An Experimental Comparison of Online Object Tracking Algorithms

Qing Wang[a], Feng Chen[a], Wenli Xu[a], and Ming-Hsuan Yang[b]

[a]Tsinghua University, Beijing, China
[b]University of California at Merced, Calfironia, USA

ABSTRACT

This paper reviews and evaluates several state-of-the-art online object tracking algorithms. Notwithstanding decades of efforts, object tracking remains a challenging problem due to factors such as illumination, pose, scale, deformation, motion blur, noise, and occlusion. To account for appearance change, most recent tracking algorithms focus on robust object representations and effective state prediction. In this paper, we analyze the components of each tracking method and identify their key roles in dealing with specific challenges, thereby shedding light on how to choose and design algorithms for different situations. We compare state-of-the-art online tracking methods including the IVT,[1] VRT,[2] FragT,[3] BoostT,[4] SemiT,[5] BeSemiT,[6] L1T,[7] MILT,[8] VTD[9] and TLD[10] algorithms on numerous challenging sequences, and evaluate them with different performance metrics. The qualitative and quantitative comparative results demonstrate the strength and weakness of these algorithms.

Keywords: Object tracking, online algorithm, appearance model, performance evaluation.

1. INTRODUCTION

The goal of object tracking is to estimate the locations and motion parameters of a target in an image sequence given the initialized position in the first frame. Research in tracking plays a key role in understanding motion and structure of objects. It finds numerous applications including surveillance,[11] human-computer interaction,[12] traffic pattern analysis,[13] recognition,[14] medical image processing,[15] to name a few. Although object tracking has been studied for several decades, and numerous tracking algorithms have been proposed for different tasks, it remains a very challenging problem. There exists no single tracking method that can be successfully applied to *all* tasks and situations. Therefore, it is crucial to review recent tracking methods, and evaluate their performances to show how novel algorithms can be designed for handling specific tracking scenarios.

A typical tracking system consists of three components: object representation, dynamic model, and search mechanism. As such, tracking algorithms can be categorized in numerous ways. Object representation is a key component as it directly corresponds to the core challenge of tracking, i.e., how to match object appearance despite all the influencing factors. Moreover, it also determines what objective function can be used for searching the target of interest in frames. To deal with the problem of appearance variations, recent tracking algorithms focus on adaptive object representation schemes based on generative or discriminative formulations. A dynamic model, either predefined or learned from certain training data, is often used to predict the possible target states (e.g., motion parameters) in order to reduce the search space and computational load. Since it is difficult to adapt an effective dynamic model for fast motion and as a result of faster processors, most current tracking algorithms use random walk model to predict the likely states.

Object tracking algorithms can be categorized as either deterministic or stochastic based on their search mechanisms. With the target of interest represented in some feature space, object tracking can always be reduced to a search task and formulated as an optimization problem. That is, the tracking results are often obtained by minimizing or maximizing an objective function based on distance, similarity or classification measures. To optimize the objective function, deterministic methods are formulated and solved with differential algorithms such as gradient descent or its variants. The Kanade-Lucas-Tomasi algorithm[16] and the mean-shift tracking algorithm[17] are deterministic methods, in which the sum of squared distance (SSD) and Bhattacharyya distance are used in the objective functions, respectively. The Kalman filter[18] is also a deterministic method based on linear dynamic models. Gradient descent based deterministic methods are usually efficient, but often suffer from local minimum problems. Different from differential optimization, sampling-based methods

Corresponding author: Ming-Hsuan Yang (mhyang@ucmerced.edu).

can be used to avoid local minimum problems at the expense of higher computational load. Stochastic methods usually optimize the objective function by considering observations over multiple frames within a Bayesian formulation. It improves robustness over deterministic methods by its capability of escaping from local minimum with much lower computational complexity than sampling-based methods that operate on each frame independently. The condensation algorithm[19] is an effective stochastic method that deals with nonlinear objective functions and non-Gaussian dynamic models.

Generative methods track a target object by searching for the region most similar to the reference model in each frame. To deal with the above-mentioned challenges in object tracking, most recent generative methods learn robust static or online appearance models. Black et al.[20] learn a subspace model offline to represent target objects at fixed views. Jepson et al.[21] use a Gaussian mixture model with an online expectation maximization (EM) algorithm to handle target appearance variations during tracking, whereas Ross et al.[1] present an online subspace algorithm to model target appearance. Adam et al.[3] develop the fragments-based appearance model to overcome pose change and partial occlusion problems, and Mei et al.[7] present a template based method using sparse representation as it has been shown to be robust to partial occlusion and image noise. Recently, Kwon et al.extend the conventional particle filter framework with multiple dynamic and observation models to account for appearance variation.[9] While these generative methods perform well, they nevertheless do not take rich scene information into account which can be useful in separating target objects from background clutters.

