. 24(a) captures the tracking performance for SDF, and Fig. 24(b) shows the feature distribution. Fig. 24(a) captures

Table 6 Weak Classifier count and HDF tracking performance.

Pose variation	SDF		HDF(same weak no.)		HDF(2pose)	
	error[%]	SD	error[%]	SD	error[%]	SD
(a)standing(front)	1.73	1.17	1.63	1.60	2.65	3.01
(b)standing(side)	1.83	4.26	2.88	5.93	4.02	4.56
(c)sitting	2.10	4.12	4.95	4.87	10.01	15.91
(d)standing(front)	3.07	4.64	6.64	5.54	14.32	21.23
(e)standing(side)	2.70	4.19	6.23	5.81	17.71	17.66
total	2.28	3.76	4.46	4.75	9.74	12.74

the tracking performance for HDF, and Fig. 24(b) shows the feature distribution. The features are overlaid over each other, and the degree of brightness in the feature distribution indicates the density of overlaid lines. For sitting pose, features for SDF are concentrated around the head and back and some features are scattered around the feet, and hence the entire human body can be extracted with the features. On the other hand for HDF, as seen in Fig. 24(d), the features are concentrated near the head, and hence the deviation in the result indicating that, HDF selected weak classifiers for fixed regions, for example head in the above example, and hence features are concentrated around the head but sparsely distributed on other parts. The proposed feature is trained offline with the transformed models that are grouped for similar poses. Hence, the features are more scattered adapting to pose changes. The difference is due to the effect of probability density function update. In the proposed method, weak classifiers that are robust towards pose changes are selected, and the probability density function is updated correctly. This is shown by the fact that tracking works in the proposed method even after encountering a sitting pose. On the other hand, HDF selects a classifier with high probability, and ignore the pose change, which results in a wrong update of the probability density function, and tracking fails for pose variations like sitting.

3.6.2 Tracking performance of No.of Weak Classifier, and Pose Variations

The number of HDF candidate weak classifier is made equal to the number of grouped shapes in the SDF (1750), and trained offline for sitting and standing poses. The tracking performance in this scenario was compared. 500 weak classifier were trained offline with 2000 images in sitting pose, while the number

of standing pose images is same as the previous experiments. The weak classifiers selected for various poses were randomly grouped with parameters similar to prior experiments. The comparison result is shown in Table 6. The results show that an increase in weak classifier resulted in performance increase, but even after this increase, SDF based method performs better. When samples with 2 pose were used for training, the performance dropped. In HDF based method, when the number of weak classifier is increased, or the number of pose for training is increased to 2, most of the features concentrate around the head whose pose does not vary much. Due to this, tracking deviates for sitting poses, and their is a decrease in performance in the vertical direction. The proposed method groups weak classifier for the transformed shapes, hence even during appearance changes SDF does not concentrate over a single area. And this enables SDF to achieve good tracking performance for objects with varying poses.

3.6.3 Comparison of Processing Speed

Tracking speed for shape invariant SDF and that of HDF for a particular shape is compared. Test parameters for HDF are similar to the one described in the previous section, the number of candidate weak classifiers trained offline is made equal to the number of grouped shapes in the SDF for both sitting and standing poses. The strong classifier is a linear combination of 500 weak classifiers. Table 7 captures the processing speed for a Intel Core 2 Duo 2.4GHz system. The processing speed for the proposed method is around 30ms for a single frame indicating the possibility of real-time tracking. While the HDF performance was 49ms. Since the selection of weak classifiers during online training is computationally intensive, it impacts the processing speed. The processing speed for HDF with 2 pose support is 42ms. This shows that the proposed method improves the processing speed by 28%. The processing time depends on the selector count and the weak classifiers per selector. By training classifiers with high classification performance in offline training, the selector count can be reduced, thereby improving processing speed.

3.7 Summary

In this section, an offline and online training framework was proposed for tracking objects with appearance changes, by using SDF that adapts to changes in appearance. In the proposed method, during online training, the shape of the

Table 7 Comparison of Processing Speed.

	SDF	HDF(same weak no.)	HDF(2 pose)
process time	30ms	49ms	42ms

weak classifier is changed along with that of the target object, and achieved good results by robustly tracking objects with varying poses. The tracking performance results for the proposed method was compared with that of HDF, and the SDF out performed over the former. The transform parameters can be altered to derive SDF that support size and rotation variations.

