Co-training Framework of Generative and Discriminative Trackers with Partial Occlusion Handling

Thang Ba Dinh Gérard Medioni
Institute of Robotics and Intelligent Systems
University of Southern, Los Angeles, CA 90089
{thangdin,medioni}@usc.edu

Abstract

Partial occlusion is a challenging problem in object tracking. In online visual tracking, it is the critical factor causing drift. To address this problem, we propose a novel approach using a co-training framework of generative and discriminative trackers. Our approach is able to detect the occluding region and continuously update both the generative and discriminative models using the information from the non-occluded part. The generative model encodes all of the appearance variations using a low dimension subspace, which helps provide a strong reacquisition ability. Meanwhile, the discriminative classifier, an online support vector machine, focuses on separating the object from the background using a Histograms of Oriented Gradients (HOG) feature set. For each search window, an occlusion likelihood map is generated by the two trackers through a co-decision process. If there is disagreement between these two trackers, the movement vote of KLT local features is used as a referee. Precise occlusion segmentation is performed using MeanShift. Finally, each tracker recovers the occluded part and updates its own model using the new non-occluded information. Experimental results on challenging sequences with different types of objects are presented. We also compare with other state-of-the-art methods to demonstrate the superiority and robustness of our tracking framework.

1. Introduction

Visual tracking is an important and challenging problem in computer vision with various practical applications such as surveillance, robotics, human-computer interfaces. One of the difficult issues is the appearance changes which may come from varying viewpoints and illumination conditions. Moreover, they can be also caused by partial occlusion, a very challenging problem. In this paper, we aim to track an arbitrary object with partial occlusion handling using very limited initial labeled data. The appearance models are learned online using both a generative and a discriminative tracker.

(a) Some partial occlusion cases (b) Occlusion segmentation

Figure 1. Partial occlusion cases and occlusion segmentation

Discriminative methods focus on finding a decision boundary to separate the object from the background [9, 3, 2]. Generative trackers instead only aim at encoding the target appearance. Examples are the histogram-based methods [1, 17] which are simple but effective at solving tracking problems. Another way of building a generative appearance model is to use linear subspaces [21, 13] which gains lots of interest from researchers.

It is established that discriminative classifiers obtain better performance than generative models if there is enough training data [12]. However, generative methods have higher generalization when limited data is provided [18]. One intuitive way of improving discriminative and generative methods is to combine them together in a hybrid way. Several methods [15, 28] have followed this trend by "discriminative training" of a generative model. They optimize a convex combination of the generative and discriminative log likelihood functions to obtain the model. Co-training, originally presented by Blum and Mitchell [4], is another way to combine different classifiers and has been applied in tracking [23, 29, 16].

However, very few methods explicitly address the partial occlusion problem, which is the critical factor causing drift in visual tracking using a single camera. Most of the proposed algorithms try to avoid partial occlusion by using a threshold to stop updating the model whenever it hap-

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
2011		**00-00-2011 to 00-00-2011**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Co-training Framework of Generative and Disciminative Trackers with Partial Occlusion Handling	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Institute of Robotics and Intelligent Systems,University of Southern, ,Los Angeles,CA,90089**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

This research was funded, in part, by MURI-ARO W911NF-06-1-0094. The first author was also supported by the Vietnam Education Foundation. We thank Zdenek Kalal for his help with the P-N Tracker. Co-training Framework of Generative and Discriminative Trackers with Partial Occlusion Handling. To appear in WMVC IEEE workshop on Motion and Video Computing , Jan 5-7, 2011, Kona, Hawaii.

14. ABSTRACT

Partial occlusion is a challenging problem in object tracking. In online visual tracking, it is the critical factor causing drift. To address this problem, we propose a novel approach using a co-training framework of generative and discriminative trackers. Our approach is able to detect the occluding region and continuously update both the generative and discriminative models using the information from the non-occluded part. The generative model encodes all of the appearance variations using a low dimension subspace which helps provide a strong reacquisition ability. Meanwhile the discriminative classifier, an online support vector machine, focuses on separating the object from the background using a Histograms of Oriented Gradients (HOG) feature set. For each search window, an occlusion likelihood map is generated by the two trackers through a codecision process. If there is disagreement between these two trackers, the movement vote of KLT local features is used as a referee. Precise occlusion segmentation is performed using MeanShift. Finally, each tracker recovers the occluded part and updates its own model using the new nonoccluded information. Experimental results on challenging sequences with different types of objects are presented. We also compare with other state-of-the-art methods to demonstrate the superiority and robustness of our tracking framework.

