Clustering Local Motion Estimates for Robust and Efficient Object Tracking

Mario Edoardo Maresca and Alfredo Petrosino(✉)

Department of Science and Technology, University of Naples Parthenope, Naples, Italy
mariomaresca@hotmail.it, alfredo.petrosino@uniparthenope.it

Abstract. We present a new short-term tracking algorithm called Best Displacement Flow (BDF). This approach is based on the idea of 'Flock of Trackers' with two main contributions. The first contribution is the adoption of an efficient clustering approach to identify what we term the 'Best Displacement' vector, used to update the object's bounding box. This clustering procedure is more robust than the median filter to high percentage of outliers. The second contribution is a procedure that we term 'Consensus-Based Reinitialization' used to reinitialize trackers that have previously been classified as outliers. For this reason we define a new tracker state called 'transition' used to sample new trackers in according to the current inlier trackers.

Keywords: Visual object tracking · Optical flow · Motion-based · Texture-less tracking

1 Introduction

The main challenge of an object tracking system is the difficulty to handle the appearance changes of the target object. The appearance changes can be caused by intrinsic changes such as pose, scale and shape variation and by extrinsic changes such as illumination, camera motion, camera viewpoint, and occlusions.

For instance, our approach Matrioska [13], while ranking closely to one of the best performing tracker EDFT [4] (see the Accuracy-Robustness plot shown in Figure 1 for the trackers that joined the VOT2013 challenge [10]), was not able to rank better due to failures on some sequences. Indeeed, as Figure 2 shows, Matrioska fails on sequences such as *hand* and *torus* mainly due to two factors: (i) texture-less objects and (ii) non-rigid transformations, resulting in low values for the Accuracy and Robustness, as reported in Table 1.

To model such variability, various approaches have been proposed, such as: updating a low dimensional subspace representation [15], MIL based [1] and template or patch based. Other approaches are reported in recent surveys ([10], [23] and [18]), and specifically [2–7,11,12,14,16,17,19,20,22,24–26].

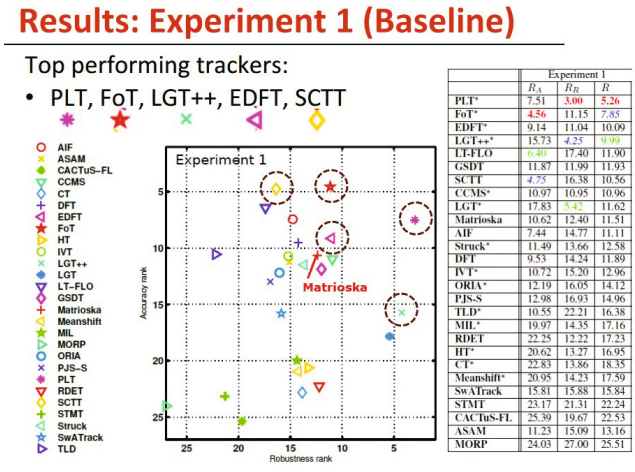
In this paper we introduce a new short-term tracking algorithm named *Best Displacement Flow* (BDF), that is aimed to avoid the Matrioska's failure cases.

L. Agapito et al. (Eds.): ECCV 2014 Workshops, Part II, LNCS 8926, pp. 244–253, 2015.
DOI: 10.1007/978-3-319-16181-5_17

To achieve a better robustness over texture-less objects we adopt a different visual representation: Matrioska is based on a sparse representation with the use of point features, whereas BDF adopts a dense approach represented by local trackers that cover the entire object.

BDF is inspired by the Flock of Features ([9], [20]) where a set of displacements, estimated by local trackers, are robustly combined to localize the target object. We propose different contributions and we show how this approach reaches state-of-the-art performance for sequences in which a re-detector module is not required.

The main contributions, i.e. the clustering procedure and the consensus-based reinitialization, are discussed in sections 2.2 and 2.3, respectively.



