ONLINE STRUCTURED HOUGH FORESTS FOR VISUAL TRACKING

Tao Qin [1] *Bineng Zhong* [1, 2] *Hanzi Wang* [1, *]

[1] School of Information Science and Technology, Xiamen University, China
[2] Department of Computer science and Technology, Huaqiao University, China
Email: {qintaohao, bnzhong}@gmail.com, hanzi.wang@xmu.edu.cn

ABSTRACT

Segmentation-based tracking methods are popular in alleviating the model drift problem during online-learning of visual trackers. However, one of the limitations of those methods is that tracking results guide the process of segmentation. The model drift problem in tracking may have significant influence on segmentation. In this paper, we propose an online structured Hough Forests to address this limitation. The results of object tracking do not have significant influence on the process of segmentation. Our algorithm shows more robust results on several challenging sequences.

***Index Terms*—** Online Structured Hough Forests, Visual Tracking, Online Learning, Segmentation

1. INTRODUCTION

In recent years, segmentation-based tracking methods have increasingly been used to alleviate the online drift problem [1–4]. The idea behind those methods is that accurate segmentation can alleviate the model drift problem by providing an object contour constraint. Typically, the segmentation-based tracking framework is performed by three steps: (1) locating a target by a tracker, (2) utilizing tracking results to segment the target, and (3) utilizing the foreground-background segmentation results to update the tracker's model. Consequently, model drift may seriously affect the results of segmentation.

Because the single atomic class labels to samples do not exhibit an inherently structural information, exploiting the structural information of the labeled images has drawn much attention in computer vision community. Structured learning [5–7] is introduced to the computer vision area for object detection [8] and tracking [9].

Our paper is related to and inspired by the recent work in [7]. The authors of [7] provide a simple but effective way to integrate structural information in the popular random forest learning algorithms [10, 11]. However, their work is proposed to deal with semantic image labeling by offline learning. In our setting, there is no enough offline labeled samples for training forests except for the first frame which is annotated manually. In this study, we propose an online Structured Hough Forests (i.e., SHF) approach which can combine the structured class-labels used in [7] and the online Hough Forests in [12] into a single structured learning framework. The main advantage of our algorithm is that the tracking and segmentation is simultaneously accomplished by using the online structured Hough Forests. Because the process of segmentation do not directly utilize the results of tracking, model drift during tracking has less influence on the results of segmentation. In addition, we propose to use level optimization to extend offline structured statistics in trees' leaf to online mode.

2. OUR METHOD

Our algorithm (see Fig. 1) integrates structured information into online Hough Forests, which is called online structured Hough Forests. It can be seen from Fig.1 that the object centers are located by Hough voting, and the binary images obtained via structured class-labels are used to guide the updating processes of the object appearance models.

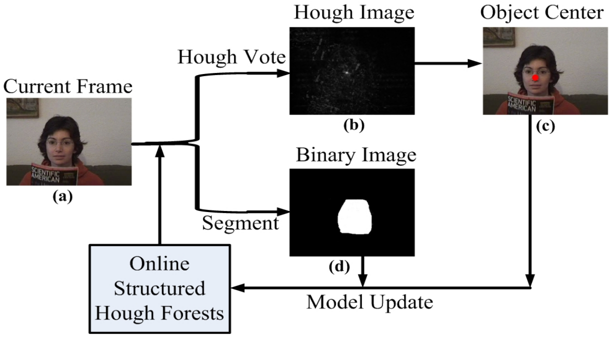

Fig. 1. The flowchart of the online Structured Hough Forest

2.1. Online Structured Hough Forests

The structured Hough Forests algorithm consists of a set of random trees [10] that are trained to learn a mapping from local image patches (the patches' size is: $d \times d$) to their corresponding Hough votes in a Hough space $\mathcal{H} \subseteq \mathbb{R}^H$ and structured labels in a structured label space $\mathcal{L}$. The online SHF approach is summarized in Algorithm 1.

[*] the corresponding author.

