arXiv:1806.02523v1 [cs.CV] 7 Jun 2018

INFORMATION-MAXIMIZING SAMPLING TO PROMOTE TRACKING-BY-DETECTION

Kourosh Meshgi[⋆] *Maryam Sadat Mirzaei*[†] *Shigeyuki Oba*[⋆]

[⋆] Graduate School of Informatics, Kyoto University, Japan
[†] RIKEN Center for Advanced Intelligence Project (AIP), Japan

ABSTRACT

The performance of an adaptive tracking-by-detection algorithm not only depends on the classification and updating processes but also on the sampling. Typically, such trackers select their samples from the vicinity of the last predicted object location, or from its expected location using a predefined motion model, which does not exploit the contents of the samples nor the information provided by the classifier. We introduced the idea of most informative sampling, in which the sampler attempts to select samples that trouble the classifier of a discriminative tracker. We then proposed an active discriminative co-tracker that embed an adversarial sampler to increase its robustness against various tracking challenges. Experiments show that our proposed tracker outperforms state-of-the-art trackers on various benchmark videos.

Index Terms— visual tracking, information-maximizing sampling, active learning, structured sample learning

1. INTRODUCTION

Visual Tracking is one of the most fundamental building blocks in understanding videos and motion in the real-world. Discriminative trackers formulate tracking as a foreground/background discrimination, to tackle problems of generative models such as complex non-linear dynamics of the object and background clutter [1]. Correlation filters and tracking-by-detection are mainstreams of such trackers.

Tracking-by-detection approaches [1–7], utilize one or more classifiers to classify the target. Despite their success in recent large benchmarks [8, 9], these trackers still suffer from several shortcomings: *(i) Uninformed sampling*, *(ii) Label noise*: even the smallest mistakes in labeling are gradually accumulated in the self-learning loop of tracking-by-detection and cause a drift in the tracker, and *(iii) Model drift*: an adaptive tracker should be updated rapidly yet remember the target appearance to recover from occlusions or target losses. Updating the model itself is not a straightforward task [4]. Label noise has been studied extensively, and various solutions

This article is based on results obtained from a project commissioned by the NEDO and was supported by Post-K application development for exploratory challenges from Japan's MEXT.

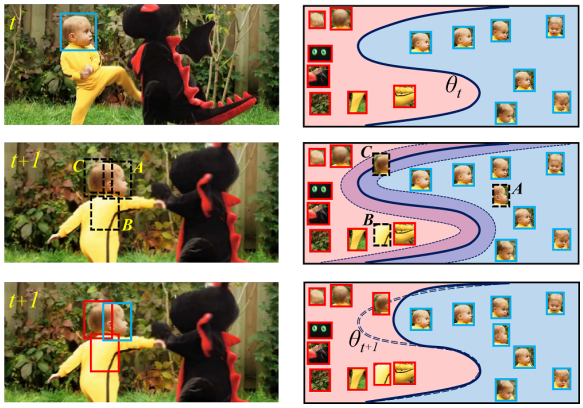

Fig. 1. In frame t the sampler selects three samples A, B, and C to evaluate by the classifier θ_t. While samples A and B are easy for the classifier to classify, a label for sample C would be uncertain. If θ_t is used to label sample C, it would be misclassified as positive sample. On the other hand, since this sample is located near the decision boundary of classifier, knowing the correct label of C is crucial to effectively update the model to θ_{t+1}. Thus among these, C is the most informative sample (principle of uncertain sampling) and an auxiliary classifier is needed to provide its label (co-learning).

such as robust loss functions [12], exploiting the information in unlabeled data [13], or even merging the sampling and learning process by structured learning is proposed [3], ensemble-based trackers [14] and co-tracking [1]. Model update, despite all the efforts, still challenges the performance of the trackers. Different online learning approaches (e.g., subspace, dictionary or incremental learning), as well as different update strategies (e.g., budgeted updating [3], using auxiliary classifiers for sanity-check of the update [2], rolling back bad updates [4], and combining long and short-term memories [5]), tried to alleviate this problem.