	Experiment 1		
	R_A	R_R	R
PLT*	7.51	3.00	5.26
FoT*	4.56	11.15	7.85
EDFT*	9.14	11.04	10.09
LGT++*	15.73	4.25	9.99
LT-FLO	6.40	17.40	11.90
GSDT	11.87	11.99	11.93
SCTT	4.75	16.38	10.56
CCMS*	10.97	10.95	10.96
LGT*	17.83	5.42	11.62
Matrioska	10.62	12.40	11.51
AIF	7.44	14.77	11.11
Struck*	11.49	13.66	12.58
DFT	9.53	14.24	11.89
IVT*	10.72	15.20	12.96
ORIA*	12.19	16.05	14.12
PJS-S	12.98	16.93	14.96
TLD*	10.55	22.21	16.38
MIL*	19.97	14.35	17.16
RDET	22.25	12.22	17.23
HT*	20.62	13.27	16.95
CT*	22.83	13.86	18.35
Meanshift*	20.95	14.23	17.59
SwATrack	15.81	15.88	15.84
STMT	23.17	21.31	22.24
CACTuS-FL	25.39	19.67	22.53
ASAM	11.23	15.09	13.16
MORP	24.03	27.00	25.51

Fig. 1. The Accuracy-Robustness plot of VOT2013 challenge

Fig. 2. Snapshots of the *hand* and *torus* sequences showing typical Matrioska failure cases: texture-less objects and non-rigid transformations

Table 1. Matrioska's results on hand, torus and diving sequences of the VOT2013 dataset

	accuracy	**robustness**	**speed (fps)**
hand	0.37	7.00	24.81
torus	0.26	8.00	16.25
diving	0.32	4.00	14.00
iceskater	0.48	4.00	11.49

2 Best Displacement Flow (BDF)

In the following sections we will describe our tracking approach for short-term sequences. A short-term tracker is an algorithm able to track an unknown object for short sequences in which the target object is visible through the entire sequence, and it usually does not have a re-detector module (if the object goes out of the scene the tracker will drift).

Our approach called Best Displacement Flow is inspired by (in order of publication) Flock of Features [9], Median Flow [8] and Flock of Trackers ([20], [21]) where a set of displacements, estimated by local trackers, are robustly combined to localize the target object. The name of our approach, BDF, remarks the most important difference between our tracker and the other approaches: we apply a clustering procedure over all local trackers estimates to filter outliers instead of using the median filter. The biggest cluster identifies what we term the *best displacement* vector used to update the position of the target bounding box.

The following sections describe in detail the main components of our system: the multi-size initialization (section 2.1), the clustering procedure (section 2.2) and the consensus-based reinitialization (section 2.3).

2.1 Multi-size Initialization

The initialization is the first step of our approach. Unlike other approaches, which use the same patch size for each tracker (both MedianFlow [8] and Flock of Trackers [20] use a single grid with a fixed cell size), we allow the initialization of local trackers with different patch sizes, as Figure 3 shows. To estimate the optical flow we use the Block Matching algorithm, i.e. each patch is used as a template to find the displacement that optimizes a cost function in the following frame.

For this reason the patch size becomes an important factor, hence the use of patches with different sizes ensures a greater robustness. Note that we do not constraint the trackers position inside a cell (i.e. the local trackers can freely move inside the object bounding box).

2.2 Displacement Clustering

Each local tracker, after the initialization in the first frame, estimates the displacement that optimizes a cost function (usually the SSD or the NCC) using the block matching algorithm for the optical flow estimation.

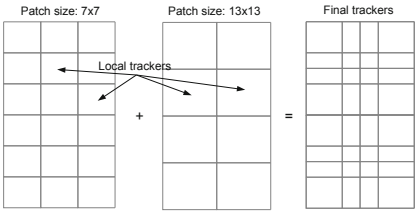


Fig. 3. The final trackers are obtained by superimposing grids with different patch sizes. In this case two grids with a size of 7x7 and 13x13 pixel. Using patches with different sizes ensure a greater robustness over different appearances of objects.

Once every tracker estimated its displacement vector (i.e. the optical flow) we need to filter each possible outlier. The median filter is robust up to 50% of outliers and this can represent a limitation in many challenging sequences. For this reason we employ a clustering procedure that produces good results even in presence of a greater percentage of outliers. The only exception to this rule is represented by rotational motion of the object, only in this case the median filter is better suited for inlier/outlier filtering.