Discriminative methods pose object tracking as a binary classification problem in which the task is to distinguish the target region from the background. Avidan[22] trains a classifier offline with the support vector machine (SVM) and combines it with optical flow for object tracking. Collins et al.[2] propose a tracking method to select discriminative low-level color features online for tracking, whereas Avidan[23] uses an online boosting method to classify pixels belonging to foreground and background. Grabner et al.[4] develop a tracking method based on online boosting, which selects features to account for appearance variations of the object caused by out-of-plane rotations and illumination change. Babenko et al.[8] use multiple instance learning (MIL) to handle ambiguously labeled positive and negative data obtained online to reduce visual drift caused by classifier update. Recently, Kalal et al.[10] treat sampled data during tracking as unlabeled ones and exploit their underlying structure to select positive and negative samples for update.

Several criteria such as success rate and center location error have been used in the tracking literature for performance evaluation. However, these methods are often evaluated with a few sequences and it is not clear which algorithm should be used for specific applications. To this end, this paper focuses on evaluating the most recent tracking algorithms in dealing with different challenges. First, we demonstrate why adaptive models are crucial to deal with the inevitable appearance change of the target and background over time. Second, we analyze tracking algorithms using the three above-mentioned components and identify their key roles to different challenging factors. Finally, we compare state-of-the-art tracking algorithms on several challenging sequences with different evaluation criteria. The evaluation and analysis are not only useful for choosing appropriate methods for specific applications but also beneficial for developing new tracking algorithms.

2. ADAPTIVE APPEARANCE MODELS

One of the most challenging factors in object tracking is to account for appearance variation of the target object caused by change of illumination, deformation and pose. In addition, occlusion, motion blur and camera view angle also pose significant difficulties for algorithms to track target objects. If a tracking method is designed to account for translational motion, then it is unlikely to handle in-plane and out-of-plane rotations or scale change of objects. For certain applications with limited illumination change, static appearance models based on SIFT,[24] HOG[25] and LBP[26] descriptors may suffice. For applications where objects undergo limited deformation, holistic representations, e.g., histograms, may work well although they do not encode spatial structure of objects. If the object shape does not change much, then representation schemes based on contour or silhouette[19] can be used to account for out-of-plane rotation. When a tracking algorithm is designed to account for in-plane motion and scale change with the similarity transform, a static appearance model may be an appropriate option. However, situations arise where different kinds of variations need to be considered simultaneously. Since it is very difficult to develop a static representation invariant to all appearance change, adaptive models are crucial for robust tracking performance.

While a dynamic model is often used mainly to reduce the search space of states, it inevitably affects the tracking results especially when the objects undergo fast motion. In some cases, a dynamic model facilitates reinitialization of a tracker after partial or full occlusions. For example, if the target state can be predicted well even when temporal occlusion occurs, it will be easy to relocate the target when it reappears in the scene. Aside from the predicated states, an object detector or sampling of state variables can also be utilized to handle occlusion. Table 1 summarizes challenges often encountered in object tracking and the corresponding solutions. It is evident that object representation plays a key role to deal with appearance change of the target object and background. Furthermore, an adaptive appearance model that accounts for all appearance variations online is of great importance for robust object tracking. However, online update methods may inadvertently affect the tracking performance. For example, if an appearance model is updated with noisy observations, tracking errors will be accumulated and result in visual drifts.

Table 1: Challenges and solutions.

Challenge	Solution
illumination change	Use descriptors which are not sensitive to illumination change; Adapt the appearance model to account for illumination change
object deformation	Use object representation which is not sensitive to deformation; Adapt the object representation to account for deformation
in-plane rotation	Use state model that accounts for the similarity transformation; Adapt object representation to such appearance change
out-of-plane rotation	Choose object representations which are insensitive to out-of-plane pose change; Adapt object representation to such appearance change
partial occlusion	Use parts-based model which are not sensitive to partial occlusion; Employ sampling of the state space so that the tracker may be reinitialized when the target reappears
full occlusion	Search the state space exhaustively so that the tracker can be reinitialized when the target reappears
fast object motion or moving background	Sophisticated dynamic model; Search a large region of the state space

3. ONLINE TRACKING ALGORITHMS

A typical tracking system is composed of three components: object representation, dynamic model and search mechanism. Since different components can deal with different challenges of object tracking, we analyze recent online tracking algorithms accordingly and show how to choose or design robust online algorithms for specific situations.

3.1 Object Representation

An object can be represented by either holistic descriptors or local descriptors. Color histograms and raw pixel values are common holistic descriptors. Color histograms have been used in the mean-shift tracking algorithm[17] and the particle-based method.[27] The advantages of histogram-based representations are their computational efficiency and effectiveness to handle shape deformation as well as partial occlusion. However, they do not exploit the structural appearance information of target objects. In addition, histogram-based representations are not designed to handle scale change although some efforts have been made to address this problem.[28,29] Holistic appearance models based on raw intensity values are used in the Kanade-Lucas-Tomasi algorithm,[30] the incremental subspace learning tracking method,[1] the incremental tensor subspace learning method[31] and the ℓ_1-minimization based tracker.[7] However, tracking methods based on holistic representation are sensitive to partial occlusion and motion blur.

