ONLINE SVM AND BACKWARD MODEL VALIDATION BASED VISUAL TRACKING

Dhruv Mullick⋆†*, AV Subramanyam*⋆*, Sabu Emmanuel*†

⋆†Netaji Subhas Institute of Technology, Delhi University, India
⋆Indraprastha Institute of Information Technology, Delhi, India
†College of Computing Sciences and Engineering, Kuwait University, Kuwait

ABSTRACT

Visual object tracking involves the challenging task of scale adaptation to the changing object appearance. Sometimes, this leads to excessive expansion or contraction of the estimated bounding box. Towards this, several generative, discriminatory and hybrid models have been proposed. In this paper, we propose to apply Backward Validation Tracking (BVT) along with an online SVM. BVT has an advantage that it creates a model pool depending on the amount of variation of object's appearance in subsequent frames. Thus while tracking an object, not only is the current appearance taken into account, but so are the previous appearances which are stored in the model pool. Further, in order to improve on the appearance model adaptation, we use an online SVM. The online SVM is a discriminatory algorithm which allows us to distinguish between foreground and background objects. On account of its online nature, the SVM adapts to the updates in the appearance of the target object. We perform extensive experiments on the Object Tracking Benchmark (OTB) dataset. Experimental results prove that the proposed tracker outperforms several popular trackers, in terms of both overlap ratio and precision.

***Index Terms*—** Appearance model updating, backward validation, online support vector machine, visual tracking.

1. INTRODUCTION

Visual tracking is an essential component of many modern day systems, due to its applications in activity recognition, video surveillance, video indexing, traffic monitoring, robot navigation and human-computer interaction [1]. Commonly, we use an appearance model based tracking algorithm, wherein we describe the target in terms of robust features, and try to track it. There are several challenges like occlusion, scale variation and shape deformation involved in developing a robust visual tracking algorithm.

In appearance model based visual tracking, we can use either static or adaptive appearance models. Using static appearance models has the disadvantage that if the target object undergoes an appearance change, then the static appearance assumption does not hold, and the tracker does not remain accurate any longer. In [2, 3, 4], authors suggest that adaptive models perform better, as compared to static models when tracking objects whose appearance may change with time. A good reference to such trackers can be obtained in [5].

Prior to updating the appearance model, we need to classify the incoming data into valid (from object appearance changes) and invalid (resulting from occlusions) data. Updating appearance model poses the challenge that if we do not correctly classify the incoming data, invalid features (appearance changes due to occlusions) would be introduced. As the tracking progresses, the appearance model does not get updated corrrectly, and the incorrect appearance model leads to the tracker drifting. We can make use of the forward validation technique (FVT) [6] for handling drifting during object tracking. However, it is difficult to differentiate between large appearance variations and occlusion, as both of them lead to major changes in the appearance of the tracked object. Therefore, there is a degree of uncertainty [7] involved when using FVT.

Using a backward model validation based tracker (BVT) [7] eliminates this uncertainty from the tracking process. BVT allows us to use the data from the next frame to backward check if the previous model updation was valid. The appearance model is updated only if the target object's appearance is similar to the existing model. If instead, there is a large change in the appearance, then we create another appearance model, which is a duplicate of the current model. This new model is updated with the incoming data. In the next frame, BVT backwards checks if the updation was valid. If it is, then it makes use of the new appearance model, and the original model if otherwise. A few visual object trackers [8, 9, 10] have made use of the BVT algorithm. To represent visual targets in a robust manner, BVT uses an adaptive feature fusion method based on incremental principle component analysis (IPCA) [3] and colour histogram [11]. IPCA and colour histogram prove to be complementary in nature, wherein IPCA is robust to distortions due to illumination changes and rotations, and colour histogram is adaptive to shape deformation [7].

On the basis of appearance modeling, we can classify tracking algorithms as generative and discriminative. Generative algorithms search for the best matched image region between the candidates and the appearance model [12, 3]. Another approach for object tracking can be to use a discriminative model which learns to distinguish between the foreground and the background objects [13]. Popular trackers like Struck [14], MIL [15] and Semi-Boost [16] use a discriminative approach. In our tracker, we make use of a discriminative model - Online Support Vector Machine (SVM) [17] along with our generative model. This is similar to [18], which explores using a hybrid of a generative model and an online SVM (discriminative model).

