Scale and Occlusion Invariant Tracking-by-Detection

Andrea Mazzeschi, Giuseppe Lisanti(✉), Federico Pernici, and Alberto Del Bimbo

Media Integration and Communication Center (MICC), University of Florence, Viale Morgagni 65, 50134 Firenze, Italy
andrea.mazzeschi@gmail.com,
{giuseppe.lisanti,federico.pernici,alberto.delbimbo}@unifi.it

Abstract. In this paper we present a solution for tracking-by-detection that is able to handle both scale variations and occlusions of the tracked object. We build upon the framework proposed in [7] based on structured output SVM and improve it in order to deal with both variations of target scale and occlusions. We first propose to modify the original solution to include the scale variations both in the patch sampling stage and in the structured output state. Then in order to deal with occlusions we introduce an incremental classifier to discriminate the target from the context. This classifier combines a learning phase with a unlearning one that help to avoid drift in the model of the tracked object. The proposed solution outperforms the method in [7] for sequences that present scale variations or occlusions while maintaining comparable performance on those sequences with none of these issues. Moreover, we outperform other state-of-the-art solutions on publicly available sequences commonly used in literature.

1 Introduction

Tracking is a fundamental problem in computer vision. Several aspects of this difficult task have been considered in literature. Generally speaking, difficulties arise depending on the type of information that needs to be tracked: 3D pose, imaged 2D location, imaged 2D shape, 3D shape, imaged 2D articulated body shape, 3D articulated body shape, etc. Besides dealing with the inherent difficulties related to the specific information of interest, effective methods must also provide robust object representation coping with nuisance factors that affect the image formation process. For example objects may have non-rigid shape or may be made of translucent or reflective materials and camera sensors may suffer from the effects of noise, sensor quantization and motion blur. In addition to these intrinsic problems, practical requirements such as: 1) long-term tracking; and 2) object reacquisition after partial or total occlusion, may prevent correct tracking. In some applications, the object to be tracked is known in advance and it is possible to incorporate specific prior knowledge when designing the tracker to alleviate some of these issues. However, the general case of tracking

V. Murino and E. Puppo (Eds.): ICIAP 2015, Part II, LNCS 9280, pp. 575–585, 2015.
DOI: 10.1007/978-3-319-23234-8_53

arbitrary objects by simply specifying a single (one-shot) training example at runtime, is a challenging open problem which deserves particular attention. In this scenario, the tracker must be able to model the appearance of the object on-the-fly by generating and labeling image features and learning the model of the object appearance.

In this paper we propose to exploit structured output SVM, extending the work proposed in [7], in order to be able to deal with some classical nuisance factors. In particular, we introduce scale sampling in the prediction of the target state in order to be able to manage target scale variations. Then we introduce an incremental classifier that act as validator of the structured output SVM in order to handle occlusions and out of view of the scene. Experimental results show that the proposed scale and occlusion handling allows to improve performance while preserving the adaptability of [7].

1.1 Related Work

A number of methods have been developed in which tracking is considered as 2D image bounding box localization, each one dealing with different nuisance factors.

However, not all nuisance factors are equal; a distinction should be made in order to better understand the problem. When we are facing occlusions, we are dealing with presence or absence of a signal while in the case of illumination and pose variations the signal is changing but still remains strongly correlated. The former nuisance factors are not invertible and do not admit invariant representations while the latter will. The latter case is generally well captured by features like HoG, Haar or LBP while the former is much more complex and cannot be explicitly modeled through the feature representation [15] invariance. MILTrack [2], for example, adopting bag of image patches can cope with misalignments and occlusions by adding novel examples as new instances for the object representation.

Recently, three methods have received a lot of attention for their positive performance and for their algorithmic design and image representation peculiarity [2,8,12]. They mainly differ on how they consider the *template update problem* which primarily impacts on the drift of the tracker [11]. Babenko et al. [2] address the problem by building an evolving boosting classifier that tracks bags of image patches. Kalal et al. [8] combine a optic flow tracker with a online random forest. This solution has been succesively extended in [9] with the TLD-Predator tracking framework where the tracking task is decomposed into tracking, learning and detection and a P-N learning method is exploited. In Mei and Ling [12] the tracking problem is formulated as finding a sparse representation of the candidate object, combining trivial templates which are primarily responsible for the presence or the absence of certain object regions.