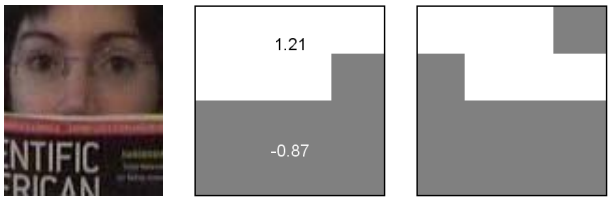
15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**Public Release**	**8**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

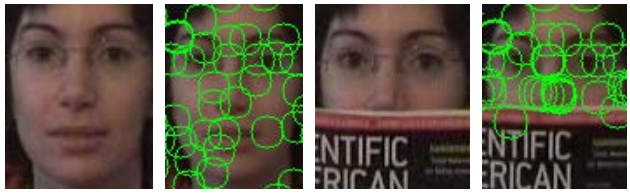
pens [9, 29]: When one updates the model with the object appearance including occlusion, we learn the noise, *i.e*. the occluding region. However, to determine such a threshold is not easy and depends on each specific sequence, which is not applicable in practice. Moreover, trying to avoid partial occlusion limits the tracker from following the target after occlusion when its appearance gradually changes during that time. This issue can be solved if we are able to detect the occlusion, replace it with the learned information, and continue updating our model. It helps the tracker to adapt to the partial appearance changes without learning the noise. Here, we assume that there is no abrupt appearance change in the occluded part during occlusion.

To address this issue, Adam *et al*. [1] proposed fragments-based tracker (Frag-Track) using integral histogram. The method simply splits the image patch into rectangular sub-regions and keeps tracking them by using spatial information and patch similarity measurements. Because only static appearance template and color information (or gray-level information) are used, it is hard to tackle challenging sequences with object appearance changes, or cluttered background and distracters. Also, the method only uses a simple rectangle representation with no scaling nor rotation, which is not practical nor descriptive enough in visual tracking. Pan and Hu [19] proposed a tracking algorithm which analyzes the occlusion by exploiting the spatiotemporal context information. The final decision is further double checked by the reference targets and motion constraints. Even though the performance of the tracker is promising, it depends on several thresholds which are not easy to set. Moreover, adaptive template matching, mainly used by this approach, may not be sophisticated enough for handling challenging situations with cluttered background. Recently, Kalal *et al*. [11] proposed the P-N Tracker using positive and negative constraints to exploit the structure of the data and get feedback about the performance of the classifier; however, it cannot deal with partial occlusion explicitly. Not directly detecting the occlusion, MILTrack [3], proposed by Babenko *et al*., learns multiple instances in an online manner to avoid drifting problem. This method is inspired from Multiple Instance Boosting proposed by Viola *et al*. [25], which considers a bag of samples labeled as positive if there is at least one positive sample, otherwise labeled as negative. Like Frag-Track [1] and PNTracker [11], it only uses simple rectangular shape without rotation. Meanwhile, Woodley *et al*. [27] proposed a tracking method using online feature selection and a local generative model with occlusion handling. However, it cannot handle illumination changes because it uses a local generative model to detect occlusion. Moreover, the method does not considered object rotation and is not extensively tested in different environment and situations such as different types of object, indoors and outdoors.



(a) Original (b) Discriminative (c) Generative

Figure 2. Partial occlusion observation on our trackers



(a) 1^{st} image (b) KLT (c) Occlusion (d) Movement

Figure 3. KLT movement when occlusion occurs.

Inspired from the HOG-LBP detector with partial occlusion handling proposed by Wang *et al*. [26], which has produced very impressive results on pedestrian detector, and the co-trained generative and discriminative trackers [29] (Co-Tracker) having presented very robust results we propose a co-training framework of generative and discriminative trackers with partial occlusion handling. We make an assumption that the appearance changes smoothly while partial occlusions create abrupt changes. Our method will show no improvement over others if this assumption is violated.

The contribution of our paper is three-fold: 1) We propose an occlusion detection method using both generative and discriminative models; 2) The movement of local feature voting process is implemented to detect if the occlusion appears; 3) An occlusion recovery and an online updating step are proposed to update both generative and discriminative models based on the non-occluded part. It is important to emphasize that the algorithm can deal with different types of objects with very limited labeled data, *i.e*. the object being selected in the first frame only.

The rest of this paper is organized as follows. The overview of our approach is presented in Section 2. The details of the generative and discriminative trackers are described in Section 3 and Section 4. The movement of local feature voting process is then presented in Section 5. The experiments are shown in Section 6, followed by conclusions and future work.

2. Overview of our approach

2.1. Motivation

After studying the classification scores of the linear SVM on the INRIA dataset [6, 7], Wang *et al*. [26] noted that the densely extracted blocks of HOG feature in the oc-

cluded area uniformly respond to the linear SVM classifier with negative inner products. Even though there is difference between detection and tracking, we observed the same effect on tracking sequences with different types of objects (Fig 2(b)). We also investigated the response of a generative model under partial occlusion and observed that the residual error is much higher in the occluded area (Fig 2(c)). Moreover, the strong edge between the object and the occluding area makes the majority of local features (here we use KLT [22]) in the region to be later occluded move in the same direction and displacement (Fig 3).