Filter responses have also been used to represent objects. Haar-like wavelets are used to describe objects for boosting-based tracking methods.[4,8] Porikli et al.[32] use features based on color and image gradients to characterize object appearance with update for visual tracking. Local descriptors have also been widely used in object tracking recently due to their robustness to pose and illumination change. Local histograms and color information are utilized for generating confidence maps from which likely target locations can be determined.[23] Features based on local histograms are selected to represent objects in the fragments-based method.[3] It has been shown that an effective representation scheme is the key to deal with appearance change in object tracking.

3.2 Adaptive Appearance Model

As mentioned above, it is crucial to update appearance model for ensuring robust tracking performance and much attention has been paid in recent years to address this issue. The most straightforward method is to replace the current appearance model (e.g., template) with the visual information from the most recent tracking result. Other update algorithms have also been proposed, such as incremental subspace learning methods,[1,31] adaptive mixture model,[21] and online boosting-based trackers.[4,23] However, simple update with recently obtained tracking results can easily lead to significant drifts since it is difficult to determine whether the new data are noisy or not. Consequently, drifting errors are likely to accumulate gradually and tracking algorithms eventually fail to locate the targets. To reduce visual drifts, several algorithms have been developed to facilitate adaptive appearance models in recent years. Matthews et al.[33] propose a tracking method with the Lucas-Kanade algorithm by updating the template with the results from the most recent frames and a fixed reference template extracted from the first frame. In contrast to supervised discriminative object tracking, Grabner et al.[5] formulate the update problem as a semi-supervised task where the drawn samples are treated as unlabeled data. The task is then to update a classifier with both labeled and unlabeled data. Specific prior can also be used in this semi-supervised approach[6] to reduce drifts. Babenko et al.[8] pose the tracking problem within the multiple instance learning (MIL) framework to handle ambiguously labeled positive and negative data obtained online for reducing visual drifts. Recently, Kalal et al.[10] also pose the tracking problem as a semi-supervised learning task and exploit the underlying structure of the unlabeled data to select positive and negative samples for update. While much progress has been made on this topic, it is still a difficult task to determine when and which tracking results should be updated in adaptive appearance models to reduce drifts.

3.3 Motion Model

The dimensionality of state vector, $\mathbf{x}_t$, at time t depends on the motion model that a tracking method is equipped with. The most commonly adopted models are translational motion (2 parameters), similarity transform (4 parameters), and affine transform (6 parameters). The classic Kanade-Lucas-Tomasi algorithm[16] is designed to estimate object locations although it can be extended to account for affine motion.[33] The tracking methods[1,7,31] account for affine transformation of objects between two consecutive frames. If an algorithm is designed to handle translational movements, the tracking results would not be accurate when the objects undergo rotational motion or scale change even if an adaptive appearance model is utilized. We note that certain algorithms are constrained by their design and it may not be easy to use a different motion model to account for complex object movements. For example, the mean-shift based tracking algorithm[17] is not equipped to deal with scale change or in-plane rotation since the objective function is not differentiable with respect to these motion parameters. However, if the objective function of a tracking algorithm is not differentiable with respect to the motion parameters, it may be feasible to use either sampling or stochastic search to solve the optimization problem.

3.4 Dynamic Model

A dynamic model is usually utilized to reduce computational complexity in object tracking as it describes the likely state transition, i.e., $p(\mathbf{x}_t|\mathbf{x}_{t-1})$, between two consecutive frames where $\mathbf{x}_t$ is the state vector at time t. Constant velocity and constant acceleration models have been used in the early tracking methods such as Kalman filter-based trackers. In these methods, the state transition is modeled by a Gaussian distribution, $p\left(\mathbf{x}_t|\mathbf{x}_{t-1}\right) = \mathcal{N}\left(\Phi_{t-1}\mathbf{x}_{t-1},\, \xi_{t-1}\right)$, where Φ_{t-1} and ξ_{t-1} are the transfer matrix and noise at time $t-1$, respectively. Since the assumption of constant velocity or acceleration is rather constrained, most recent tracking algorithms adopt random walk models[1,7] with particle filters.