An online support vector machine allows us to maintain an adaptive discriminative model, since it can be trained at every frame using image patches. One could argue that an SVM may not be adaptive because while evaluating candidates from frame n, it pays equal importance to training data from frame $n-1$ and from frame $n-m$ ($m >> 1$). To solve this problem, we add a forgetting factor [19] to the SVM so that while predicting for frame n, we pay a greater emphasis to the training data from frames closest to the current frame.

We train our SVM to learn to differentiate between foreground and background objects. While training, we make use of the patches where the object is most likely to be present. The confidence is judged from the output returned from the generative model. In order to improve on the discrimination capabilities of the SVM, we make use of the HOG features [20] of the images for training and evaluation. In [21], authors suggest that Linear SVMs perform better when used with HOG features of the images, instead of the raw image pixels. We expand this research to using HOG features in a Gaussian Online SVM, and find that in this case too, the results improve drastically.

The results obtained from the generative model are fed to the discriminative model, which returns the candidate which best fits as a foreground object. Experiments on the OTB dataset [22] prove that the proposed algorithm outperforms several popular object trackers.

Rest of the paper is organised as follows: in Section 2 we explain in detail the proposed tracking algorithm; the experiments performed, along with their analysis, are outlined in Section 3; in Section 4, we conclude our paper.

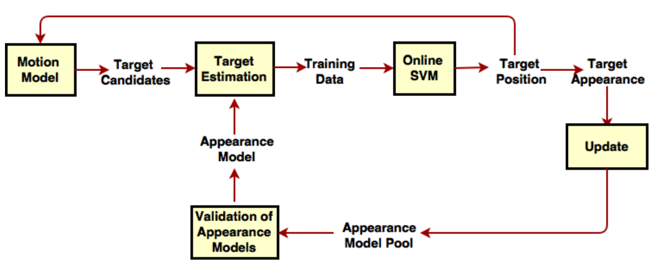


Fig. 1: Abstract diagram of the proposed tracking algorithm

2. PROPOSED TRACKING ALGORITHM

We propose a tracking algorithm - BVTSVM, which uses a combination of BVT [7] and an Online SVM [17]. We first use a Particle Filter to generate a set of candidate solutions. By means of IPCA and Colour Histogram, the candidates are evaluated and we obtain a confidence value corresponding to each of them. While evaluating the confidence, we select the closest appearance model from the pool for every candidate. Further, we apply an online SVM to distinguish between foreground and background objects. The SVM is evaluated on all the candidate solutions, and the one with the highest probability of being a foreground object is selected as the target object. This target object image is used to update our appearance model pool. If there is a major change in the appearance model, then it is added as a new model to the pool. Otherwise, the original model is updated. We train the online SVM every fifth frame, similar to [3]. The training is done using the candidates we had evaluated for that frame. Instead of training with all of them, we select the ones with the top five confidence values as foreground objects, and the ones with the least five confidence values as background objects. Using these ten samples, we train our SVM to learn to differentiate between foreground and background objects.

2.1. Particle Filter Framework

Particle Filter framework [23] is used while generating candidate target solutions. More specifically, we can estimate the posterior probability by using random particles, each of which has a weight associated to it. The posterior probability at time t is calculated using:

$$p(\mathbf{x}_t|\mathbf{z}_{1..t}) \propto p(\mathbf{z}_t|\mathbf{x}_t) \int p(\mathbf{x}_t|\mathbf{x}_{t-1}) p(\mathbf{x}_{t-1}|\mathbf{z}_{1..t-1}) d\mathbf{x}_{t-1} \tag{1}$$

In this equation, $\mathbf{x}_t$ represents state of target object at time t [7], and is a function of affine transformation parameters $(x_t, y_t, \theta_t, \alpha_t, \phi_t)$. (x, y) gives the bounding box centre, θ_t gives the rotations angle, α_t gives the aspect ratio and ϕ_t gives the skew direction. $\mathbf{z}_t$ represents state of observed appearance of target object. Here, $p(\mathbf{z}_t|\mathbf{x}_t)$ denotes the appearance model at time t [7].