These observations allow us to design a framework to detect occlusion.

2.2. Overview

The overview of our approach is illustrated in Fig 4. A particle filter framework [10] is used for sampling to estimate the hidden state of the object given a sequence of observations.

Denote $s_t = [x, y, \rho_x, \rho_y, \theta]$ as the state of the object where (x, y) is the center of the tracking box, (ρ_x, ρ_y) is the scale w.r.t the predefined size of the object, and θ is the in-plane rotation angle. To avoid drifting, the tracker needs to find the object with an accurate center position at the right scale, rotation. At frame I_t, the result given by the tracker is a cropped image determined by the state of the tracked object. Let $O_t = (o_1, o_2, ...o_t)$ be the sequence of observed image regions over time t, our goal is to find the hidden state s_t. Assuming a Markovian state transition, a recursive equation is applied to formulate the posterior:

$$p(s_t|O_t) \propto p(o_t|s_t) \int p(s_t|s_{t-1}) p(s_{t-1}|O_{t-1}) ds_{t-1} \quad (1)$$

where $p(s_{t-1}|O_{t-1})$ is the posterior distribution from all the previous observations while $p(o_t|s_t)$ and $p(s_t|s_{t-1})$ are the observation and transition model, respectively. The critical issue is to estimate the likelihood of the new observation given the posterior distribution. In our approach, the likelihood comes from two independent models. One is the generative model, a linear subspace, which is learned online to encode the variations in appearance. The other is the discriminative model which is also trained in online manner using HOG feature set [6]. The co-training framework helps these two models train each other from the beginning when limited initialization is provided. Each model estimates the occlusion likelihood of each block in a sample; and they make the decision together. KLT features [22] are also generated and tracked in order to determine when occlusions happen by detecting uncertain region through a movement voting process. Because of the independence of these observers, the final likelihood result is the dot product of these likelihood functions.

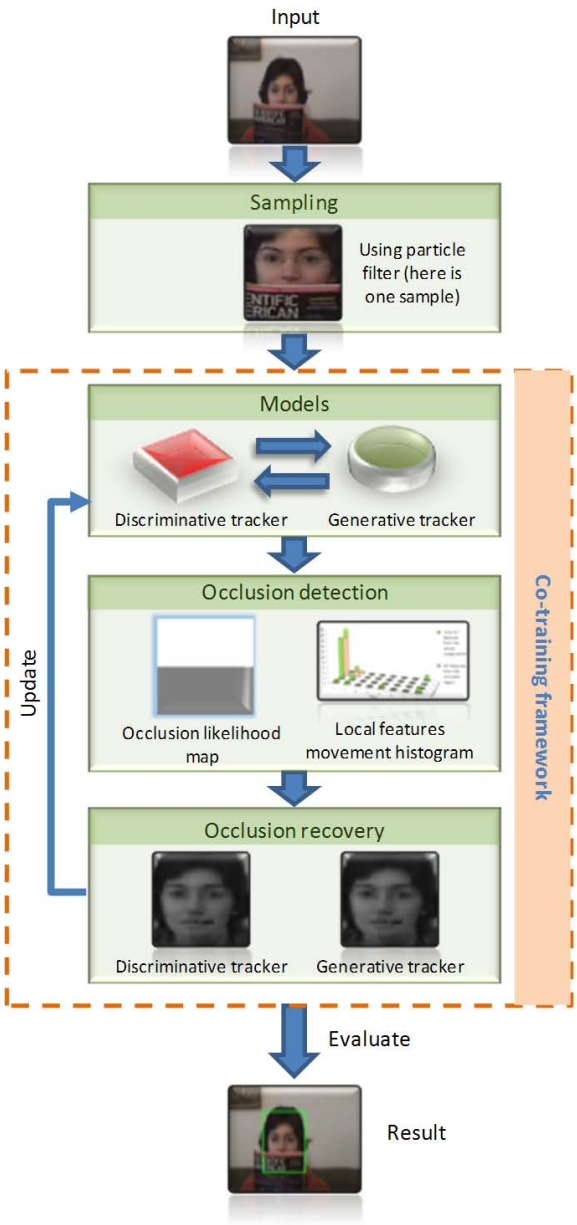


Figure 4. Overview of our approach

3. Discriminative tracker using online SVM

We adopt LASVM [5], an incremental online SVM, to train a classifier to separate object from background. Here we only discuss how to get the classifier score on each block. For more details about the training scheme, see [5].