To define the region-of-interest for sampling, some tracking-by-detection used context information [7], optical flow [2], dynamics model [15], or a pool of motion models [16] but most of these methods suffers from sudden failures. This is mainly characterized to the assumption that the last predicted target location is accurate, an assumption that can be violated under challenging real-world scenarios such as abrupt or fast target movements, occlusions or severe clutters. Object pro-

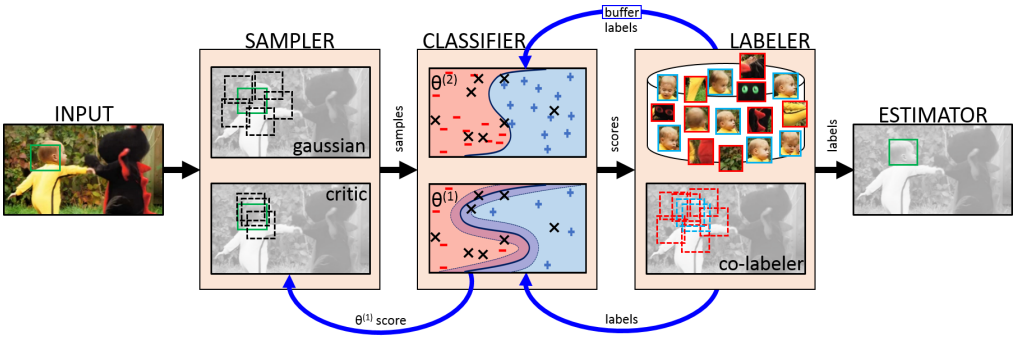

Fig. 2. Schematic of the system. The proposed tracker, collect half of the sample from a Gaussian distribution around the last target state. Meanwhile, the critic learns how to generate transformations that maximally challenge the short-memory classifier ($\theta^{(1)}$), in order to accelerate learning and improve the accuracy. The classifiers exchange information via an active learning scheme, to realize a collaborative robust labeling. The classifiers are then updated and the target state is estimated (See Alg. 1).

posal generators, such as Edge boxes [17], CPMC [18], and Selective Search [19], are a group of models that provide a fine-selected set of candidates that potentially contain the object in the image. These models –provided that they can provide efficient and reliable candidates– can serve as the sampler in a tracking-by-detection pipeline, such as Edge Box which is employed in EBT [20].

In this study, to illustrate the role of sampling in providing better information for the classifier, we address the problem of uninformed sampling by proposing information-maximizing sampling, in which a "critic" unit learns to sample the essential parts of the image: the most informative ones for the classifier. It is realized by using a semi-supervised co-tracking framework in which the information exchange is managed by active learning, selecting the most uncertain samples of each classifier and querying it from the other. In this framework, two classifiers are used: one with long-term memory that is updated infrequently, and the other with a short-term memory updated every frame. The "critic" in this framework learn to challenge the latter classifier by monitoring that classifier's performance over various samples to select future samples that challenge the classifier the most forcing it to collaborate with the long-memory classifier. Using critic solves uninformed sampling, active learning tackles treat samples unequally, co-learning addresses label noise and together with short-long memory mixture resist model drift. The proposed tracker demonstrate a superior performance compared to the state-of-the-art on challenging sequences.

2. PROPOSED METHOD

2.1. Tracking by Detection

Online visual tracking is the task of finding the proper transformation $\mathbf{y}_t$ that transform the previous state $\mathbf{p}_{t-1}$ into the new state $\mathbf{p}_t = \mathbf{p}_{t-1} \circ \mathbf{y}_t$. In tracking-by-detection framework, it is realized by obtaining several samples $\mathbf{x}_t^j (j = 1, \ldots, n)$ from the new frame and evaluating them using a classifier θ_t to distinguish if they contain the target or the background. The most confident sample according to the classifier is typically selected as the next target state. To accommodate target evolutions throughout the tracking scenario, the classifier should be updated.

A typical pipeline for this process is to sample transformations $\mathbf{y}_t \in \mathcal{Y}_t$ using dense sampling or sparse sampling with regard on the previous target state, $\mathbf{p}_t^j = \mathbf{p}_{t-1} \circ \mathbf{y}_t^j \in \mathcal{P}_t$. The samples are defined using these transformations, and their corresponding image patches $\mathbf{x}_t^j \in \mathcal{X}_t$ is selected from image. After an optional feature extraction stage, these image patches are evaluated by classifier θ_t and scored by its scoring function $h : \mathcal{X} \to \mathbb{R}$.

$$s_t^j = h\big(\mathbf{x}_t^{\mathbf{p}_{t-1} \circ \mathbf{y}_t^j} | \theta_t\big). \tag{1}$$

If the score is above a threshold τ, the sample is considered as a possible target match,

$$\ell_t^j = \text{sign}(s_t^j - \tau). \tag{2}$$

A weighted average of the positive samples is selected as the next target state (indicating its location and size),

$$\hat{\mathbf{y}}_t = \sum_k s_t^k \mathbf{y}_t^k \text{ , s.t. } \ell_t^k > 0 (j = 1, \ldots, n). \tag{3}$$

Finally, the classifier is updated by its own labeled data,

$$\theta_{t+1} = u(\theta_t, \mathcal{X}_t, \mathcal{L}_t) \tag{4}$$

in which $u(.)$ is the update function (e.g., budgeted SVM update [3]), and $\ell_t^j \in \mathcal{L}_t$ is the corresponding labels of $\mathcal{X}_t$.