2.2. Motion Model

As the object under consideration is in motion, we can say that its state changes at every frame. Following [3] and [24], we predict $\mathbf{x}_t$ from $\mathbf{x}_{t-1}$ by adding a mean Gaussian distribution independent from $\mathbf{x}_t$. Thus the motion of the target can be estimated as

$$\mathbf{x}_t = \mathbf{x}_{t-1} + \boldsymbol{\epsilon}_{(x,y,\theta,\alpha,\phi)} \tag{2}$$

where $\boldsymbol{\epsilon}_{(x,y,\theta,\alpha,\phi)}$ represents a Gaussian noise vector whose elements are generated by Gaussian distributions with zero mean and variances σ_x^2, σ_y^2, σ_θ^2, σ_α^2 and σ_ϕ^2 respectively.

2.3. Appearance Model

As time progresses, the appearance of the target object changes, and there is a definite need to update the appearance model. Or else, the tracker would lose track of the object. As explained in [7], we make use of dual features from IPCA [3] and colour histogram [11]. IPCA is used to generate multi-view appearance models for the candidate solutions. IPCA is robust to distortions due to illumination changes and rotations, however it is sensitive to shape deformation [7]. On the other hand, colour histogram is robust to shape deformations and sensitive to distortions. IPCA and colour histogram prove to be complementary in nature. Based on the discrimination of colour histogram features between target and background, we adaptively fuse these two features. Using the dual feature, we can represent both deformable and rigid targets with rotation and illumination variation [7].

Let $\mathbf{I}_t$ be the wrapped grey scale image template predicted from $\mathbf{x}_t$, then the posterior probability that $\mathbf{I}_t$ was generated from the target object is given by $p(\mathbf{I}_t|\mathbf{x}_t)$. Similarly, if $\mathbf{H}_t$ is the colour histogram feature vector predicted from $\mathbf{x}_t$, then the posterior probability that $\mathbf{H}_t$ was generated from the target object is given by $p(\mathbf{H}_t|\mathbf{x}_t)$. These probabilities are then adaptively fused as ([7])

$$p(\mathbf{z}_t|\mathbf{x}_t) = p(\mathbf{I}_t|\mathbf{x}_t)p(\mathbf{H}_t|\mathbf{x}_t) \tag{3}$$

By keeping certain thresholds related to the IPCA and Colour Histogram features, BVT judges whether to create a new appearance model, or to update the existing model in the model pool.

2.4. Online Support Vector Machine

For small datasets, SVMs have a better discriminating capability than other models [18]. In order to adapt to the changing object appearance, we maintain an incrementally trained SVM. We make use of a forgetting factor [19] while evaluating candidate solutions on the SVM as we need to ensure that we focus more on the recently acquired training data. The SVM training and evaluation gets slower with time, as it is trained with more data. Hence, we train our SVM at every fifth frame, and find that the SVM continues to be adequately adaptive. We train and evaluate our SVM on the HOG features of the image patches, instead of the raw features. We find that there is a definite performance improvement on doing so. [21] suggests that HOG improves performance of Linear SVMs. We find the same true for Gaussian SVMs as well. [1] While evaluating the candidates, a conventional SVM would focus equally on the candidate data obtained from the frames encountered. However, in order to ensure that our SVM adapts to the ever changing object appearance, we use a forgetting factor [19]. We use a forgetting factor of the form $-1/f$, where f is the frame whose candidates are being evaluated. From the nature of the forgetting factor, it is evident that as the tracking progresses, the SVM pays more importance to the candidate data obtained from the frames closer to f.

[1] To train our online SVM, we use the code [25] written by C.P. Diehl, which is based on the theory he has suggested in [17].