978-1-4799-0356-6/13/$31.00 ©2013 IEEE

Algorithm 1 Online Structured Hough Forests

```
Input: The figure-ground mask $m_0$ and an object center $c_0$ in the first frame; the video sequence $F_{seq} = \{f_0, f_1, ..., f_{N_{\max}}\}$
Output: The figure-ground masks $M_{rem} = \{m_1, ..., m_{\max}\}$ and the object centers $C_{rem} = \{c_1, ..., c_{\max}\}$ in the following frames in the video sequence.
Initialization:
1) Warp $m_0$ and the corresponding frame $f_0 \in F_{seq}$ to get three sets: $M_{init} = \{m_0, m_1^w, ..., m_{T-1}^w\}$, $F_{init} = \{f_0, f_1^w, ..., f_{T-1}^w\}$ and $C_{init} = \{c_0, c_1^w ..., c_{T-1}^w\}$.
2) Generate randomly sampled image patches from $F_{init}$, simultaneously record the corresponding structured labels and voting vectors according to $M_{init}$ and $C_{init}$.
3) Extract HOG-like features for the samples.
4) Train structured Hough Forests by using the labeled samples.
for $\mathrm{n} = 1 : N_{\max}$ do
    ▷ Obtain Hough Image $V_n^{map}$ and structured label prediction by using SHF at the current frame $f_n$.
    ▷ Estimate the object center $c_n$ from $V_n^{map}$.
    ▷ Fuse the structured predictions to get the figure-ground masks $m_n$ at the current frame $f_n$.
    ▷ Extract samples $\{\mathbf{S}_{u_1,v_1,f_n}, ..., \mathbf{S}_{u_M,v_M,f_n}\}$ randomly from $f_n$, $c_n$ and $m_n$ to update the SHF.
```

2.1.1. Hough Voting and Structured Labels

Hough votes [13] are the vectors which point toward the expected object center. An object position estimate h can be computed by accumulating Hough votes from local image patches. The Hough image can be generated by accumulating the weighted voting vectors from the image patches in the current test image. For a detailed description, we refer readers to the work [2, section 2.2].

In previous approaches, an image patch is assigned with one single atomic class label. The structured label l is a matrix whose size is $d' \times d'$. The elements of the matrix are the figure-ground labels, i.e., $l_{ij} \in \mathcal{Y}$ ($\mathcal{Y} = \{0, 1\}$), which denotes the ij-entry of the label patch l. When an image patch pixel corresponding to l_{ij} is a foreground pixel, the value of l_{ij} is 1; otherwise, it is 0.

2.1.2. Training Samples

Let $\mathbf{p}_{(u,v,I)}$ denote a patch which is extracted at a position (u, v) in an image I (i.e., the patch's center is at (u, v)), a training sample at (u, v) is obtained in the form $\{\mathbf{S}_{u,v,I} = (\mathbf{p}_{(u,v,I)}, \chi_{(u,v,I)}, l_{(u,v,I)}, \mathbf{v}_{(u,v,I)})\}$, where $\mathbf{p}_{(u,v,I)}$ is the original image patch centered at (u, v) of the image I; $\chi_{(u,v,I)}$ is the feature of $\mathbf{p}_{(u,v,I)}$; and $l_{(u,v,I)}$ is the structured label of $\mathbf{p}_{(u,v,I)}$. The Hough vote $\mathbf{v}_{(u,v,I)}$ is defined for the training sample when the percentage of foreground pixels is larger than a threshold α in $\mathbf{p}_{(u,v,I)}$ ($\alpha = 1/3$ in our case).

2.1.3. Extremely Randomized Decision Forests

Before the structured Hough Forests can be applied for prediction, the tree structure and the statistics in the leaf nodes have to be established. However, the labeled samples at the first frame are not enough to optimize the binary splitting tests at a tree's non-leaf node. Thus, for simplicity and efficiency, we adopt the random initialization of the tree structures [2, section 3.1] to solve the problem.

2.1.4. Leaf Node Statistics

After training samples have been routed through the tree to the leafs, we need to model the corresponding structured labels and voting map of the leaf nodes.