Our work as many others [2,4,6,8,10,17] makes use of context information to extract the features of the background surrounding the target. Features are then used to improve the distinction of the target against its background, either by feature selection or by training classifiers as in [1,3]. In [18] the CT-Tracking

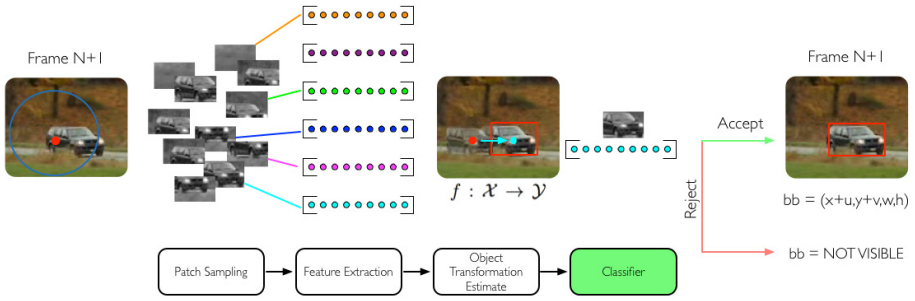

Fig. 1. Scheme of the proposed solution. The classifier (green box) validates the prediction of the structured output SVM and controls the target position change and the model update.

exploits an appearance model based on features extracted from foreground and background at multiple scales of the image and employs non-adaptive random projections to preserve the structure of the image feature space. A critical issue here is in the accuracy between foreground and background image regions. Generally they are divided by the bounding box of the object; such a partition is too rough and it could happen that background regions are treated as part of the foreground. This typically causes a gradual degradation in object appearance representation which results in template drift.

2 Scale and Occlusion Invariant Tracking-By-Detection

Hare et al. proposed a novel tracking approach [7] that directly estimates the object transformation between frames rather than performing a detection. In this solution structured output SVM has been used to learn a prediction function $f : \mathcal{X} \rightarrow \mathcal{Y}$, which maps the target features into the space of the in-plane translations:

$$\mathcal{Y} = \{(u, v) \mid u^2 + v^2 \leq r_s^2\} \tag{1}$$

where (u, v) are respectively the x and y translation components.

This solution, however, suffers from some limitations. Firstly, the tracker cannot handle target scale and this mostly affects the tracking quality. Moreover, no occlusions and out-of-field detection mechanisms are present and this can introduce erroneous data into the appearance model, compromising the long term performance.

To overcome these limitations we propose to slightly modify the original formulation of [7] in order to be able to handle target scale variations. Moreover, we introduce a detector that is able to overcome occlusion and out-of-view during the tracking. A scheme of the whole approach is shown in Fig. 1.

2.1 Scale Invariance

In order to manage the target scale variation it is necessary to extend the output space of the prediction function in Eq. 1 as follows:

$$\tilde{\mathcal{Y}} = \{(u, v, s) \mid u^2 + v^2 \leq r_s^2; s \in \{s_{min} \dots s_{max}\}\} \tag{2}$$

where s represents the percentage variation of the target's bounding box dimension compared to the previous frame. The values s_{min} and s_{max} are the smallest and the greatest change allowed between two consecutive frames.[1]

A feature vector of 4288 elements, obtained as concatenation of HOG (4096) and Haar-like (192) features, is adopted as patch descriptor.

2.2 Occlusion and Out-of-View Handling

An occlusion and out-of-view detection mechanism is crucial in order to be able to manage complex tracking situations. Our main idea is to introduce a classifier to discriminate the target from the context [5,14]. This classifier acts as a validator of the structured output SVM prediction by accepting or rejecting the target model update and therefore the target position update, see Fig. 1.

The classifier initialization phase lasts for the first K frames of the sequence. During this period the classifier is not considered and it is assumed that the target is completely visible in the scene. Positive and negative examples are collected at every frame. The positive examples are chosen as warped versions of the region of interest (roi) of the target position. The negative samples are chosen, instead, from eight patches around the current target location.