3.1. Conventional learning

The decision function of SVM [24] is

$$f(x) = \beta + \sum_{k=1}^{n_{sv}} \alpha_k K(x, x_k) \quad (2)$$

where x is a sample and $x_k : k \in \{1, 2, ..., n_{sv}\}$ are the support vectors. $K(x, x_k)$ is the kernel function; and β is the bias constant. Here, a linear kernel is used, which means $K(x, x_k)$ is the inner scalar product of two vectors in $\Re^n$. To use the observation in Section 2, instead of computing the classification score for the whole sample patch, we compute that of each block to infer whether partial occlusion occurs, and where it is. It is important to note that we follow the way of splitting block when computing HOG features.

Following the algorithm of Wang *et al*. [26], we review the derivation to obtain the distribution of the bias constant over each block, then formulate it in online manner to fit into our training framework. Due to the linear characteristics, when using linear kernel, Eq. 2 is rewritten as:

$$f(x) = \beta + X^T.\sum_{k=1}^{n_{sv}} \alpha_k x_k = \beta + W^T.X \qquad (3)$$

where $W = \sum_{k=1}^{n_{sv}} \alpha_k x_k = \begin{pmatrix} \tilde{w}_1 \\ \vdots \\ \tilde{w}_{n_{blk}} \end{pmatrix}$ is the weighted sum of support vectors. Now we have to distribute the bias constant β to each block B_i; so that the contribution score of each block in the final classifier confidence score can be computed after subtracting that local bias β_i from the total feature inner production over that block.

For consistency, we use the same notation as in [26]. Let us denote x_p^+ as the set of HOG features of positive samples and x_q^- as the set of HOG features of negative samples, where $p = 1, ..., N^+$ (N^+ is the number of positive samples) and $q = 1, ..., N^-$(N^- is the number of negative ones). $B_{p;i}^+$ and $B_{q;i}^-$ are denoted as the ith blocks of x_p^+ and x_q^-, respectively.

Let $A = -\frac{S^-}{S^+}$ where S^- and S^+ are the summation classification scores of the positive and negative samples.

$$S^+ = \sum_{p=1}^{N^+} f(x_p^+) = N^+\beta + \sum_{p=1}^{N^+}\sum_{i=1}^{n_{blk}} \tilde{w}_i^T.B_{p;i}^+ \qquad (4)$$

$$S^- = \sum_{q=1}^{N^-} f(x_q^-) = N^-\beta + \sum_{q=1}^{N^-}\sum_{i=1}^{n_{blk}} \tilde{w}_i^T.B_{q;i}^- \qquad (5)$$

From Eq. 4 and Eq. 10, with the factor A, we have:

$$AN^+\beta + N^-\beta + \sum_{i=1}^{n_{blk}} \tilde{w}_i^T.\left(A\sum_{p=1}^{N^+} B_{p;i}^+ + \sum_{q=1}^{N^-} B_{q;i}^-\right) \qquad (6)$$

which can be written as:

$$\beta = B\sum_{i=1}^{n_{blk}} \tilde{w}_i^T.\left(A\sum_{p=1}^{N^+} B_{p;i}^+ + \sum_{q=1}^{N^-} B_{q;i}^-\right) \qquad (7)$$

where $B = -\frac{1}{A.N^+ + N^-}$. Now, we can have the distribution of bias constant on each block:

$$\beta_i = B.\tilde{w}_i^T.\left(A\sum_{p=1}^{N^+} B_{p;i}^+ + \sum_{q=1}^{N^-} B_{q;i}^-\right) \qquad (8)$$

3.2. Online learning

Up to this point, β_i is only calculated in off-line manner when all the training samples, $i.e.$ positive and negative ones, are known. Now we consider all of the notations above is for the current trained model. Assuming we have N_{new}^+ new positive samples and N_{new}^- new negative ones. Now we have $N^{'+} = N^+ + N_{new}^+$ and $N^{'-} = N^- + N_{new}^-$ are the new number of positive and negative samples in total, respectively. Because of the independence between blocks, $S^{'+}$ and $S^{'-}$ are computed as follows:

$$S'^+ = \sum_{p=1}^{N'^+} f(x_p'^+) = N'^+\beta' + \sum_{p=1}^{N'^+}\sum_{i=1}^{n_{blk}} \tilde{w'}_i^T.B_{p;i}'^+ \qquad (9)$$

$$S'^- = \sum_{q=1}^{N'^-} f(x_q'^-) = N'^-\beta' + \sum_{q=1}^{N'^-}\sum_{i=1}^{n_{blk}} \tilde{w'}_i^T.B_{q;i}'^- \qquad (10)$$

where β' and w_i' are output of LASVM after online training new samples; $B_{p;i}'^+ = B_{p;i}^+ + B_{new(p;i)}^+$ and $B_{q;i}'^- = B_{q;i}^- + B_{new(q;i)}^-$ are the updated of ith blocks in positive and negative samples, i.e. $x_p'^+$ and $x_q'^-$, respectively. Following the same computation, we have $A' = -\frac{S'^-}{S'^+}$ and $B' = -\frac{1}{A'.N'^+ + N'^-}$

The new bias constant β_i' for each block is updated using Eq. 8 with all updated parameters.