2.2. Information Maximizing Sampling Strategy

Obtaining samples for a discriminative tracker, have been understudied in the literature. While dense sampling using sliding windows, Gaussian sampling around the last known target location ($\mathbf{y}_t^j \sim \mathcal{N}(\mathbf{p}_{t-1}, \Sigma_{search})$), using 1st or 2nd-degree motion models, and particle filters are known as successful practices for sampling, still the need for an informed sampling that uses the content of samples to obtain better samples is needed. In addition, circular shift [21] to increase the number

input : Last state $\mathbf{p}_{t-1}$, Classifiers $\theta_t^{(i)}$, Critic Ψ
output: New state $\mathbf{p}_t$, Updated models $\theta_{t+1}^{(i)}, \Psi_{t+1}$
for $j \leftarrow 1$ **to** n **do**
  if $j < \frac{n}{2}$ **then**
    Random sampling $\mathbf{y}_t^j \sim \mathcal{N}(\mathbf{p}_{t-1}, \Sigma_{search})$
  else
    Guided sampling $\mathbf{y}_t^j \sim g(\mathbf{p}_{t-1}|\Psi)$
  Calculate position and score (eq(7))
  Obtain label and queries (eq(8, 9)
  Calculate error rate and weights (eq(10, 11)
  Update critic (eq(6))
Update classifiers (eq(12, 13)
Estimate the transformation and new state (eq(3))

Algorithm 1: Information Maximizing Sampling Tracker

of positive samples, despite its computation efficiency benefits, inject noise into the tracking loop in the long-term that leads to tracking drift [22]. To address this issue, the content of the samples must be considered in sample selection to realize an informed sampling. For instance, in [20] the silhouette of the target is searched within samples to provide fewer samples with a higher chance of being the target. However as argued in [3], the sampling and classification have two different objectives. While the former tries to provide better samples from the target, the latter tries to construct a better classifier, demanding representative negative samples and supports for defining an accurate classification boundary.

In this study, we take another approach, by exploiting the uncertainties of the classifier, we try to obtain samples that knowing their labels, would maximally improve the classification accuracy, in other words, *most informative samples*.

Recently, generative adversarial networks implemented by a system of two neural networks competing against each other in a zero-sum game framework [23] gains much attention. In this framework, one network is generative which is taught to map from a latent space to a particular data distribution, and the other is a discriminative network that is simultaneously taught to discriminate between true data and synthesized instances produced by the generator. Inspired by this framework, we proposed a "critic" that tries to expose the weaknesses of the tracker's classifier, and the classifier tries to improve its classification in those area to provide a good classification for those sort of samples. To this end, we employed a structured learning for critic, with learning prediction function $g : \mathcal{X}_t \rightarrow \mathcal{Y}_t$. In our approach, a labeled example is a pair $\langle \mathbf{x}^{\mathbf{p}_{t-1}}, \mathbf{y}_t^j \rangle$ where $\mathbf{y}_t^j$ is a challenging transformation given the last known target position $\mathbf{p}_{t-1}$. We learn $G : \mathcal{X}_t \times \mathcal{Y}_t \rightarrow \mathbb{R}$ on-the-fly using a structured-output SVM framework governed by Ψ which introduces a discriminant function that can be used for prediction

$$\mathbf{y}_t^j = g(\mathbf{p}_{t-1}|\Psi) = \max_{k=1,\ldots,j-1} G(\mathbf{p}_{t-1}|\mathbf{y}_t^k, \Psi) \quad (5)$$

The critic is updated with every selected sample $\mathbf{x}_t^j$ and its label ℓ_t^j to help finding the next challenging sample,

$$\Psi \leftarrow u_c(\Psi, \mathbf{x}_t^j, \ell_t^j), \quad (6)$$

where $u_c(.)$ is a budgeted SVM update inspired by [3]. If the last generated sample falls within the uncertain region of the classifier, its addition to the critic reinforce the ability of the critic to generate challenging samples similar to the last sample, otherwise, it signals the critic to explore other ways of generating samples to challenge the main classifier.