The classifier is incrementally trained during the sequence. In particular, at every frame a prediction of the position of the target is performed, following Eq. 2, and the classifier evaluates if the predicted region contains the tracked target or not. When the classifier is not able to identify the target in any of the τ_c consecutive frames, the tracking procedure is interrupted and the reacquisition phase starts. During the reacquisition, a whole image target search is accomplished and neither the structured output SVM model nor the classifier model are updated, in order to prevent degradation of the target model.

A huge number of examples and their variability can corrupt the classifier decision boundary. To avoid this problem, similarly to [5], we integrated an unlearning phase during which a defined percentage of training examples, randomly picked every M frames, is removed. In [5] this was done following a temporal window mechanism that keeps only the latest examples. We argue that this choice produces a classifier which focuses only on the latest target appearances. This may limit the reacquisition ability for all of these cases where the target appearance differs from the latest seen. For this reason we perform a uniform random sampling in order to prevent the data distribution corruption.

[1] We chose to handle only fixed aspect ratio scale changes mainly due to computational limits.

In order to decide if the prediction is correct or not we need to report the classifier output in a probabilistic form following [13], such as:

$$p(\mathbf{T} \mid f(\mathbf{x})) = \frac{1}{1 + \exp(Af(\mathbf{x}) + B)} \tag{3}$$

where $f(\mathbf{x})$ is the SVM output value and A, B are the sigmoid parameters. Eq. 3 gives the probability that $\mathbf{x}$ is an instance of the target ($\mathbf{T}$) given the SVM output $f(\mathbf{x})$. The sigmoid parameters (A and B) are updated during the tracking procedure every M frames as described in [13] in order to be able to adapt the tracker to the visual changes.

The target prediction will be classified as correct if the 80% of the positive examples of the training set are correctly classified:

$$\frac{\sum_{\mathbf{x}_i s.t. y_i = 1} \mathbb{1}(\frac{1}{1 + \exp(Af(\mathbf{x}) + B)} \geq \theta)}{N^+} \geq 0.8 \tag{4}$$

where $\mathbb{1}(d)$ returns 1 if the inequality d is satisfied and 0 otherwise, N^+ is the amount of positive examples in the training set and θ is a threshold automatically estimated every M frames from the observed samples (in order to adapt the learned model to the new observed examples).

3 Experiments

We performed a series of experiments to show the effectiveness of the proposed solution. In particular, we first show how managing scale variations allows to improve the performance of [7] (we refer to this method as Struck); then we show the effectiveness of the occlusion handling mechanism; finally we report a comparison against some of the state-of-the-art tracking-by detection methods [2,7,9,18].

Tests were conducted on 21 public available sequences from the dataset in [16].

3.1 Parameters

In our experiments we set the scale sampling values s_{min} and s_{max} to 0.8 and 1.2 respectively. We found that these values are a good tradeoff between the computational burden and the fact that in the reality the target scale does not change abruptly between consecutive frames. Before extracting HOG and Haar-like features, every patch is preliminary resized to a fixed resolution of 32×96 pixels.

As regards the classifier for occlusion detection we set the initialization to $K = 8$ frames in order to be able to collect sufficient information about the target appearance. For the unlearning phase we decided to remove the 20% of training examples randomly picked every $M = 70$ frames. All these parameters were set accordingly to [5,13] and considering the validation reported in Fig. 2.

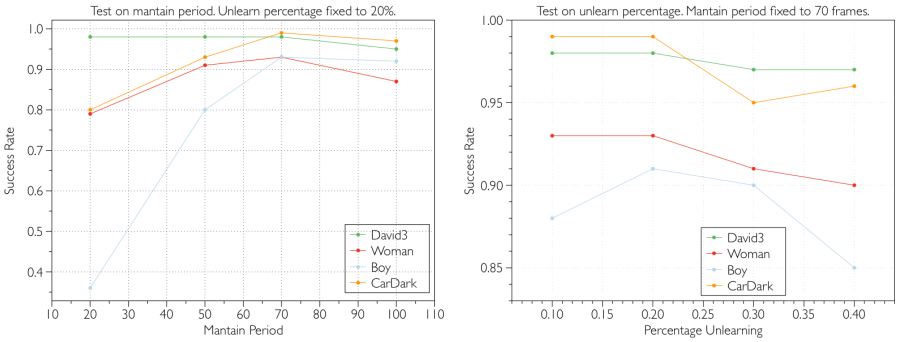

Fig. 2. *Left* The success rate variation according to the maintain period. *Right* The success rate variation according to the unlearn percentage.