Let $\mathcal{L}_\kappa$ ($\mathcal{L}_\kappa \subseteq \mathcal{L}$) be the set of the structured label patches arriving at the leaf κ. Because the class label l representing the leaf node is now a structured label with a size of $d' \times d'$, we need to use a mechanism to select a structured label from $\mathcal{L}_\kappa$ which can represent the label patches in $\mathcal{L}_\kappa$. A good selection for the structured class label should represent a mode of the joint distribution of the label patches in $\mathcal{L}_\kappa$. For simplicity, we compute the joint probability as follows:

$$Pr(l|\mathcal{L}_\kappa) = \prod_{(i,j)} Pr^{(i,j)}(l_{ij}|\mathcal{L}_\kappa) \tag{1}$$

where l_{ij} is the value at the position (i, j) of the structured label patch l.

$$Pr^{(i,j)}(l_{ij}|\mathcal{L}_\kappa) = 1 - \frac{\sum_{r=1}^{N_{\mathcal{L}_\kappa}} |l_{ij} - l_{ij}^r|}{N_{\mathcal{L}_\kappa}} \tag{2}$$

where $N_{\mathcal{L}_\kappa}$ denotes the number of the structured label patches in the set $\mathcal{L}_\kappa$. l_{ij}^r is the label at (i, j) of the $r\text{-}th$ structured label patch in the set $\mathcal{L}_\kappa$. The label patch π in $\mathcal{L}_\kappa$ is finally selected for the leaf κ as the single label patch prediction which can maximize the joint probability:

$$\pi = \underset{l \in \mathcal{L}_\kappa}{arg\,max}\, Pr(l|\mathcal{L}_\kappa) \tag{3}$$

The Hough vector set $\mathcal{V}_\kappa$ at the leaf node κ is the set of the vectors derived from the samples which reach the leaf node κ during initialization and updating steps. The voting weight can be computed as follows:

$$w_\kappa = \frac{\sum_{r=1}^{N_{\mathcal{L}_\kappa}} h\left(\frac{S_{fore}(l^r)}{S(l^r)}\right)}{N_{\mathcal{L}_\kappa}} \tag{4}$$

where $h(\cdot)$ is a step function with a threshold $\theta = 1/3$. $S_{fore}(\cdot)$ is the area of the foreground region in the sample's structured label patch, and $S(\cdot)$ is the corresponding total area. Only when the foreground area is more than one-third of the entire region, this sample will contribute a Hough vote.

2.2. Structured Prediction and Hough Localization

The structured predictions gathered from the trees of the structured Hough Forests have to be combined into a single label patch prediction. We use the mechanism, similar to (3), to obtain the label patch prediction for the test patch $\mathbf{p}_{(u',v',I)}$.

For each pixel in a test image, we can obtain $d' \times d'$ class predictions from the adjacent pixels' structured predictions, which have to be integrated into a single class prediction. A simple way is to use the voting mechanism which selects the mostly voted class per pixel. After the fusion of the structured predictions, we get a mask m_n for the test image.

During localization, we locate the object center by searching for the maxima at the voting map V_n^{map}.

2.3. Initialization and Online Update

We follow the step **Initialization** at **Algorithm 1** to initialize the SHF tracker. The figure-ground mask can not only provide an accurate contour of a target, but also provide structured labels for the samples. Therefore, we use the figure-ground mask m_0 in the first frame to train the SHF tracker.

After the completion of the structured prediction and localization, the tracked object is not only located, but also accurately segmented from the background. Then, we extract new training samples $\{\mathbf{S}_{u_1,v_1,f_n}, ..., \mathbf{S}_{u_M,v_M,f_M}\}$ randomly from the tracked object region and the background surrounding the object. The training samples are also routed through the trees of the structured Hough Forests to the leafs.

For the update of Hough votes of the leafs, we simply add Hough votes from new samples to their corresponding leaf nodes. We also recalculate the voting weight w of each leaf node.

Because the value of $N_{\mathcal{L}_\kappa}$ increases after several times of update, the computational cost for solving (3) significantly grows with time. We use exhaustive search to solve it.