2.3. Information Maximizing Sampling Tracking

The proposed tracker is consisted of two classifiers $\theta_t^{(1)}$ and $\theta_t^{(2)}$, having short-term and long-term memory respectively. This mixture of memories balances the stability-plasticity of the tracker. The data exchange of two classifiers is conducted by active learning, in which the most uncertain samples of one classifier is labeled by the other classifier. Finally, these labeled data are used to update the classifiers.

To realize an informed sampling, we proposed a hybrid of Gaussian sampling (based on the last known target position) and critic-generated sampling (that challenges the main classifier to improve its decision boundary). In each frame t, half of the samples are obtained using the Gaussian sampling, classified and use to update the critic with the recent changes of the target and background. Then the critic, finds several challenging samples (*candidates*) for $\theta_t^{(1)}$ using eq(5). If a candidate is not challenging for the classifier, $|h\big(\mathbf{x}_t^{\mathbf{p}_{t-1} \circ \mathbf{y}_t^j} | \theta_t^{(1)}\big)| \geq \tau_t$, it is discarded and a new candidate is seek. The selected samples are scored using

$$s_t^{j,(i)} = h\big(\mathbf{x}_t^{\mathbf{p}_{t-1} \circ \mathbf{y}_t^j} | \theta_t^{(i)}\big) \quad (7)$$

and their labels are determined by

$$\ell_t^j = \begin{cases} \text{sign}(s_t^{j,(1)}) & , s_t^{j,(2)} < \tau_t, s_t^{j,(1)} \geq \tau_t \\ \text{sign}(s_t^{j,(2)}) & , s_t^{j,(1)} < \tau_t, s_t^{j,(2)} \geq \tau_t \\ \text{sign}(\alpha_t^{(1)} s_t^{j,(1)} + \alpha_t^{(2)} s_t^{j,(2)}) & , \text{otherwise} \end{cases} \quad (8)$$

Next, the queries of classifiers (i.e., the data they want the other classifier to label) are determined following the principle of uncertainty sampling [24]

$$q_t^{1 \rightarrow 2} = \{\mathbf{x}_t^j | s_t^{j,(1)} < \tau_t\}. \quad (9)$$

Here, τ_t is selected such that the m most uncertain samples falls in $q_t^{1 \rightarrow 2}$. To calculate the weight of classifiers, first their errors are calculated as the number of mismatches between the classifier label and the co-tracking label,

$$e_t^{(i)} = \sum_j \mathbb{1}(\ell_t^j \neq \text{sign}(s_t^{j,(i)})), \quad (10)$$

then it is used to calculate the weights of classifiers

$$\alpha_t^{(i)} = 1 - \frac{e_t^{(i)} + \epsilon}{\sum_{i \in \{1,2\}} e_t^{(i)} + \epsilon}, \quad (11)$$

where $\mathbb{1}(.)$ is the indicator function and ϵ is a small constant. After having all samples update short-term classifier

$$\theta_{t+1}^{(1)} = u_1(\theta_t^{(1)}, q_t^{2\to1}, \mathcal{L}_t) \tag{12}$$

And long-term one

$$\theta_{t+1}^{(2)} = \begin{cases} u_2(\theta_t^{(2)}, \mathcal{X}_{t-\Delta,\dots,t}, \mathcal{L}_{t-\Delta,\dots,t}) & , t = k\Delta \\ \theta_t^{(2)} & , t \neq k\Delta \end{cases} \tag{13}$$

Then the target transformation is estimated from eq(3) and determine the new state from $\mathbf{p}_t = \mathbf{p}_{t-1} \circ \mathbf{y}_t$. Algorithm 1 summarizes the proposed tracker. Tracker's parameters (n, Σ_{search}, m and Δ) are determined with the cross-validation.