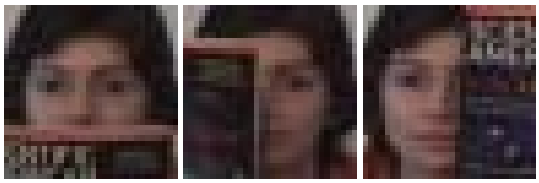
The occlusion likelihood map is generated as a binary image based on the score of each block , which is 0 if the score is negative and 1, otherwise. Each pixel in the likelihood image corresponds to a block in the sample.

3.3. Updating the model

As discussed in Section 2, if the classifier is updated including the occluding region, it may drift because the noise (occluding area) becomes part of the model. To avoid this issue, the non-occluded part is kept while the occluded area is inferred from a previous frame (as shown in Figure 5(c)). In a long-term partial occlusion, we can consider this step as a recursive process where the occluded area of the object in the current frame is projected from that of the object in the previous frame, which may also be drawn from its previous one.

4. Generative tracker using linear subspace

Although the use of multiple linear subspaces [29] produces good results in tracking, the ambiguity is high when



(a) Some partial occlusion cases at frame 28, 88, and 185

(b) Generative tracker model update

(c) Discriminative tracker model update

Figure 5. Occlusion recovery from our trackers (the image patch is scaled to 32x32 for training)

deciding whether to create a new subspace and merge a pair of existing ones or not. It is even more ambiguous when partial occlusion appears. Some new subspaces may be created, which do not reflect the correct appearance model of the object. Also, more noise is included in the model by encoding the appearance that way.

Here we propose to use a single linear subspace to approximate the appearance model of the object. This is similar to the incremental visual tracker (IVT) by Ross *et al*. [21], but with partial occlusion handling.

4.1. Online learning

In the initialization step, after collecting several samples by simple template matching, we train the model of the object from those *n* training images $\mathcal{I}_{ini} = \{I_1, ..., I_n\}$ by computing the eigenvectors U of the covariance matrix $\frac{1}{n-1}\sum_{i=1}^{n}(I_i - \bar{I})(I_i - \bar{I})^T$, where $=\frac{1}{n}\sum_{i=1}^{n} I_i$ is the mean of the training images. It can be solved by singular value decomposition (SVD) $P = U\Sigma V^T$ of the centered data matrix $[(I_1 - \bar{I})...(I_n - \bar{I})]$

Given new m images $\mathcal{I}_{add} = \{I_{n+1}...I_{n+m}\}$, the subspace needs to be incrementally updated by calculating $[P\ Q]$ where Q is the new observation matrix according to $\mathcal{I}_{add}$. As the result of the derivation in [21], we have

$$[P\ \ Q] = \left([U\ \tilde{Q}]\tilde{U}\right)\tilde{\Sigma}\left(\tilde{V}^T\begin{bmatrix} V^T & 0 \\ 0 & 1 \end{bmatrix}\right) \qquad (11)$$

In which $\tilde{Q}$ is the component of Q orthogonal to U. Finally, we have $U' = [U\ \tilde{Q}]\tilde{U}$ and $\Sigma' = \tilde{\Sigma}$ as the updated eigensystem. In our implementation, for efficiency, the top k eigenvectors ($k = 10$) are maintained to represent the model of the learned object.

4.2. Evaluation

Given a subspace Ω with the first k eigenvectors, the projection of a sample x on Ω is $y = (y_1, ..., y_m)^T = U^T(\mathbf{x} - \hat{\mathbf{x}})$. Then the likelihood of x can be expressed:

$$p(\mathbf{x}|\Omega) = \left[\frac{\exp\left(-\frac{1}{2}\sum_{i=1}^{k}\frac{y_i^2}{\lambda_i}\right)}{(2\pi)^{k/2}\prod_{i=1}^{k}\lambda_i^{1/2}}\right] \cdot \left[\frac{\exp\left(-\frac{\varepsilon^2(\mathbf{x})}{2\rho}\right)}{(2\pi\rho)^{(d-k)/2}}\right] \qquad (12)$$

Where λ_i is the eigenvalue with respect to y_i, d is the dimension of the input, $\varepsilon(x) = |x - UU^T x|$ is the projection error. The parameter $\rho = \frac{1}{d-k}\sum_{i=k+1}^{d}\lambda_i$, can be approximated as $\rho = \frac{1}{2}\lambda_{k+1}$.