3.5 Search Mechanism

Since object tracking can be formulated as an optimization problem, the state search strategy depends mainly on the objective function form. In the literature, either deterministic or stochastic methods have been utilized for state search. If the objective function is differentiable with respect to the motion parameters, then gradient descent methods can be used.[16,17,33] Otherwise, either sampling[4,8] or stochastic methods[1,7] can be used. Deterministic methods based on gradient descent are usually computationally efficient, but suffer from the local minimum problems. Exhaustive search methods are able to achieve good tracking performance at the expense of very high computational load, and thus seldom used in tracking tasks. Sampling-based search methods can achieve good tracking performance when the state variables do not change drastically. Consequently, stochastic search algorithms such as particle filters are trade-offs between these two extremes, with the ability to escape from local minimum without high computational load. Particle filters have been widely used in recent online object tracking with demonstrated success.[1,7–9]

4. EXPERIMENTAL COMPARISON

In this section, we empirically compare tracking methods based on the above discussions and demonstrate how to choose and design effective algorithms. We evaluate 10 state-of-the-art tracking algorithms on 15 challenging sequences using different criteria. The test algorithms include: incremental visual tracker (IVT),[1] variance ratio tracker (VRT),[2] fragments-based tracker (FragT),[3] online boosting tracker (BoostT),[4] semi-supervised trackers (SemiT),[5] extended semi-supervised tracker (BeSemiT),[6] ℓ_1 tracker (L1T),[7] multiple instance learning tracker (MIL),[8] visual tracking decomposition algorithm (VTD),[9] and track-learning-detection method (TLD).[10] Based on the above analysis, we categorize these algorithms in Table 2 which describes their object representation, motion model, dynamic model, search mechanism and characteristics. The challenging factors of the test sequences are listed in Table 3. For fair evaluation, we use the the source codes provided by the authors in all experiments. For the tracking methods which use particle filtering (i.e., IVT, L1T, and VTD), we use 300 particles in all tests. The other parameters of each tracking method are carefully selected in each method for best performance. It is worth noting that the FragT method is not an online method although the experimental comparison shows the necessity of adaptive appearance models.

Table 2: Tracking algorithms. The entries denoted with "-" indicate no dynamic model is employed.

Algorithm	Motion Model	Object Representation	Dynamic Model	Searching Mechanism	Characteristics
IVT	affine transform	holistic gray-scale image vector	Gaussian	particle filter	generative
FragT	similarity transform	local gray-scale histograms	-	sampling	generative
VRT	translational motion	holistic color histograms	-	mean-shift	discriminative
BoostT	translational motion	holistic representation based on Haar-like, HOG and LBP descriptors	-	sampling	discriminative
SemiT	translational motion	holistic representation based on Haar-like descriptor	-	sampling	discriminative
BeSemiT	translational motion	holistic representation based on Haar-like, HOG, and color histograms	-	sampling	discriminative
L1T	affine transform	holistic gray-level image vector	Gaussian	particle filter	generative
MILT	translational motion	holistic representation based on Haar-like descriptor	-	sampling	discriminative
VTD	similarity transform	holistic representation based on hue, saturation, intensity, and edge template	Gaussian	particle filter	generative
TLD	similarity transform	holistic representation based on Haar-like descriptor	-	sampling	discriminative

Some of the tracking results are shown in Figure 1 and high resolution images as well as videos can be found on our web site (`http://faculty.ucmerced.edu/mhyang/pubs/spie11a.html`). We use two criteria, tracking success rate and location error with respect to object center, for quantitative evaluations. To compute the success rate, we employ the criterion used in the PASCAL VOC challenge[34] to evaluate whether each tracking result is a success or not. Given the tracked bounding box ROI_T and the ground truth bounding box ROI_G , the score is defined as

$$score = \frac{area(ROI_T \bigcap ROI_G)}{area(ROI_T \bigcup ROI_G)}. \quad (1)$$

The tracking result in one frame is considered as a success when this score is above 0.5 and the success rate is computed with all the frames. The center location error is defined as the distance between the central locations of the tracked target and the manually labeled ground truth. The success rates and average center location errors of all these trackers are listed in Table 4 and Table 5, respectively. Figure 2 shows the details of the tracking errors.

Table 3: The tracking sequences used in our experiments.

Sequences	Main challenging factors	Resolution	Number of frames
Sylvester	in-plane/out-of-plane pose change, fast motion, illumination change	320 × 240	1343
Wall-E	scale change, out-of-plane pose change	608 × 256	178
David-indoor	illumination variation, out-of-plane pose change, partial occlusion	320 × 240	461
surfing	fast motion, large scale change, small object, moving camera	320 × 240	870
singer	scale change, significant illumination change	624 × 352	350
shaking	in-plane pose change, significant illumination change	624 × 352	365
Gymnastic	deformation, out-of-plane pose change	426 × 234	765
jumping	image blur, fast motion	352 × 288	312
car	image blur, partial occlusion	320 × 240	280
faceocc	in-plane pose change, partial occlusion	320 × 240	812
PETS2009	heavy occlusion, out-of-plane pose change, distraction from similar objects	768 × 576	146
CAVIAR	heavy occlusion, distraction from similar objects	320 × 240	608
board	background clutter, out-of-plane pose change	640 × 480	698
Avatar	occlusion, out-of-plane pose change, illumination change	704 × 384	192
David-outdoor	low-contrast images, occlusion, out-of-plane pose change	320 × 240	251

Our experimental results show that the FragT[3] method performs well only in the *Sylvester* and *Gymnastics* sequences as it is able to deal with appearance variation due to pose change. While the FragT method is designed to handle partial occlusion, it is not equipped to deal with objects with in-plane rotation. The tracking results in the *faceocc* sequence show that it does not perform well when the object undergoes both in-plane rotation and partial occlusion simultaneously.