3. EXPERIMENTS

We evaluate our tracker - BVTSVM, on the OTB dataset [22], which contains 50 video sequences. The dataset covers challenging situations for visual tracking, which allows for a thorough evaluation. We strictly follow the Visual Tracker Benchmark protocol v1.0 [22] while evaluating our tracker. In order to evaluate our tracker, we make use of the popular metrics: Success rate and Precision. Success measures the overlap between the bounding boxes. The overlap score is defined as $S = \frac{|r_t \cap r_a|}{|r_t \cup r_a|}$, where r_t is the tracked bounding box and r_a is the ground truth bounding box. $\cup$ and $\cap$ refer to the union and intersection respectively, of the two regions. Typically, we consider $S > 0.5$ to be successful tracking. Success Rate is defined as the percentage of frames successfully tracked. Precision metric gives the percentage of frames in which the predicted center location is within a given threshold distance from the ground truth. This threshold, by convention, is set at 20 pixels. For Success, we rank the trackers on the basis of their Area Under Curve (AUC). For Precision, we rank them on the basis of the results at error threshold of 20.

For BVT, we make use of the parameters suggested by the authors in [7]: patch size of $\mathbf{I}_t$ is set as 64x64; the batch size for updating the eigenbasis is set to 5; the colour histogram bins N_h, N_v and N_s are set to 10; the size of the appearance model pool is set to 4; model evaluation thresholds for IPCA and Colour Histogram are set to 120 and 0.2 respectively. For the particle filtering framework, we use number of particles $= 300$. For the online SVM: the regularisation parameter for the SVM is set to 0.1; type is set to Gaussian; scale is set to 3; number of frames after which it is used for evaluation, is set to 30.

3.1. Tracking Results

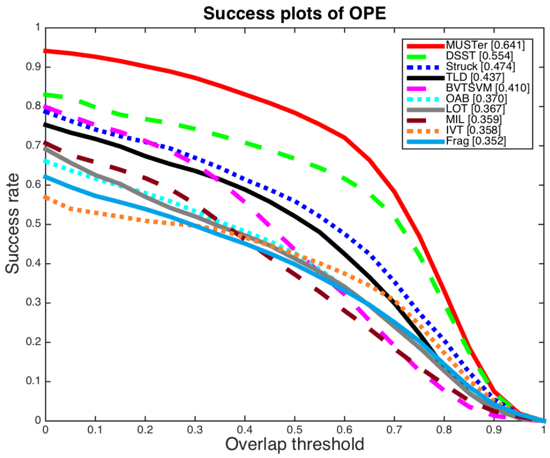
Fig. 2.(a), 2.(b) show the success and precision plots respectively for the proposed tracker (BVTSVM) and 9 popular trackers for the 50 OTB video sequences [22] under the One Pass Evaluation (OPE) scheme.

In Fig. 3. we show the qualitative comparison of the performances of the top-5 trackers - MUSTer [26], DSST [27], Struck [14], BVTSVM (proposed) and TLD [28]. The rankings of the trackers are as per the success plots.

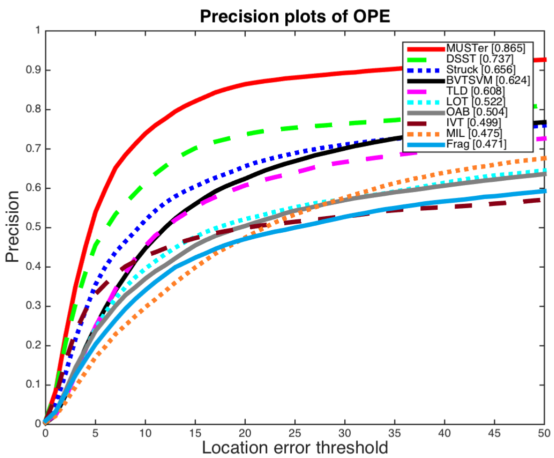
Table 1. and Table 2. show a comparison of the Success Rates and Precisions on 15 OTB video sequences, between the proposed tracker and the popular trackers: Frag [29], OAB [30], MIL [15], CT [31], TLD [28], LOT [32], Struck [14], and the modern trackers DSST [27] and MUSTer [26].

3.2. Result Analysis

The proposed algorithm tracks 0.8 frames per second on an average, when applied to nearly 500 frames of a sequence. We find that the training and evaluation speeds decrease as the number of frames trained on increase. This is expected, as we make use of an SVM, whose number of support vectors



(a)



(b)

Fig. 2: Success plots (a) and Precision plots (b) for all the 50 benchmark sequences using One Pass Evaluation (OPE)

Table 1: Success Rates for trackers. Red: best performance, Green: second best, Blue: third best.