3.2 Scale Handling Test

With these experiments we evaluate the performance of the scale handling as described in 2.1. For this test we chose four different video sequences in which the scale changes is the main nuisance factor. During these tests the classifier component for occlusion handling was not ran.

In Table 1 we report this comparison in terms of Success Rate and Average Overlap. Clearly handling scale variations increases both the tracking ability (higher success rate) and the tracking quality (higher avg. overlap).

Table 1. Success Rate and Average Overlap comparison for the Struck with and without our scale handling on four test sequences. Best results in bold.

		Success Rate		Avg. Overlap	
Sequence	Frames	Struck	Struck+Scale	Struck	Struck+Scale
CarScale	252	0.34	**0.63**	0.36	**0.52**
Jogging	307	0.25	**0.95**	0.16	**0.63**
Singer1	351	0.30	**0.99**	0.36	**0.62**
Walking2	500	0.42	**0.99**	0.50	**0.82**
Mean	-	0.32	**0.89**	0.34	**0.65**

In Fig. 3 we also show the percentage of frame associated to the overlap score, for the Jogging and Walking2 sequences. It is possible to appreciate that our solution with scale handling obtains an overlapping score between 0.5 and 0.7 for the Jogging sequence and between 0.7 and 0.9 for the Walking2 sequence, for a large number of frames. On the contrary, Struck is not able to track the target for a large number of frames in the Jogging sequence, while for the Walking2 sequence it presents a decreasing overlapping score.

3.3 Occlusion Handling Test

To evaluate the re-acquisition ability of the classifier component we used synthetic sequences to overcome the problem of the limited number of video sequences where out-of-field cases occur. In particular, we used 5 sequences and split them in clips of 50 frames each. Then, for each clip we replaced the 30 central frames with a synthetic image obtained by removing the target from the original scene in order to simulate an out-of-field scenario. A total of 31 cases have been analysed. In Table 2 we report the number of splits generated from the sequence, the number of cases in which the target is correctly tracked after the synthetic occlusion, the number of cases in which the target is missed after the synthetic occlusion and the number of cases in which the target is wrongly tracked during the occlusion. We assume that the target is correctly tracked if the Pascal Overlap Score is greater than 0.5.

Table 2. Reacquisition performance on synthesized sequences.

Sequence	# Splits	# Tracked	# Missed	# F. Alarm
Clutter	10	8	2	0
Couple	3	3	0	0
Jumping	5	3	1	1
Subway	3	2	1	0
Sylvester	10	4	6	0
Total	31	20	10	1

It can be observed that most of our *Missed* cases come from the Sylvester sequence. This is mainly due to the change in appearance that occurs when the target re-enter in the scene (after the synthetic out-of-field) in an extremely different pose with respect to the latest one observed by the tracker.

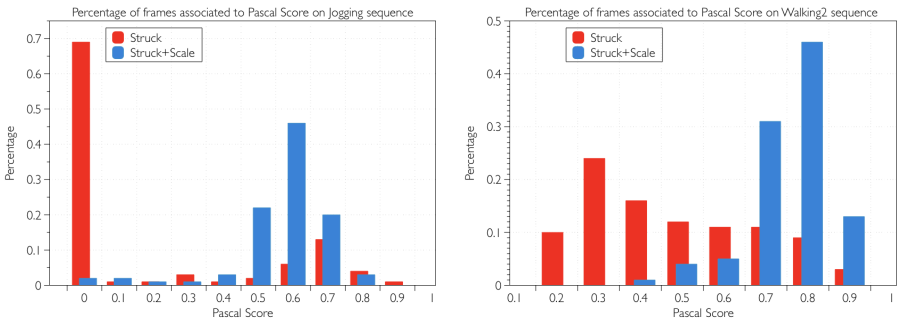

Fig. 3. Percentage of frames associated per Pascal Score for the sequences Jogging (*Left*) and Walking2 (*Right*) for Struck without (red) and with scale handling (blue).

3.4 Final Test

In this section we report the experiments performed with our full solution over various tracking conditions. Illumination changes, occlusions, out of plane rotations and scale changes are some of the issues present in the 21 sequences that we chose from the dataset in [16].