For the IVT method, it is designed to account for affine motion and appearance change (e.g., *Gymnastics*, *jumping*, and *faceocc* sequences). Since a holistic representation is used, it does not deal partial occlusion well. On the other hand, the use of Gaussian random walk model with particle filer make it robust to full occlusion to some degree since it is able to search around when the target object reappears in the scene after occlusion. However, as the IVT method uses all the tracking results for appearance update (though with a forgetting factor), it is prone to the effects of noisy observations and tracking errors are likely to accumulate. Therefore, the IVT method does not work well in long videos (e.g., *Sylvester* and *surfing*) and the sequences where the objects undergo large out-of-plane pose change (e.g., *Wall-E* and *David-outdoor*). As it is a generative method, the IVT method is also less effective in dealing with background clutter and low-contrast images (e.g., *board* and *Avatar* sequences).

The VRT method selects discriminative color features in each frame, and works better than the IVT method on the *Sylvester* and *board* sequences. However, since it does not deal with scale or pose change, the VRT method does not work well in cluttered background. As the VRT method uses holistic color histograms for representation, it is not effective in handling illumination change (e.g., *David-indoor*, *singer*, and *shaking*), low-contrast (e.g., *Avatar*), and background clutter (e.g., *PETS2009* and *Avatar*).

The BoostT method[4] uses Haar-like features, HOG and color histograms for representation, which are not so sensitive to image blur. As such, this tracker works well in the *jumping*, *car*, and *Sylvester* sequences. However, this method is sensitive to parameter setting and prone to drift when there are similar objects in the scenes (e.g., *PETS2009* and *CAVIAR*), or when the object undergoes large pose change in a cluttered background (e.g., *board*). As the BoostT method does not deal with scale change, it does not work well in the *Wall-E* and *singer* sequences. The experimental results with the SemiT method have less drifting errors than the BoostT method. In the *surfing* sequence, it is able to track the target object throughout the entire image sequence. However, when objects undergo fast appearance change (e.g., *Sylvester* and *David-outdoor*), this method does not work as well as the BoostT method. The BeSemiT can be regarded as a combination of the BoostT and SemiT methods. It works better than the above two boosting-based trackers in the *Gymnastics* sequence where the objects undergo pose change and deformation, in the *PETS2009* sequence where the targets are occluded along with similar objects, and in the *Avatar* sequence where the object is occluded as well as observed from different camera view angles.

The L1T method works well in the *singer* sequence where the target appearance can be reconstructed by a small number of the templates using ℓ_1-minimization even when there is drastic illumination change. It also works well in the *Wall-E* and

Table 4: Success rates (%).

	IVT	VRT	FragT	BoostT	SemiT	BeSemiT	L1T	MILT	VTD	TLD
Sylvester	45	72	77	78	36	46	36	68	79	86
Wall-E	11	8	8	8	11	11	51	8	20	70
David-indoor	57	2	44	24	26	28	35	41	72	61
surfing	45	37	14	37	100	95	39	5	33	43
singer	56	20	21	23	27	37	100	23	99	36
shaking	3	1	22	5	8	5	27	88	96	8
Gymnastics	79	88	85	20	16	72	7	47	71	56
jumping	99	51	29	96	96	63	99	99	88	99
car	50	9	18	91	88	88	29	95	96	95
faceocc	94	3	46	85	58	51	65	89	57	81
PETS2009	21	21	3	17	66	83	24	24	14	72
CAVIAR	12	13	12	11	14	11	39	12	10	11
board	21	77	44	24	13	5	9	44	32	13
Avatar	31	10	10	26	76	91	29	26	36	57
David-outdoor	12	2	16	44	12	30	42	30	40	28

Table 5: Average center location errors (in pixels). The entries denoted with "-" indicate the corresponding method fails from the beginning.