	Frag	OAB	MIL	CT	TLD	LOT	Struck	DSST	MUSTer	**BVTSVM**
Boy	46.01	98.67	38.53	68.77	93.52	64.95	97.50	100.00	99.16	**55.81**
Couple	62.85	45.00	67.14	68.57	100.00	56.42	54.28	10.71	80.00	**85.00**
David3	81.34	33.73	68.25	34.92	10.31	93.65	33.73	52.77	100.00	**83.73**
Football	91.71	36.46	73.75	78.45	41.16	83.14	66.02	70.99	62.70	**68.23**
Football1	33.78	24.32	78.37	8.10	39.18	41.89	87.83	39.18	39.18	**67.56**
Jogging1	70.03	86.31	22.47	22.47	96.74	6.51	22.47	22.47	94.78	**92.18**
Jogging2	57.32	49.51	16.28	14.00	83.06	13.68	24.75	18.24	97.39	**66.77**
Liquor	37.27	48.24	20.10	20.85	58.24	96.26	40.60	40.89	98.04	**70.87**
Singer1	21.93	23.36	27.63	24.78	99.14	24.21	29.91	100.00	86.32	**91.73**
Skiing	3.70	9.87	7.40	7.40	7.40	1.23	3.70	6.17	4.93	**35.80**
Soccer	19.38	8.67	15.5	20.15	12.24	21.68	15.56	39.03	16.58	**22.70**
Subway	57.71	21.71	79.4	76.57	22.85	68.00	90.85	22.28	100.00	**57.71**
Tiger1	31.23	9.87	9.74	24.64	45.55	16.33	18.33	59.31	66.76	**4.23**
Tiger2	12.32	13.97	44.65	36.98	17.26	17.53	64.93	29.58	45.20	**18.35**
Walking2	34.80	38.00	38.00	38.40	34.00	39.00	43.40	100.00	100.00	**39.20**

increase with every training.

From Fig. 1. and Fig. 2., we find that the proposed tracker (BVTSVM) beats several of the popular trackers in terms of both Success Rate and Precision. We find that:

- In terms of the overall success, our tracker ranks 5th among the 10 trackers considered.

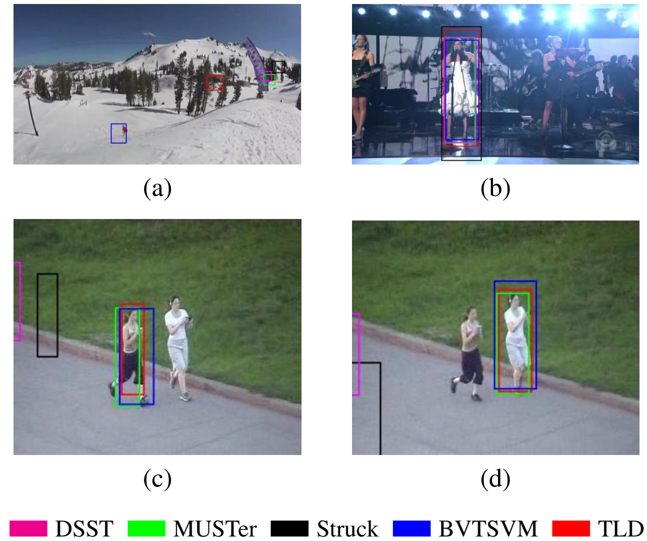


Fig. 3: Qualitative comparison of the performance of the top-5 trackers, as per the ranking from the success plots.

Table 2: Precisions for trackers. Red: best performance, Green: second best, Blue: third best ones.