Table 3. Success Rate/Average Location error(px) - Bold numbers indicate the best score, underlined numbers indicate the second best.

Sequence	Frames	Struck	Our	CT	TLD	MIL
Boy	602	**0.98**/**3.21**	0.93/6.77	0.68/8.75	0.93/7.93	0.38/13.15
Car4	659	0.40/25.58	0.70/**17.28**	0.27/87.03	**0.78**/19.90	0.27/40.21
CarDark	393	**1**/**2.21**	0.99/3.54	0.01/119.10	0.52/28.32	0.18/42.88
CarScale	252	0.34/70.90	**0.72**/24.95	0.45/63.68	0.43/28.52	0.45/**11.26**
Couple	140	0.63/32.51	0.83/8.12	0.68/32.79	**1**/**4.82**	0.67/33.80
Crossing	120	0.86/3.25	**1**/**1.80**	0.98/3.33	0.52/22.70	0.98/2.30
David2	537	**1**/**1.32**	0.77/4.70	0.01/76.84	0.95/5.86	0.32/11.01
Deer	71	**1**/8.71	0.90/8.55	0.04/243.10	0.73/**3.16**	0.12/94.83
Dudek	1145	**0.94**/18.70	0.86/**16.10**	0.85/20.17	0.84/22.20	0.85/24.70
Fish	476	**1**/**3.70**	**1**/3.71	0.89/10.91	0.96/10.60	0.38/26.66
FleetFace	707	**0.69**/62.74	0.64/**38.34**	0.63/67.47	0.56/55.56	0.53/71.63
Freeman1	326	0.20/13.60	**0.64**/**11.74**	0.10/121.45	0.21/42.22	0.15/19.22
Jogging	307	0.21/89.49	**0.97**/**3.28**	0.22/93.22	**0.97**/5.72	0.22/93.03
Lemming	1336	0.64/33.86	**0.83**/**7.08**	0.68/32.58	0.59/19.95	0.81/13.59
MountainBike	228	**0.98**/**6.60**	**0.98**/6.72	0.17/216.62	0.26/96.38	0.57/74.39
Singer1	351	0.30/79.19	**1**/**8.81**	0.24/74.39	0.99/17.91	0.27/76.54
Subway	175	0.38/108.29	0.76/**4.92**	0.7/10.86	0.23/50.43	**0.79**/7.62
Suv	945	0.73/28.94	**0.90**/**4.13**	0.22/62.91	0.83/7.76	0.13/72.04
Sylvester	1345	0.80/14.88	0.81/**3.94**	0.83/10.98	**0.93**/11.97	0.54/16.13
Walking	412	0.55/8.18	**0.99**/**4.05**	0.50/18.75	0.38/7.16	0.54/13.69
Walking2	500	0.42/7.45	**0.97**/**2.87**	0.38/50.32	0.34/27.43	0.38/50.10
Mean	-	0.67/29.68	**0.86**/**9.12**	0.45/68.30	0.67/23.65	0.46/38.51

We compare our solution against Struck, CT-Tracking (CT), TLD-Predator (TLD) and MILTrack (MIL). In particular, Struck results are obtained using the original source code provided by the authors while for the other methods we used the public available results[2]. Performance are expressed in terms of Success Rate, Average Center Location Error and Average Overlap and are reported respectively in Table 3 and Table 4.

Experiments show how the proposed solution obtains state-of-the-art results in almost every sequence under test. In terms of Success Rate the average increase is about 20%. It's also worth to note that in all the sequences where Struck achieves best results, our solution produces similar performances. This fact

[2] http://cvlab.hanyang.ac.kr/tracker_benchmark/v1.0/tracker_benchmark_v1.0_results.zip

Table 4. Average Overlap - Bold numbers indicate the best score, underlined numbers indicate the second best.