	IVT	VRT	FragT	BoostT	SemiT	BeSemiT	L1T	MILT	VTD	TLD
Sylvester	49	17	11	14	8	8	20	12	12	8
WallE2	58	21	34	53	44	-	14	28	18	30
David-indoor	19	114	58	32	20	32	45	30	18	13
surfing	43	9	35	6	3	5	37	51	36	34
singer	9	107	21	14	14	12	3	16	3	23
shaking	130	206	107	25	2	-	22	10	7	167
Gymnastics	10	10	8	15	7	12	123	16	9	12
jumping	5	70	33	10	10	19	7	6	10	6
car	57	42	70	5	10	2	72	4	5	7
faceocc2	18	66	44	22	23	5	33	16	53	9
PETS2009	72	64	154	737	59	79	43	66	73	85
CAVIAR	62	19	73	55	27	49	36	101	53	33
board	93	74	73	118	30	16	152	90	93	118
Avatar	54	160	104	14	50	122	49	48	53	69
David-outdoor	99	42	68	40	108	52	36	41	40	43

CAVIAR sequences, but is prone to drift in other videos (e.g., *surfing*, *Gymnastics*, and *car*). The experimental results can be explained by the use of rectangular templates for sparse representation as it is not equipped to deal with pose change, deformation, or full occlusion (e.g., *PETS2009* and *board*).

The MILT method utilizes multiple instance learning to reduce the visual drifts in updating appearance models. However, as the MILT method is not designed to handle large scale change, it does not perform well in the sequences where the targets undergo large scale changes (e.g., *Wall-E*, *singer*, and *surfing*). The VTD method uses multiple dynamic and observation models to account for appearance change. It works well in the *singer* and *shaking* sequences where there is significant illumination change, and in the *car* sequence where image blur and distractors appear in the scenes. The TLD method performs better than the other methods in the *Sylvester* and *Wall-E* sequences where the target objects undergo pose change. In the *CAVIAR*, *board* and *David-outdoor* sequences where distractors similar to the target objects appear in the cluttered background, none of these trackers work well. It is of great interest to design more discriminative appearance models to separate the target object from the cluttered background, and update method without accumulating tracking errors.

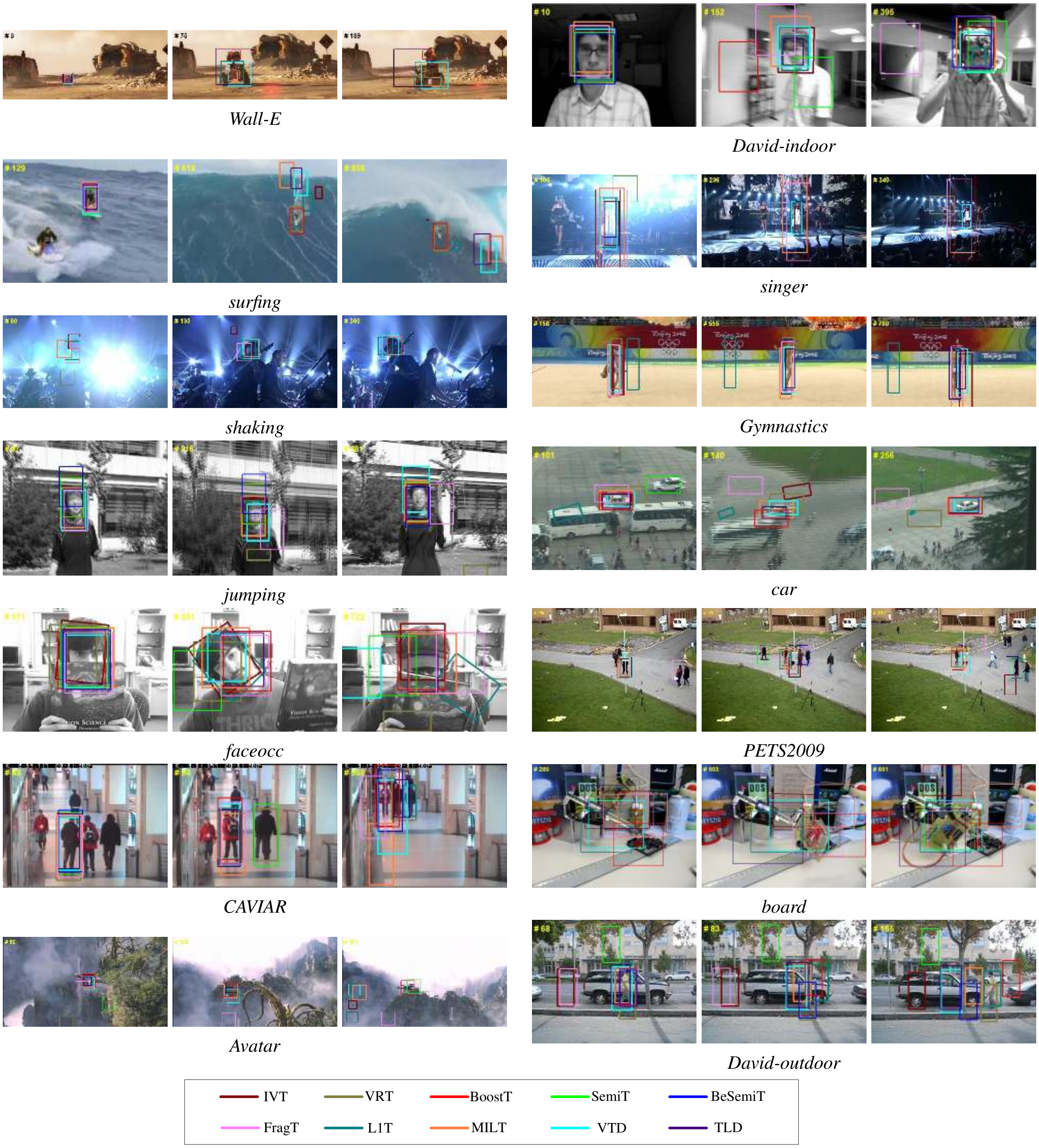

Figure 1: Tracking results on challenging sequences.