Therefore, we use a mechanism of level optimization to select the single label patch prediction for each leaf node. We use $\mathcal{L}_\kappa^{0:n} = \{\mathcal{L}_\kappa^{m_0}, \mathcal{L}_\kappa^{m_1}, ..., \mathcal{L}_\kappa^{m_t}, ..., \mathcal{L}_\kappa^{m_n}\}$ to represent the samples' structured label patches reaching to the leaf κ from the beginning to the current time n. $\mathcal{L}_\kappa^{m_t}$ represents the samples reaching to the leaf κ at time t. If there is no sample reaching to the leaf κ at time t, $\mathcal{L}_\kappa^{m_t}$ is null. Firstly, if $\mathcal{L}_\kappa^{m_t}$ exists, we select the label patch π^{m_t} as the single label patch prediction to represent $\mathcal{L}_\kappa^{m_t}(t = \{0, 1, ..., n\})$ which can maximize the joint probability:

$$\pi^{m_t} = \underset{l \in \mathcal{L}_\kappa^{m_t}}{arg\, max}\, Pr(l|\mathcal{L}_\kappa^{m_t}) \tag{5}$$

Then, we get the structured label patch set $\mathcal{L}_\kappa^{*} = \{\pi^{m_t} | t \in (0, 1, ..., n)\}$. At last, we also select a label patch π as the single label patch prediction to represent $\mathcal{L}_\kappa^{*}$ by using a strategy similar to (5). We use the label patch π to represent $\mathcal{L}_\kappa^{0:n}$.

During updating at time n, we need to calculate π^{m_n} corresponding to $\mathcal{L}_\kappa^{m_n}$ ($\{\pi^{m_0}, \pi^{m_1}, ..., \pi^{m_{n-1}}\}$ is obtained at the previous initialization and updates), and then combine it with $\{\pi^{m_0}, \pi^{m_1}, ..., \pi^{m_{n-1}}\}$ to obtain the label patch π by using above-mentioned method. The mechanism of level optimization can largely decrease the computational cost.

For online tracking, not only the distribution of the samples changes over time, but also the computational cost for online update increases over time. Therefore, we need to forget old information by discarding the old samples. However, in order to alleviate the model drift problem, we should retain the samples from the initialization. We can also easily obtain $\mathcal{V}_\kappa^{0:n} = \{\mathcal{V}_\kappa^{m_0}, \mathcal{V}_\kappa^{m_1}, ..., \mathcal{V}_\kappa^{m_t}, ..., \mathcal{V}_\kappa^{m_n}\}$ to represent the samples' Hough votes reaching to the leaf κ from the beginning to the current time n. In order to keep the validity of the level optimization, we propose to discard two subsets $\mathcal{L}_\kappa^{m_\beta}$ and $\mathcal{V}_\kappa^{m_\beta}$ from the two sets $\mathcal{L}_\kappa^{0:n}$ and $\mathcal{V}_\kappa^{0:n}$, when m_n exceeds a certain threshold α and m_β is randomly obtained in the interval of $[1, m_n - 1]$ by using the random process of the uniform distribution.

3. RESULTS

In this section, the proposed SHF tracker is applied to several publicly available challenging video sequences and compared with four state-of-the-art trackers: the online boosting tracker (Boost) [14], the fragments based tracker (FRAG) [15], the online Hough forest tracker based on online tree's growth (HF1) [12], and the Hough forest tracker based on extremely random tree (HF2) [2].

We use the same settings to our tracker for all the experiments. The features used are as follows: Lab color space (3 channels), the first and second order derivatives in x and y (4 channels), and 9-bin histogram of the gradients (9 channels). The used image patch size is 12 × 12 pixels. The threshold value of α in the Section 2.3 is set to 6.

Table 1. The average errors (in pixels) of the estimated object locations.

Video	**SHF**	Boost	FRAG	HF1	HF2
Bolt	**18**	113	81	80	273
Face	**11**	11	22	17	32
Pedxing-seq3	**17**	40	48	20	18
RIGHT	**6**	44	36	35	42
Seq2	**4**	5	37	5	4
Woman	**17**	120	73	74	73

3.1. Qualitative Comparison

The sequences of Bolt, Pedxing-seq3, RIGHT, and Woman contain non-rigid deformation. The sequences of Bolt and RIGHT have background distractors around the target. The sequences of Face and Woman contain occlusions. The sequences of RIGHT and Seq2 involve the challenging illumination variation. In Figure 2, we show the qualitative comparison results.