Figure 4 shows this process: to efficiently cluster all displacements each tracker votes its displacement in the accumulator space. After all votes have been casted, the bucket with most votes identifies what we call the *best displacement* vector $\boldsymbol{\beta}$. Note that this process is equivalent to the hierarchical clustering using the infinity norm $\|\mathbf{d}\|_\infty = \max\{|d_1|, \ldots, |d_n|\}$ and a cut-off threshold of 1 but it is much more efficient. In this illustrative scenario the median filter would not produce a good results due to a high percentage of outliers (8 trackers out of 10 are outliers). Note that we use the infinity norm and not the Euclidean norm because: (i) it is more efficient and (ii) the accumulator space is partitioned into squares.

Furthermore, to improve the clustering process we assign a weight for every tracker based on its past performances (i.e. the weight is increased each time the tracker response agrees with the best displacement vector) that is used to cast a weighted vote in the accumulator space. The best displacement $\boldsymbol{\beta}$ is used to shift the center of the bounding box as follows: $\mathbf{O^{bb}_{t+1}} = \mathbf{O^{bb}_{t}} + \boldsymbol{\beta}_{t+1}$ where $\mathbf{O^{bb}}$ represents the center of the bounding box.

2.3 Consensus-Based Reinitialization

After the clustering procedure, each tracker response $\boldsymbol{\Delta_i}$ is compared to the best displacement vector $\boldsymbol{\beta}$. If their distance is greater than a threshold δ_s we set the tracker state $State(t_i)$ to outlier as follows:

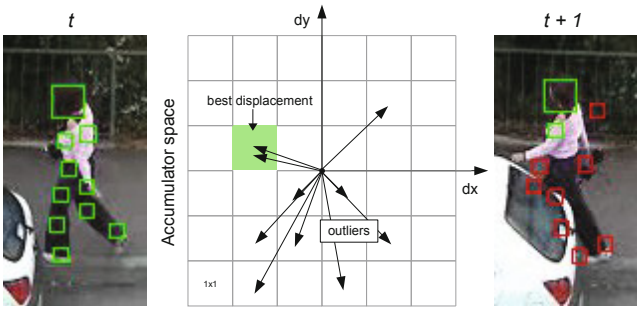


Fig. 4. Each tracker votes its displacement in the accumulator space. The most voted bucket identifies the *best displacement* vector used to update the bounding box.

$$State(t_i) = \begin{cases} inlier & \text{if } \|\boldsymbol{\Delta_i} - \boldsymbol{\beta}\|_\infty < \delta_s \\ outlier & \text{otherwise} \end{cases} \tag{1}$$

where δ_s is equal to 7. Once a tracker state is outlier it will not be used in the following frames to cast new displacement votes. For this reason we need a procedure to reinitialize the trackers when the number of inliers falls under a certain threshold δ_n (we set δ_n to 25% of the total number of trackers).

The *consensus-based reinitialization*, for every outlier tracker, performs two steps: (i) reinitializes the default position of the tracker inside the current bounding box and (ii) sets the state of the tracker to *transition*.

When a tracker state is equal to transition it will not contribute to the clustering procedure. The transition state indicates that the tracker has been reinitialized and it needs to be validated.

This validation is based on the consensus with the current inlier trackers, i.e. a tracker whose state is transition can be promoted to inlier if its response agrees (see equation 1) with the best displacement vector for at least δ_t frames following its reinitialization (we set δ_t to 3 frames) otherwise it is classified again as outlier.

Figure 5 shows the state diagram of this process, note that a tracker state, at any given time, can be either inlier or outlier or transition.

In the first frame all trackers are initialized as inliers. When the distance between a tracker displacement and the best displacement $\boldsymbol{\beta}$ is greater than a threshold δ_s, the tracker state is set to outlier and it will not be used again until the reinitialization. When the number of inliers falls under a threshold δ_n, the consensus-based reinitialization sets the state to transition to every outlier tracker. Only the transition trackers that agree for at least δ_t frames with the current inliers are promoted to the inlier state.