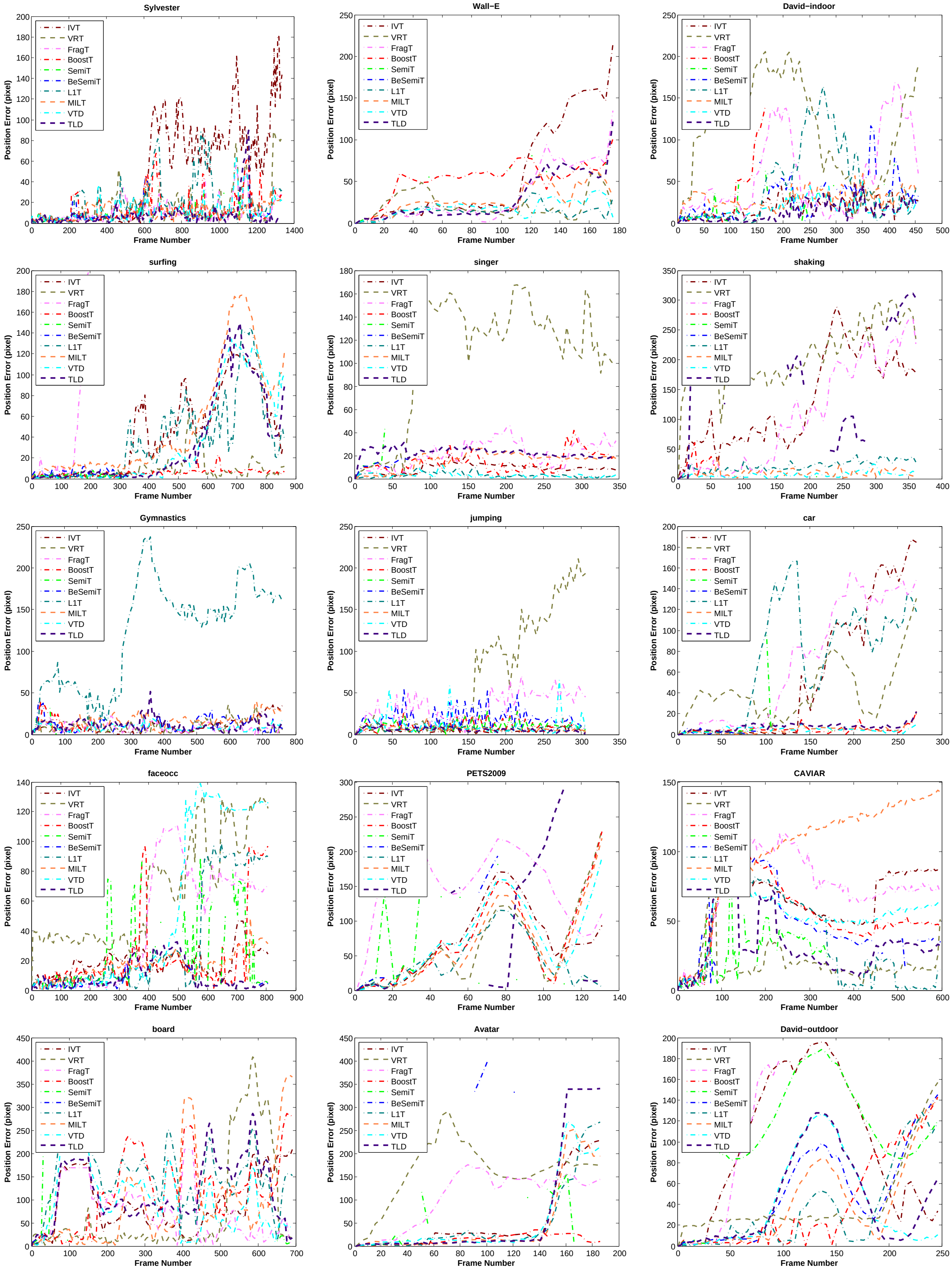

Figure 2: Error plots of all the test sequences.

5. CONCLUSION

In this paper, we review tracking methods in terms of their components and identify their roles in handling challenging factors of object tracking. We evaluate state-of-the-art online tracking algorithms with detailed analysis on their performance. The experimental comparisons demonstrate the strength as well as weakness of these tracking algorithms, and shed light on future research directions.