For the Bolt sequence, only SHF successfully tracks the target throughout the sequence, while the others fail because

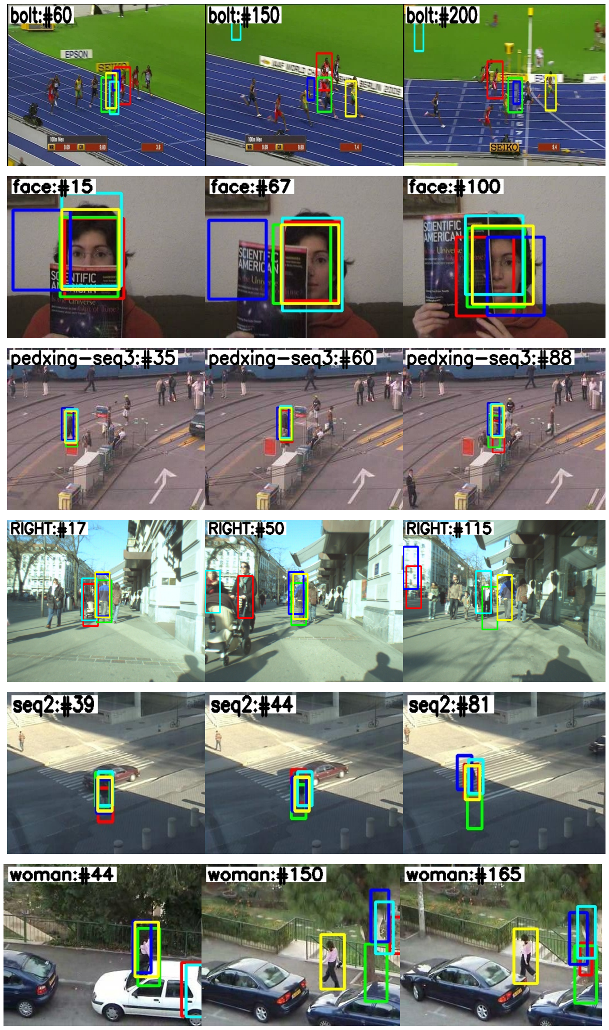

Fig. 2. The comparison results obtained by SHF (yellow), the OBT (red), FRAG (green), HF1 (blue), HF2 (cyan) on the six challenging video sequence.

of the non-rigid deformation and the background distractors around the object. For the RIGHT sequence, the background distractors, non-rigid deformation and illumination variation lead the other trackers to lose the target at the intermediate stage. In comparison, the proposed SHF tracker can robustly deal with occlusions and non-rigid deformation in the Woman sequence, and it never loses the target until the end of the sequence. For the Face, Pedxing-seq3, and Seq2 sequences, when the challenging situations occur in those sequences, the other trackers easily deviate the target center while SHF achieves more robust results.

3.2. Quantitative Comparison

The quantitative comparison results of the competing trackers are shown in Figure 3. The center location error is defined as the L1-Norm error $(i.e., |x_n^{tr} - x| + |y_n^{tr} - y|)$ between the target's ground truth location (x_n^{tr}, y_n^{tr}) and the estimated location (x_n, y_n) obtained by the trackers at the n-th frame. Figure 3 plots the center location errors obtained by the five competing methods on the six sequences. Table 1 gives the averaged errors of the estimated target locations obtained by the competing methods. As can be seen in Figure 3 and Table 1, the other trackers fail to track the target when the challenging situations occur in the video sequences. In comparison, only SHF can successfully track the target throughout all the sequences and achieved the most accurate results.