3. EVALUATION

In this section, we compare the proposed tracker, with its baseline (the co-tracker with a long and a short memory) and the state-of-the-art algorithms on the object tracking benchmark (OTB-50 [8]). The sequences of OTB-50 are attributed by one or more tracking challenges: illumination (IV), scale (SV), in-plane rotation (IPR), out-of-plane rotation (OPR), deformation (DEF), occlusion (OCC), out-of-view (OV), background clutter (BC), low resolution (LR), fast motion (FM) and motion blur (MB). The performance of the trackers is compared with the area under curve of success plot and precision plot, on all of the sequences (Figure 3), or a subset of them with the given attribute (Table 1). Success plot indicates the reliability of the tracker and its overall performance while precision plot reflects the accuracy of the localization. Figure 3 presents that using the proposed sampling method by keeping a fixed number of samples significantly improved the performance of the IMST tracker, over its baseline. To establish a fair comparison with the state-of-the-art of tracking-by-detection algorithms, TLD [2] and STRUCK [3] are selected based on the results of [8], MUSTer [5], STAPLE [6] and MEEM [4] are selected based on the results of VOT2016, and VTS [16] and EBT [20] was selected to compare the effectiveness of sampling methods. The results reported here is the average of five independent runs. As Figure 3 and Table 1 illustrates, the proposed tracker outperforms the state-of-the-art. The proposed algorithm also has superior performance in most of the subcategories. High performance in SV, LR, OV, and MB are specifically the results of proposed sampling and comparable results in BC by adding the short/long memory combination.

Table 1. Quantitative evaluation of trackers under different tracking challenges using AUC(%) of success plot on OTB-50. The first, second and third best results are shown in color.

Attribute	VTS	TLD	STRK	MEEM	STPL	MSTR	EBT	IMST
IV	54.3	47.8	53.0	62.3	67.7	72.6	61.2	72.6
SV	51.8	49.1	51.8	58.3	67.6	70.6	58.0	72.4
IPR	55.1	50.4	53.7	57.7	68.9	68.5	56.9	73.5
OPR	54.9	47.8	53.2	62.1	67.5	70.2	61.4	72.5
DEF	54.1	38.2	51.3	61.9	70.4	68.9	64.6	66.1
OCC	52.2	46.1	50.2	60.8	69.1	71.0	59.6	71.8
OV	51.5	53.5	51.5	68.5	61.8	73.3	70.0	71.1
LR	35.0	36.2	33.3	43.5	47.4	50.2	30.6	56.3
BC	56.7	39.4	51.5	67.0	66.9	71.7	67.2	71.2
FM	43.5	44.6	52.0	64.6	55.9	65.0	65.4	64.3
MB	39.4	41.0	46.7	62.8	61.5	65.2	63.8	65.5
FPS	5.3	21.2	11.3	14.2	48.1	8.3	3.8	23.2

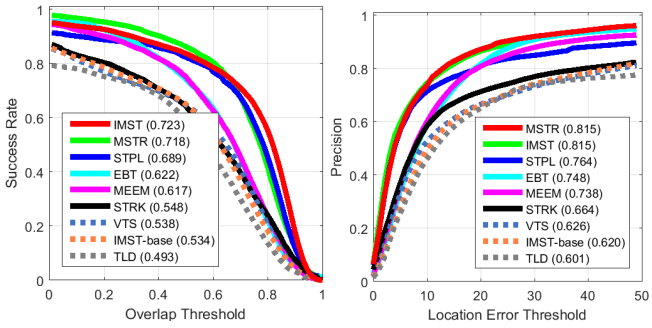

Fig. 3. Quantitative evaluation of trackers using precision plot (left) and success plot (right) for all sequences in OTB-50 [8]. The AUC of plots are used for fair comparison.

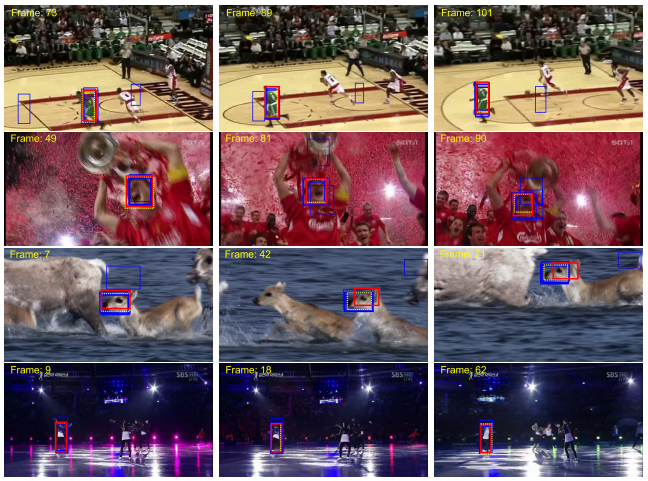

Fig. 4. Exemplary tracking results of proposed tracker (in red) and other evaluated trackers (blue) on several challenging video sequences. The ground truth is depicted in yellow. More results are available from http://ishiilab.jp/member/meshgi-k/imst.html.