Sequence	Struck	Our	CT	TLD	MIL
Boy	**0.77±0.10**	0.66±0.11	0.59±0.18	0.66±0.09	0.49±0.21
Car4	0.49±0.19	0.55±0.20	0.21±0.30	**0.63±0.22**	0.26±0.31
CarDark	**0.80±0.07**	0.76±0.10	0.00±0.05	0.44±0.35	0.19±0.25
CarScale	0.36±0.28	**0.59±0.25**	0.43±0.31	0.42±0.24	0.40±0.31
Couple	0.48±0.35	0.65±0.20	0.46±0.32	**0.77±0.08**	0.49±0.34
Crossing	0.63±0.11	**0.76±0.06**	0.68±0.09	0.40±0.35	0.72±0.11
David2	**0.86±0.04**	0.66±0.13	0.00±0.04	0.69±0.12	0.45±0.21
Deer	**0.73±0.07**	0.67±0.10	0.03±0.18	0.59±0.36	0.12±0.24
Dudek	**0.72±0.14**	0.67±0.22	0.64±0.13	0.64±0.15	0.70±0.15
Fish	**0.87±0.06**	0.83±0.05	0.71±0.14	0.80±0.14	0.45±0.19
FleetFace	0.57±0.25	**0.58±0.23**	0.52±0.23	0.48±0.25	0.49±0.23
Freeman1	0.38±0.18	**0.52±0.23**	0.14±0.19	0.27±0.28	0.34±0.18
Jogging	0.16±0.31	**0.80±0.15**	0.17±0.32	0.76±0.14	0.18±0.33
Lemming	0.49±0.31	**0.65±0.29**	0.54±0.26	0.53±0.22	0.64±0.18
MountainBike	0.67±0.10	**0.69±0.07**	0.14±0.30	0.19±0.32	0.45±0.30
Singer1	0.36±0.25	**0.79±0.08**	0.34±0.24	0.72±0.08	0.35±0.25
Subway	0.28±0.34	**0.59±0.14**	0.57±0.10	0.18±0.33	0.64±0.14
Suv	0.62±0.38	**0.71±0.24**	0.23±0.27	0.67±0.24	0.20±0.26
Sylvester	0.63±0.27	0.62±0.30	**0.66±0.16**	0.67±0.16	0.52±0.23
Walking	0.56±0.16	**0.70±0.09**	0.52±0.13	0.44±0.21	0.54±0.15
Walking2	0.50±0.19	**0.77±0.14**	0.26±0.29	0.29±0.34	0.28±0.34
Mean	0.57	**0.70**	0.37	0.53	0.42

underlines how the introduction of the classifier does not compromise Struck's adaptive ability. A similar observation can be also made for the Average Center Location Error results and the Average Overlap, respectively in Table 3 and Table 4.

In Fig. 4 we report some sample frames for four different sequences and in comparison with [2,7,9,18]. Compared to other methods only TLD can reach similar results in terms of precision. For the Jogging sequence at the 77-th frame the tracking procedure is interrupted by the classifier due to an occlusion. In Lemming, again, the classifier stops the tracking due to a strong out-of-plane rotation. In both cases our method is able to correctly reacquire the target after some frames.