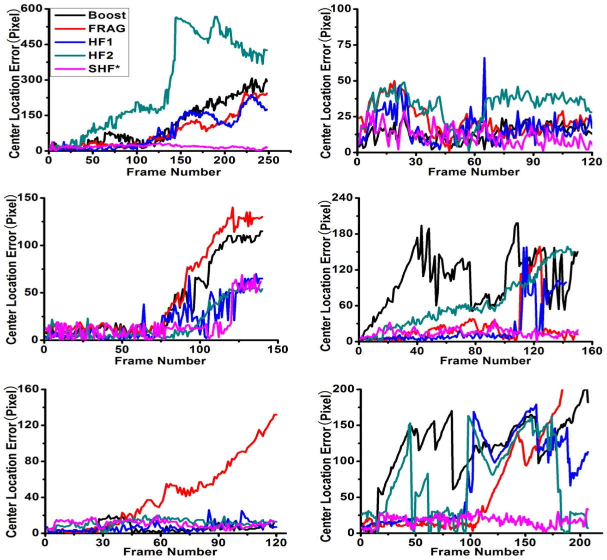

Fig. 3. The error plots of the estimated object locations obtained by the five competing trackers. 1st is for the Bolt and Face sequences; 2nd row is for the Pexding-seq3 and RIGHT sequences; 3rd row is for the Seq2 and Woman sequences.

4. CONCLUSIONS

In this paper, we develop a novel online structured Hough Forests learning method for visual tracking, in which the structured label is incorporated into online Hough Forests to simultaneously implement object localization and segmentation. This can effectively alleviate the model drift problem. In addition, we propose to use level optimization to reduce the computational cost. The results demonstrate that the proposed online structured Hough Forests method is robust to illumination changes, non-rigid deformation, occlusion, and clutter background.

5. ACKNOWLEDGEMENT

This work is supported by Natural Science Foundation of China (No. 61170179, No. 61202299 and No. 61201359), China Postdoctoral Science Foundation (No. 2011M501081), and the Xiamen Science & Technology Planning Project Fund (3502Z20116005) of China.

6. REFERENCES

[1] J. Fan, X. Shen, and Y. Wu, "Scribble tracker: A matting-based approach for robust tracking," *IEEE Trans. PAMI*, vol. 34, pp. 1633–1644, 2012.

[2] M. Godec, P. M.Roth, and H. Bischof, "Hough-based tracking of non-rigid objects," in *Proc. ICCV*, 2011, pp. 81–88.

[3] Z. Yin and R.T. Collins, "Shape constrained figure-ground segmentation and tracking," in *Proc. CVPR*, 2009, pp. 731–738.

[4] B.N. Zhong, H.X. Yao, S. Chen, R.R. Ji, X.T. Yuan, S.H. Liu, and W. Gao, "Visual tracking via weakly supervised learning from multiple imperfect oracles," in *Proc. CVPR*, 2010, pp. 1323–1330.

[5] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun, "Support vector learning for interdependent and structured ouput spaces," in *Proc. ICML*, 2004, pp. 104–112.

[6] C. Yu and T. Joachims, "Learning structured svms with latent variables," in *Proc. ICML*, 2009, pp. 1169–1176.

[7] P. Kontschieder, S.R. Bulo, H. Bischof, and M. Pelillo, "Structured class-labels in random forests for semantic image labelling," in *Proc. ICCV*, 2011, pp. 2190–2197.

[8] M.B. Blaschko and C.H. Lampert, "Learning to localize objects with structured ouput regression," in *Proc. ECCV*, 2008, pp. 2–15.

[9] S. Hare, A. saffari, and P.H.S. Torr, "Struck: Structured output tracking with kernels," in *Proc. ICCV*, 2011, pp. 803–806.

[10] L. Breiman, "Random forests," *Machine Learning*, vol. 45(1), pp. 5–32, 2001.

[11] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Machine Learning*, vol. 63, pp. 3–42, April 2006.

[12] S. Schulter, C. Leistner, P.M. Roth, L.V. Van, and H. Bischof, "On-line hough forest," in *Proc. BMVC*, 2011, pp. 1–11.

[13] J. Gall and V. Lempitsky, "Robust fragments-based tracking using the integral histogram," in *Proc. CVPR*, 2009, pp. 1022–1029.

[14] H. Grabner, M. Grabner, and H. Bischof, "Real-time tracking via on-line boosting," in *Proc. BMVC*, 2006, pp. 1–10.

[15] E. Rivlin A. Adam and I. Shimshoni, "Robust fragments-based tracking using the integral histogram," in *Proc. CVPR*, 2006, pp. 798–805.