	Frag	OAB	MIL	CT	TLD	LOT	Struck	DSST	MUSTer	**BVTSVM**
Boy	48.17	100.0	84.55	93.02	100.0	66.61	100.0	100.0	100.0	**96.01**
Couple	90.00	47.14	67.85	69.28	100.0	62.85	73.57	10.71	73.57	**96.42**
David3	76.98	39.28	73.80	41.26	11.11	98.80	33.73	59.92	33.73	**92.46**
Football	98.61	42.54	79.00	79.83	80.38	99.72	75.13	79.83	75.13	**83.97**
Football1	58.10	37.83	100.0	35.13	55.40	100.0	100.0	94.59	100.0	**95.94**
Jogging1	73.94	98.69	23.12	23.12	97.39	19.21	24.10	23.12	24.10	**96.09**
Jogging2	56.02	57.98	18.56	16.61	85.66	16.28	25.40	18.56	25.40	**86.97**
Liquor	35.38	40.83	19.87	20.85	58.75	93.62	39.00	40.43	39.00	**74.78**
Singer1	24.78	88.60	50.14	84.04	100.0	23.64	64.10	100.0	64.10	**99.43**
Skiing	3.70	13.58	7.40	8.64	12.34	2.46	3.7	13.58	3.7	**70.37**
Soccer	19.89	8.67	19.13	21.93	11.47	33.41	25.25	69.38	25.25	**24.48**
Subway	74.28	24.00	99.42	98.85	25.14	98.28	98.28	24.57	98.28	**93.71**
Tiger1	28.93	8.59	9.45	21.48	45.55	16.33	17.47	57.02	17.47	**9.39**
Tiger2	12.60	13.9	41.36	36.43	38.63	18.08	63.01	29.86	63.01	**43.83**
Walking2	34.20	48.60	40.60	43.20	42.60	39.20	98.20	100.0	98.20	**39.4**

- In terms of the overall precision, our tracker ranks 4th among the 10 trackers considered.

On examining the success rates (Table 1.) and precisions (Table 2.) for the trackers for individual videos, we find that our tracker beats most of the popular trackers in several cases. The proposed tracker is performing well, being the 3rd best tracker very often.

4. CONCLUSION

We propose a visual object tracker - BVTSVM, based on backward validation technique, which uses a combination of an adaptive and a discriminative model. Experimental results show that the tracker performs well against several challenging videos from the OTB dataset. This is due to the fact that we apply a generative model in terms of IPCA as well as a discriminative model using online SVM. In addition, BVT employs the advantage of a model pool through which multiple appearances of the object can be taken into account. This is particularly useful in occluded videos. Our proposed tracker has a consistently good performance, and proves to be a robust visual object tracking algorithm.

5. REFERENCES

[1] Alper Yilmaz, Omar Javed, and Mubarak Shah, "Object tracking: A survey," *ACM Comput. Surv.*, vol. 38, no. 4, Dec. 2006.

[2] Allan D Jepson, David J Fleet, and Thomas F El-Maraghi, "Robust online appearance models for visual tracking," *IEEE transactions on pattern analysis and machine intelligence*, vol. 25, no. 10, pp. 1296–1311, 2003.

[3] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *International Journal of Computer Vision*, vol. 77, no. 1-3, pp. 125–141, May 2008.

[4] Yuru Wang, Longkui Jiang, Qiaoyuan Liu, and Minghao Yin, "Optimal appearance model for visual tracking," *PLOS ONE*, vol. 11, no. 1, pp. 1–15, 01 2016.

[5] Samuele Salti, Andrea Cavallaro, and Luigi Di Stefano, "Adaptive appearance modeling for video tracking: Survey and evaluation," *Trans. Img. Proc.*, vol. 21, no. 10, pp. 4334–4348, Oct. 2012.

[6] L. Matthews, T. Ishikawa, and S. Baker, "The template update problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 810–815, June 2004.

[7] Yuan Yuan, Sabu Emmanuel, Yuming Fang, and Weisi Lin, "Visual object tracking based on backward model validation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 11, 2014.

[8] Y. Yuan, H. Yang, Y. Fang, and W. Lin, "Visual object tracking by structure complexity coefficients," *IEEE Transactions on Multimedia*, vol. 17, no. 8, pp. 1125–1136, Aug 2015.

[9] W Jasper Princy Shinora, L Agilandeeswari, and K Muralibabu, "An adaptive approach for validation in visual object tracking," *Procedia Computer Science*, vol. 58, pp. 478–485, 2015.

[10] T. Benlefki, R. Liu, B. Dai, and M. B. Boubekeur, "Combining color and vibe foreground mask for better tracking in video sequences," in *2015 8th International Congress on Image and Signal Processing (CISP)*, Oct 2015, pp. 18–22.