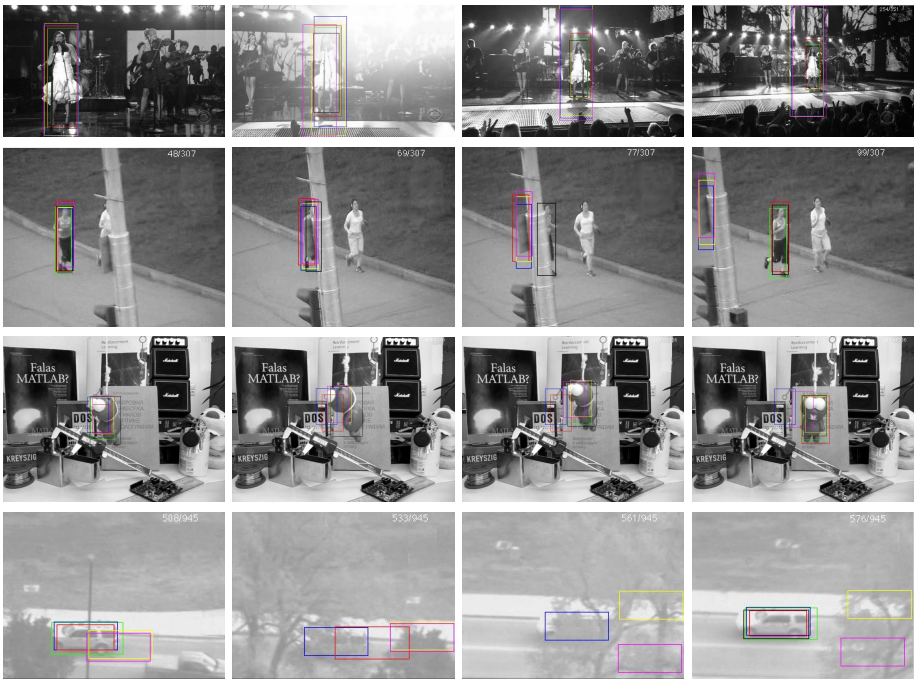

Fig. 4. Tracking results sample frames. From top to bottom: Singer1, Jogging, Lemming and Suv sequences. In each frame the results color are: Our, Struck, TLD, MIL, CT, in black the ground-truth (GT). In Singer1 the scale variation handling of our method is highlighted.

4 Conclusion

In this paper we have proposed a tracking-by-detection solution, starting from [7], that is able to deal with both variations of target scale, occlusions and out-of-view. We have shown how including scale information during the tracking allows us to achieve better performance compared to [7]. After that we have introduced a classifier to discriminate the target from the context. The proposed solution outperforms the method in [7] for those sequences that present scale variations or occlusions while it maintains comparable performance on those sequences with none of these issues.

References

1. Avidan, S.: Ensemble tracking. IEEE Trans. Pattern Anal. Mach. Intell. **29**, 261–271 (2007)
2. Babenko, B., Yang, M.H., Belongie, S.: Robust object tracking with online multiple instance learning. Pattern Analysis and Machine Intelligence **33**(8), 1619–1632 (2011)

3. Collins, R., Liu, Y.: On-line selection of discriminative tracking features. In: Proc. of ICCV (2003)
4. Dinh, T.B., Vo, N., Medioni, G.: Context tracker: Exploring supporters and distracters in unconstrained environments. In: Proc. of CVPR (2011)
5. Dinh, T.B., Yu, Q., Medioni, G.: Co-trained generative and discriminative trackers with cascade particle filter. Computer Vision and Image Understanding **119**, 41–56 (2014)
6. Gu, S., Zheng, Y., Tomasi, C.: Efficient Visual Object Tracking with Online Nearest Neighbor Classifier. In: Kimmel, R., Klette, R., Sugimoto, A. (eds.) ACCV 2010, Part I. LNCS, vol. 6492, pp. 271–282. Springer, Heidelberg (2011)
7. Hare, S., Saffari, A., Torr, P.H.: Struck: Structured output tracking with kernels. In: Proc. of ICCV (2011)
8. Kalal, Z., Matas, J., Mikolajczyk, K.: P-n learning: Bootstrapping binary classifiers by structural constraints. In: Proc. of CVPR (2010)
9. Kalal, Z., Mikolajczyk, K., Matas, J.: Tracking-learning-detection. Pattern Analysis and Machine Intelligence **34**(7), 1409–1422 (2012)
10. Li, H., Shen, C., Shi, Q.: Real-time visual tracking using compressive sensing. In: Proc. of CVPR (2011)
11. Matthews, I., Ishikawa, T., Baker, S.: The template update problem. In: Proc. of BMVC (2003)
12. Mei, X., Ling, H.: Robust visual tracking using 1 minimization. In: ICCV 2009, pp. 1436–1443 (2009)
13. Platt, J., et al.: Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. Advances in Large Margin Classifiers **10**(3), 61–74 (1999)
14. Santner, J., Leistner, C., Saffari, A., Pock, T., Bischof, H.: Prost: Parallel robust online simple tracking. In: Proc. of CVPR (2010)
15. Soatto, S.: Actionable information in vision. In: Proc. of ICCV (2009)
16. Wu, Y., Lim, J., Yang, M.H.: Online object tracking: A benchmark. In: Proc. of CPVR (2013)
17. Yu, Q., Dinh, T.B., Medioni, G.G.: Online Tracking and Reacquisition Using Co-trained Generative and Discriminative Trackers. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 678–691. Springer, Heidelberg (2008)
18. Zhang, K., Zhang, L., Yang, M.-H.: Real-Time Compressive Tracking. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part III. LNCS, vol. 7574, pp. 864–877. Springer, Heidelberg (2012)