However, to detect partial occlusion, as discussed in Section 2 and shown in Fig. 2(a), intuitively, the projection error is split into blocks, the same way done in the discriminative tracker. We simply compute the occlusion likelihood by using the projection error over each block.

These likelihood values are then normalized to generate the occlusion likelihood map which is a binary image. The 0 value corresponds to the block having score lower 50% of the maximum score block, and 1, otherwise.

4.3. Updating the model

To avoid modeling the occluding part when partial occlusion occurs, instead of updating the whole image patch as described in Section 4.1, we propose an algorithm to recover the occluded part. Using the generative model, we project the information encoded in the learned subspace onto the occluded area to fill up the image patch (Figure 5) and follow the online learning in Section 4.1.

5. Local features movement voting using KLT

Taking advantage of the simplicity and fast computation of KLT features [22], tracking consistency is checked based on the movement of these features in the object region at every frame. Due to the discontinuity between non-occluded and occluding regions, some KLT features are driven in the same direction and velocity which are different from the remaining part. Taking account this observation, we propose a voting scheme on the movement of these local features to detect the occlusion.

After being detected in the first frame, these features are tracked in every frame. After removing all of the outliers, the magnitude displacement of each feature is then normalized to [0,1] and encoded in a 4-bin histogram. The direction of the movement is encoded in a 8-bin histogram, each

(a) Frame 372 (b) Frame 373 (c) Generative (d) Discriminative

Figure 6. Disagreement in occlusion detection from the two trackers.

of which covers a $\frac{\pi}{4}$ span. All displacement vectors, thus, accumulate into a 4x8 2D histogram.

Let R be the candidate occluded region voted by discriminative and generative trackers, $H = \{h_{i,j}\}$ is the histogram of all the KLT features in the current frame, $H' = \{h'_{i,j}\}$ is the histogram of the KLT features which was originally in the current occluded region. Let $\hat{H} = H - H'$ be the histogram of the non-occluded part. We have:

$$h'_{max} = argmax(h'_{i,j}) \tag{13}$$

where $h'_{i,j} \in H'$. Let us call $h'_{max} = h'_{i_{max},j_{max}}$, the condition for R to be considered as occluded part is

$$y = \begin{cases} 1 & \text{if} \quad \frac{h'_{i_{max},j_{max}}}{h_{i_{max},j_{max}}} \geq \theta \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

This equation can be understood as a checking process of local features uncertainty in the occluded region. When there is a majority of KLT features in a region having different movement behavior than the rest, partial occlusion is detected. In practice, we choose $\theta = 0.7$. It is important to note that the KLT features are re-initialized after occlusion and this step is only applied as a referee when there is disagreement on occlusion detection between the two generative and discriminative trackers. An example of occlusion detection disagreement between these two trackers is shown in Fig. 6. This disagreement is resolved with the use of KLT feature occlusion detection serving as a referee.

6. Experiments

6.1. Implementation Details

To implement the generative and discriminative models, depending on the size ratio of the object, we use an image vector of size 32x32 for square-shape and 32x64 for rectangle-shape. For the generative tracker, the subspace is maintained by the top k=10 eigenvectors because it gives us the best trade-off in precision and running time. Every 5 frames, we update the subspace once. For the discriminative tracker, we use the linear kernel LASVM [5] with R-HOG feature set [6] (16x16 block size and 8x8 cell size). To allow the overlapping HOG descriptor, we use the step size of 8. For a block we have 36-bin oriented histogram. Because of the growth in number of support vectors, a sliding window of 300 frames is applied to focus on the current appearance of the object. In the first frame, we manually select the object and apply simple template matching for the next 4 frames. These initial labeled data are then transferred to both generative and discriminative trackers for training. Our Bayesian framework generates 600 particles at each frame. The combined tracker is implemented in C++ and runs at 4fps on an Intel QuadCore 3.0GHz system. At every frame, each of the trackers independently predicts the unlabeled data based on its trained model. Following [29], in the discriminative tracker, we also convert score of SVM to probability output [20] and follow the same threshold settings for both trackers.

In the co-decision step to combine the two occlusion likelihood maps, we simply use an AND operator to integrate them into the final one. However, when there are more than 70% of the pixels different between these two likelihood maps, we use local features movement voting process to choose the detector to rely on. In our experiments, it happens mostly when there is local change causing false occlusion detection by generative model.