[11] Patrick Pérez, Carine Hue, Jaco Vermaak, and Michel Gangnet, "Color-based probabilistic tracking," in *Proceedings of the 7th European Conference on Computer Vision-Part I*, London, UK, UK, 2002, ECCV '02, pp. 661–675, Springer-Verlag.

[12] X.Mei and H.Ling, "Robust visual tracking and vehicle classification via sparse representation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 11, pp. 2259–2272, Nov 2011.

[13] Hieu T. Nguyen and Arnold W. M. Smeulders, "Robust tracking using foreground-background texture discrimination," *International Journal of Computer Vision*, vol. 69, no. 3, pp. 277–293, 2006.

[14] S. Hare and S. Golodetz et al, "Struck: Structured output tracking with kernels," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 10, pp. 2096–2109, Oct 2016.

[15] B. Babenko, M.-H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1619–1632, 2011.

[16] Helmut Grabner, Christian Leistner, and Horst Bischof, "Semi-supervised on-line boosting for robust tracking," in *Proceedings of the 10th European Conference on Computer Vision: Part I*, Berlin, Heidelberg, 2008, ECCV '08, pp. 234–247, Springer-Verlag.

[17] C. P. Diehl and G. Cauwenberghs, "Svm incremental learning, adaptation and optimization," in *Proceedings of the International Joint Conference on Neural Networks, 2003.*, July 2003, vol. 4, pp. 2685–2690 vol.4.

[18] Qian Yu, Thang Ba Dinh, and Gérard Medioni, "Online tracking and reacquisition using co-trained generative and discriminative trackers," in *Proceedings of the 10th European Conference on Computer Vision: Part II*, Berlin, Heidelberg, 2008, ECCV '08, pp. 678–691, Springer-Verlag.

[19] Hiroyuki Funaya, Yoshihiko Nomura, and Kazushi Ikeda, *A Support Vector Machine with Forgetting Factor and Its Statistical Properties*, pp. 929–936, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[20] Navneet Dalal and Bill Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. IEEE, 2005, vol. 1, pp. 886–893.

[21] Hilton Bristow and Simon Lucey, "Why do linear svms trained on hog features perform so well?," *arXiv preprint arXiv:1406.2419*, 2014.

[22] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang, "Online object tracking: A benchmark," *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[23] M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *Trans. Sig. Proc.*, vol. 50, no. 2, pp. 174–188, Feb. 2002.

[24] P. Zhang, S. Emmanuel, and M. Kankanhalli, "Pedestrian tracking based on hidden-latent temporal markov chain," in *Proceedings of Advances in Multimedia Modeling*, Jan. 2011, pp. 285–295.

[25] "Incremental svm learning," https://github.com/diehl/Incremental-SVM-Learning-in-MATLAB.

[26] Z. Hong, Zhe Chen, C. Wang, X. Mei, D. Prokhorov, and D. Tao, "Multi-store tracker (muster): A cognitive psychology inspired approach to object tracking," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 749–758.

[27] Martin Danelljan, Gustav Haumlger, Fahad Shahbaz Khan, and Michael Felsberg, "Accurate scale estimation for robust visual tracking," in *Proceedings of the British Machine Vision Conference*. 2014, BMVA Press.

[28] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1409–1422, July 2012.

[29] Amit Adam, Ehud Rivlin, and Ilan Shimshoni, "Robust fragments-based tracking using the integral histogram," in *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1*, Washington, DC, USA, 2006, CVPR '06, pp. 798–805, IEEE Computer Society.

[30] Helmut Grabner, Michael Grabner, and Horst Bischof, "Real-time tracking via on-line boosting," in *Proceedings of BMVC*, 2006, pp. 1–10.

[31] Kaihua Zhang, Lei Zhang, and Ming-Hsuan Yang, "Real-time compressive tracking," in *Proceedings of the 12th European Conference on Computer Vision - Volume Part III*, Berlin, Heidelberg, 2012, ECCV'12, pp. 864–877, Springer-Verlag.

[32] Shaul Oron, Aharon Bar-Hillel, Dan Levi, and Shai Avidan, "Locally orderless tracking," *International Journal of Computer Vision*, vol. 111, no. 2, pp. 213–228, 2015.