REFERENCES

[1] D. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *International Journal of Computer Vision* **77**(1-3), pp. 125–141, 2008.

[2] R. T. Collins, Y. Liu, and M. Leordeanu, "Online selection of discriminative tracking features," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **27**(10), pp. 1631–1643, 2005.

[3] A. Adam, E. Rivlin, and I. Shimshoni, "Robust fragments-based tracking using the integral histogram," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 798–805, 2006.

[4] H. Grabner and H. Bischof, "On-line boosting and vision," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 260–267, 2006.

[5] H. Grabner, C. Leistner, and H. Bischof, "Semi-supervised on-line boosting for robust tracking," in *Proceedings of European Conference on Computer Vision*, pp. 234–247, 2008.

[6] S. Stalder, H. Grabner, and L. Van Gool, "Beyond semi-supervised tracking: Tracking should be as simple as detection, but not simpler than recognition," in *Proceedings of IEEE Workshop on Online Learning for Computer Vision*, 2009.

[7] X. Mei and H. Ling, "Robust visual tracking using l1 minimization," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1436–1443, 2009.

[8] B. Babenko, M.-H. Yang, and S. Belongie, "Visual tracking with online multiple instance learning," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 983–990, 2009.

[9] J. Kwon and K. Lee, "Visual tracking decomposition," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1269–1276, 2010.

[10] Z. Kalal, J. Matas, and K. Mikolajczyk, "P-n learning: Bootstrapping binary classifiers by structural constraints," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 49–56, 2010.

[11] I. Haritaoglu, D. Harwood, and L. Davis, "W4s: A real-time system for detecting and tracking people," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 962–968, 1998.

[12] M. de La Gorce, N. Paragios, and D. Fleet, "Model-based hand tracking with texture, shading and self-occlusions," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[13] Z. Sun, G. Bebis, and R. Miller, "On-road vehicle detection: A review," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **28**(5), pp. 694–711, 2006.

[14] M. Kim, S. Kumar, V. Pavlovic, and H. Rowley, "Face tracking and recognition with visual constraints in real-world videos," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2008.

[15] X. Zhou, D. Comaniciu, and A. Gupta, "An information fusion framework for robust shape tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **27**(1), pp. 115–129, 2005.

[16] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of International Joint Conference on Artificial Intelligence*, pp. 674–679, 1981.

[17] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **25**(5), pp. 564–575, 2003.

[18] A. Azarbayejani and A. Pentland, "Recursive estimation of motion, structure, and focal length," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **17**(6), pp. 562–575, 1995.

[19] M. Isard and A. Blake, "Condensation – conditional density propagation for visual tracking," *International Journal of Computer Vision* **29**(1), pp. 5–28, 1998.

[20] M. Black and A. Jepson, "Eigentracking: Robust matching and tracking of articulated objects using a view-based representation," in *Proceedings of European Conference on Computer Vision*, pp. 329–342, 1996.

[21] A. D. Jepson, D. J. Fleet, and T. F. El-Maraghi, "Robust online appearance models for visual tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **25**(10), pp. 1296–1311, 2003.

[22] S. Avidan, “Support vector tracking,” *IEEE Transactions on Pattern Analysis and Machine Intelligence* **26**(8), pp. 1064–1072, 2004.

[23] S. Avidan, “Ensemble tracking,” *IEEE Transactions on Pattern Analysis and Machine Intelligence* **29**(2), pp. 261–271, 2007.

[24] D. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision* **60**(2), pp. 91–110, 2004.

[25] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[26] T. Ojala, M. Pietikainen, and T. Maenpaa, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**(7), pp. 971–987, 2002.

[27] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet, “Color-based probabilistic tracking,” in *Proceedings of European Conference on Computer Vision*, pp. 661–675, 2002.

[28] R. T. Collins, “Mean-shift blob tracking through scale space,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 234–240, 2003.

[29] S. T. Birchfield and S. Rangarajan, “Spatiograms versus histograms for region-based tracking,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, **2**, pp. 1158–1163, 20–25 June 2005.

[30] S. Baker and I. Matthews, “Lucas-Kanade 20 years on: A unifying framework,” *International Journal of Computer Vision* **56**(3), pp. 221–255, 2004.

[31] X. Li, W. Hu, Z. Zhang, X. Zhang, and G. Luo, “Robust visual tracking based on incremental tensor subspace learning,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2007.

[32] F. Porikli, O. Tuzel, and P. Meer, “Covariance tracking using model update based on lie algebra,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 728–735, 2006.

[33] L. Matthews, T. Ishikawa, and S. Baker, “The template update problem,” *IEEE Transactions on Pattern Analysis and Machine Intelligence* **26**(6), pp. 810–815, 2004.

[34] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (voc) challenge,” *International Journal of Computer Vision* **88**(2), pp. 303–338, 2010.