6.2. Comparison

We tested our algorithm on several challenging published video sequences of different types of objects in indoor and outdoor environments. Several related state-of-the-art trackers included in the comparison are the Co-Tracker [29], which is the most related to our tracker, the Frag-Tracker [1], the Online and Semi-Boosting Tracker (OAB, SB) [8, 9], the P-N Tracker (PNT) [11], the MIL-Tracker [3] and its new variation with no regret MIO Tracker [14]. We use the provided results and published source code from the authors[1,2,3] . These methods were also demonstrated on published benchmark video sequences for comparison. We also demonstrated the robustness of our proposed partial occlusion handling co-training framework by comparing our tracker with each component without occlusion handling. For the MIL, MIO, OAB, SB trackers, we use the settings described in the papers with some optimized parameters in search range and number of selectors. The parameters (except the search range) in Frag-Tracker are kept as default. In the Co-Tracker and the two components of our tracker, the parameters are set the same as our tracker. To prove the precision of our tracker, we used the same measurement, average center location errors (in pixels), used for evaluation in [3, 14].

The testing sequences include five videos reported by MILTracker and one video published in [30]. The ground truth centers of every five frames are also provided by Babenko *et al.*[1]. We also labeled the ground truth for the new sequence in the same manner. The resolution of all video frames is 320x240, except the Occluded Face which

[1] MILTracker: http://vision.ucsd.edu/ babenko/project_miltrack.shtml

[2] Frag-Tracker: http://www.cs.technion.ac.il/ amita/fragtrack/fragtrack.htm

[3] Semi-boosting Tracker: http://www.vision.ee.ethz.ch/boostingTrackers/index.htm

Video Sequence	Frames	GT	DT	FT	OAB	ST	PNT	MILT	MIO	CoT	**Ours**
Coke Can	292	102	9	67	25	85	**8**	21	22	10	**8**
Occluded Face 1	900	86	17	*7*	44	41	8	27	14	16	**5**
Occluded Face 2	808	14	12	21	21	43	*8*	20	13	12	**7**
Person	200	35	73	44	37	154	44	34	n/a	*33*	**5**
Tiger 1	354	52	6	40	35	46	13	15	24	*5*	**4**
Tiger 2	365	43	*7*	37	34	53	21	17	23	*7*	**5**

Table 1. Average center location errors. (GT: Generative Tracker, DT: Discriminative Tracker, FT: Frag-Tracker [1], OAB: Online Boosting Tracker [8], ST: Semi-Boosting Tracker [9], PNT: P-N Tracker [11], MILT: MILTracker [3], MIO: MIL No Regret Tracker [14], CoT: Co-Tracker [29]) in different challenging datasets. The best performance is in bold, the second best is in italic.

(a) Occluded Face 1

(b) Occluded Face 2

(c) Person

(d) Tiger 2

Figure 7. Some screen shots from the testing results. Because of clarity issue, we only choose Frag-Tracker [1] and MILTracker [3] to show some results comparing with our tracker.

is 352x288. The quantitative comparison results, which are presented in Table 1, clearly show the advantages of our approach. "N/a" is reported when we do not have the results from that method. All of the other trackers cannot adapt well to the object appearance changes including lighting change, pose change (occluded face 2) and fail when total occlusion appears in seq. "person". Also, our algorithm helps to avoid drift shown by the lowest position error compared to others in all sequences. All of the sequences provide long-term and heavy partial and total occlusions, and challenging appearance changes such as illumination changes, abrupt motion, rotation, and cluttered backgrounds.

Occluded Face and Occluded Face 2: although the "Occluded Face" sequence contains many occlusion cases, the object does not move much and is quite distinctive from the background. However, it is a good example for us to test our occlusion detection. Our tracker outperforms others, especially Frag-Tracker, which is proposed to solve the partial occlusion problem using a part-based model. The "Occluded Face 2" is much more complicated, it contains illumination change, in-plane rotation, and heavy occlusion. Other trackers hardly get the precise position for the center when occlusion and rotation happen while our tracker tracks the face consistently with very good rotation.

Coke Can, Tiger, and Tiger2: These three sequences share the same challenges: illumination changes, abrupt motion, partial occlusion, and rotation changes. The generative model is not very effective because of abrupt changes in appearance, whereas the discriminative one obtains excellent performance.

Person: This sequence was taken outdoor and contains total occlusion [30]. While Frag-Tracker is stuck at a similar area in the background when the person rotates and MILTracker cannot handle the occlusion when another person passes through our object, our tracker can re-initialize to the target immediately after the total occlusion based on the occlusion recovery information.

Please refer to our supplemental video for the details.

7. Conclusions and future work

We have proposed a novel co-training framework of generative and discriminative trackers with partial occlusion handling. Our algorithm can encode the global appearance

model of the object in a compact linear subspace while strengthening the discriminative power to separate the object and background. The co-decision process for occlusion handling, with the help of the local features movement voting process, robustly detects the occluded region and helps the trackers ignore that region and update the new model consistently. Moreover, the co-training framework helps the two trackers update each other on-the-fly, which is especially helpful when each of them fails during tracking.