4. CONCLUSION

In this study, we have proposed an information maximizing sampling paradigm to be integrated into a discriminative active co-tracker. It is realized by a structured learning scheme which maps the sample space to transformation space and select the most informative samples to accelerate classifier learning and foster the accurate tracking. The proposed tracker, IMST, obtain samples by considering target's spatiotemporal properties and uncertainty-analysis of the classifier and provides the required labels from a long-memory auxiliary classifier and outperformed the state-of-the-art.

5. REFERENCES

[1] Feng Tang, Shane Brennan, Qi Zhao, and Hai Tao, "Co-tracking using semi-supervised support vector machines," in *ICCV'07*.

[2] Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas, "Tracking-learning-detection," *PAMI*, vol. 34, no. 7, pp. 1409–1422, 2012.

[3] Sam Hare, Amir Saffari, and Philip HS Torr, "Struck: Structured output tracking with kernels," in *ICCV'11*, 2011.

[4] Jianming Zhang, Shugao Ma, and Stan Sclaroff, "Meem: Robust tracking via multiple experts using entropy minimization," in *ECCV'14*.

[5] Zhibin Hong, Zhe Chen, Chaohui Wang, Xue Mei, Danil Prokhorov, and Dacheng Tao, "Multi-store tracker (muster): a cognitive psychology inspired approach to object tracking," in *CVPR'15*.

[6] Luca Bertinetto, Jack Valmadre, Stuart Golodetz, Ondrej Miksik, and Philip HS Torr, "Staple: Complementary learners for real-time tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1401–1409.

[7] Thang Ba Dinh, Nam Vo, and Gérard Medioni, "Context tracker: Exploring supporters and distracters in unconstrained environments," in *CVPR'11*, 2011.

[8] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang, "Online object tracking: A benchmark," in *CVPR'13*. IEEE, 2013, pp. 2411–2418.

[9] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang, "Object tracking benchmark," *PAMI*, vol. 37, no. 9, pp. 1834–1848, 2015.

[10] Yuwei Wu, Mingtao Pei, Min Yang, and Yunde Jia, "Robust discriminative tracking via landmark-based label propagation," *TIP*, 2015.

[11] David A Ross, Jongwoo Lim, Ruei-Sung Lin, and Ming-Hsuan Yang, "Incremental learning for robust visual tracking," *IJCV*, vol. 77, no. 1-3, pp. 125–141, 2008.

[12] Hamed Masnadi-Shirazi, Vijay Mahadevan, and Nuno Vasconcelos, "On the design of robust classifiers for computer vision," in *CVPR'10*, 2010.

[13] Helmut Grabner, Christian Leistner, and Horst Bischof, "Semi-supervised on-line boosting for robust tracking," in *ECCV'08*. 2008.

[14] Kourosh Meshgi, Shigeyuki Oba, and Shin Ishii, "Robust discriminative tracking via query-by-committee," in *AVSS'16*, 2016.

[15] Luka Cehovin, Matej Kristan, and Ale Leonardis, "Robust visual tracking using an adaptive coupled-layer visual model," *PAMI*, vol. 35, no. 4, pp. 941–953, 2013.

[16] Junseok Kwon and Kyoung Mu Lee, "Tracking by sampling trackers," in *ICCV'11*. IEEE, 2011, pp. 1195–1202.

[17] C Lawrence Zitnick and Piotr Dollár, "Edge boxes: Locating object proposals from edges," in *European Conference on Computer Vision*. Springer, 2014, pp. 391–405.

[18] Joao Carreira and Cristian Sminchisescu, "Constrained parametric min-cuts for automatic object segmentation," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 3241–3248.

[19] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders, "Selective search for object recognition," *International journal of computer vision*, vol. 104, no. 2, pp. 154–171, 2013.

[20] Gao Zhu, Fatih Porikli, and Hongdong Li, "Beyond local search: Tracking objects everywhere with instance-specific proposals," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 943–951.

[21] João F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista, "High-speed tracking with kernelized correlation filters," *PAMI*, vol. 37, no. 3, pp. 583–596, 2015.

[22] Hamed Kiani Galoogahi, Ashton Fagg, and Simon Lucey, "Learning background-aware correlation filters for visual tracking," *arXiv*, 2017.

[23] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[24] David D Lewis and William A Gale, "A sequential algorithm for training text classifiers," in *ACM SIGIR'94*, 1994, pp. 3–12.