4. Conclusion

In this paper, we proposed an object tracking method, which is essential for understanding human actions. In chapter 2, a method termed Online Real Boosting with Real Adaboost and probability density function was proposed. Real Adaboost enabled the reduction of number of weak classifier, there by improving the processing speed. Tracking errors were reduced with a probability distribution for the target and background. In chapter 3, shape invariant tracking method based on SDF was proposed. Soft Decision Features selects efficient features for tracking based on shape changes. With this method, tracking performance for pose changing scenes was achieved.

References

- 1) Isard, M. and Blake, A.: Condensation - Conditional Density Propagation for Visual Tracking, *International Journal of Computer Vision*, Vol.29, No.1, pp.5-28, 1998.
- 2) Schapire, R.E. and Singer, Y.: Improved Boosting Algorithms Using Confidence-rated Predictions, *Machine Learning*, No.37, pp.297-336, 1999.
- 3) Oza, N. and Russell, S.: Online bagging and boosting, In *Proc. Artificial Intelligence and Statistics*, pp.105-112, 2001.
- 4) Comaniciu, D. and Meer, P.: Mean Shift: A Robust Approach toward Feature Space Analysis, *IEEE Trans. Pattern Analysis and Image Intelligence*, Vol.25, No.5, pp.603-619, 2002.
- 5) Jepson, A.D., Fleet, D.J. and El-Maraghi, T.F.: Robust Online Appearance Models for Visual Tracking, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.25, No.10, pp.1296-1311, 2003.
- 6) Viola, P. and Jones, M.: Robust real-time face detection, *International Journal of Computer Vision*, Vol.57, No.2, pp.137-154, 2004.
- 7) Levi, K. and Weiss, Y.: Learning Object Detection from a Small Number of Examples: The Importance of Good Features, In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp.53-60, 2004.
- 8) Wu, B. and Nevatia, R.: Detection of multiple, partially occluded humans in a single image by Bayesian combination of edgelet part detectors, In *Proc. IEEE International Conference on Computer Vision*, 2005.
- 9) Zhu, Q., Yeh, M.-C., Cheng, K.-T., and Avidan, S.: Fast Human Detection Using a Cascade of Histogram of Oriented Gradients, In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Vol.2, pp.1491-1498, 2006.
- 10) Grabner, H. and Bischof, H.: On-line Boosting and Vision, In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Vol.1, pp.260-267, 2006.
- 11) Yilmaz, A., Javed, O. and Shah, M.: Object tracking: A survey, In *Proc Journal ACM Computing Surveys*, Vol.38 Issue 4, 2006.
- 12) Huang, C., Ai, H., Yamashita, T., Lao, S., Kawade, M.: Incremental Learning of Boosted Face Detector, In *Proc. IEEE Conf. on International Conference of Computer Vision*, 2007.
- 13) Avidan, S.: Ensemble Tracking, In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Vol.2, pp.494-501, 2005.
- 14) Hou, C., Ai, H., Lao, S.: Multiview Pedestrian Detection Based on Vector Boosting, In *Proc. Asian Conference on Computer Vision*, 2007.
- 15) Li, Y., Ai, H., Yamashita, T., Lao, S., Kawade, M.: Tracking in Low Frame Rate Video: A Cascade Particle Filter with Discriminative Observers of Different Lifespans, In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2007.
- 16) Wu, B., Ai, H., Huang, C., Lao, S.: Fast Rotation Invariant Multi-view Face Detection Based on Real AdaBoost, In *Proc. IEEE Conf. on Automatic Face and Gesture Recognition*, 2007.
- 17) Lim, J., Ross, D., Lin, R., and Yang, M.-H.: Incremental learning for visual tracking, *International Journal of Computer Vision*, Vol.77 Issue 1, 2008.
- 18) Liu, X., and Yu, T.: Gradient feature selection for online boosting, In *Proc IEEE International Conference on Computer Vision*, 2007.
- 19) Grabner, H., Leistner, C., and Bischof, H.: Semi-Supervised On-line Boosting for Robust Tracking?, In *Proc. European Conf. of Computer Vision*, pp.234-247, 2008.
- 20) Babenko, B., Yang, M.-H., and Belongie, S.: Visual Tracking with Online Multiple Instance Learning, In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2009.