Currently, our tracker cannot handle the case when there is an abrupt change during the occlusion because there is no learned knowledge to predict the changes in the hidden region according to the revealed one. In the future, we expect to build a learning algorithm to cope with this issue. We also expect to develop specific trackers for different types of objects such as face, people, and vehicle whose model can be learned offline.

8. Acknowledgements

This research was funded, in part, by MURI-ARO W911NF-06-1-0094. The first author was also supported by the Vietnam Education Foundation. We thank Zdenek Kalal for his help with the P-N Tracker [11].

References

[1] A. Adam, E. Rivlin, and I. Shimshoni. Robust fragments-based tracking using the integral histogram. In *CVPR*, pages 798–805, 2006.

[2] S. Avidan. Ensemble tracking. In *PAMI*, volume 29, pages 261–271, 2007.

[3] B. Babenko, M.-H. Yang, and S. Belongie. Visual tracking with online multiple instance learning. In *CVPR*, pages 983–990, 2009.

[4] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *COLT*, pages 92–100, 1998.

[5] A. Bordes, S. Ertekin, J. Weston, and L. Bottou. Fast kernel classifiers with online and active learning. In *JMLR*, pages 1579–1619, 2005.

[6] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, pages 886–893, 2005.

[7] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. In *ECCV*, pages 428–441, 2006.

[8] H. Grabner, M. Grabner, and H. Bischof. Real-time tracking via online boosting. In *BMVC*, pages 47–56, 2006.

[9] H. Grabner, C. Leistner, and H. Bischof. Semi-supervised on-line boosting for robust tracking. In *ECCV*, pages 234–247, 2008.

[10] M. Isard and A. Blake. Condensation - conditional density propagation for visual tracking. In *IJCV*, volume 29, pages 5–28, 1998.

[11] Z. Kalal, J. Matas, and K. Mikolajczyk. P-n learning: Bootstrapping binary classifiers by structural constraints. In *CVPR*, 2010.

[12] J. Lasserre, C. Bishop, and T. Minka. Principled hybrids of generative and discriminative models. In *CVPR*, pages 87–94, 2006.

[13] K.-C. Lee and D. Kriegman. Online learning of probabilistic appearance manifolds for video-based recognition and tracking. In *CVPR*, pages 852–859, 2005.

[14] M. Li, J. T. Kwok, and B.-L. Lu. Online multiple instance learning with no regret. In *CVPR*, 2010.

[15] R. S. Lin, D. Ross, J. Lim, and M. H. Yang. Adaptive discriminative generative model and its applications. In *NIPS*, pages 801–808, 2004.

[16] R. Liu, J. Cheng, and H. Q. Lu. A robust boosting tracker with minimum error bound in a co-training. In *ICCV*, pages 1459–1466, 2009.

[17] S. M. S. Nejhum, J. Ho, and M.-H. Yang. Visual tracking with histograms and articulating blocks. In *CVPR*, pages 1–8, 2008.

[18] A. Y. Ng and M. I. Jordan. On discriminative vs. generative classifiers: a comparison of logistic regression and naive Bayes. In *NIPS*, pages 438–451, 2001.

[19] J. Pan and B. Hu. Robust occlusion handling in object tracking. In *CVPR*, 2007.

[20] J. C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*, 1999.

[21] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang. Incremental learning for robust visual tracking. In *IJCV*, volume 77, pages 125–141, 2008.

[22] J. Shi and C. Tomasi. Good features to track. In *CVPR*, pages 593–600, 1994.

[23] F. Tang, S. Brennan, Q. Zhao, and H. Tao. Co-tracking using semi-supervised support vector machines. In *ICCV*, pages 1–8, 2007.

[24] V. N. Vapnik. Statistical learning theory. In *John Wiley Sons*, 1998.

[25] P. Viola, J. C. Platt, and C. Zhang. Multiple instance boosting for object detection. In *NIPS*, pages 1417–1426, 2005.

[26] X. Wang, T. X. Han, and Z. He. An HOG-LBP human detector with partial occlusion handling. In *ICCV*, 2009.

[27] T. Woodley, B. Stenger, and R. Cipolla. Tracking using online feature selection and a local generative model. In *BMVC*, volume 2, pages 790–799, 2007.

[28] M. Yang and Y. Wu. Tracking non-stationary appearances and dynamic feature selection. In *CVPR*, pages 1059–1066, 2005.

[29] Q. Yu, T. Dinh, and G. Medioni. Online tracking and reacquisition using co-trained generative and discriminative trackers. In *ECCV*, pages 678–691, 2008.

[30] Q. Yu and G. Medioni. Multiple-target tracking by spatiotemporal Monte Carlo Markov chain data association. In *PAMI*, volume 31, pages 2196–2210, 2009.