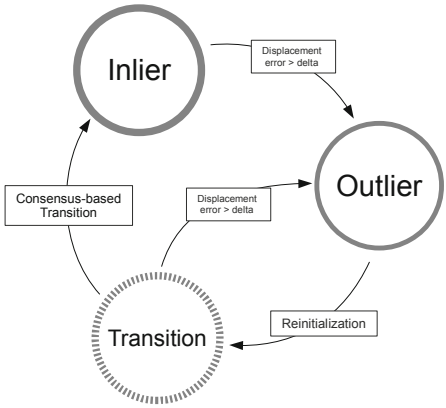


Fig. 5. The three tracker states, displayed in a state diagram

3 Quantitative Evaluation

In this section we evaluate our approach with benchmark sequences that are commonly used in the literature with the VOT2014 evaluation kit. The kit performs

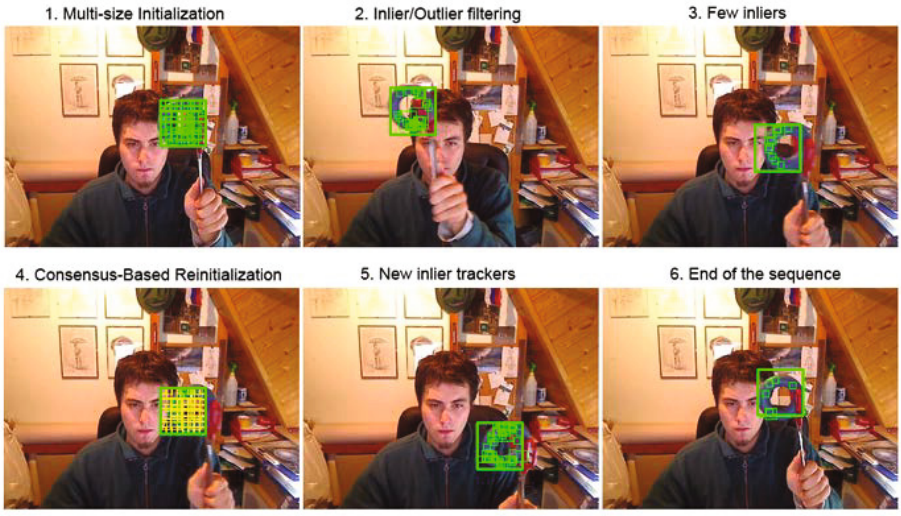


Fig. 6. BDF on *torus* sequence

Table 2. Results for tracker *BDF*

	Baseline			Region Noise		
	accuracy	**robustness**	**speed (FPS)**	**accuracy**	**robustness**	**FPS**
ball	0.52	2.00	177.21	0.52	2.80	182.02
basketball	0.56	2.00	99.82	0.47	2.33	100.47
bicycle	0.46	1.00	157.77	0.48	1.00	171.23
bolt	0.47	5.00	86.87	0.40	5.20	88.79
car	0.41	1.00	106.23	0.42	1.00	132.56
david	0.70	0.00	94.95	0.66	0.00	105.12
diving	0.29	2.00	91.68	0.30	2.13	99.40
drunk	0.53	1.00	76.59	0.49	0.80	83.82
fernando	0.42	1.00	53.48	0.40	1.47	53.78
fish1	0.29	2.00	112.43	0.28	2.67	123.92
fish2	0.23	5.00	81.10	0.17	5.53	89.39
gymnastics	0.57	1.00	62.77	0.50	1.53	67.57
hand1	0.55	1.00	89.23	0.55	1.07	92.99
hand2	0.48	1.00	87.22	0.46	1.00	85.14
jogging	0.75	2.00	117.94	0.62	1.13	113.35
motocross	0.41	0.00	64.53	0.39	1.13	91.79
polarbear	0.53	0.00	62.58	0.52	0.00	67.95
skating	0.57	2.00	69.53	0.49	1.20	75.06
sphere	0.36	0.00	109.03	0.62	0.20	110.91
sunshade	0.75	0.00	108.44	0.69	0.00	110.73
surfing	0.49	0.00	181.57	0.43	0.13	185.70
torus	0.61	0.00	66.08	0.63	0.27	78.06
trellis	0.48	0.00	124.56	0.45	0.20	163.76
tunnel	0.29	0.00	56.89	0.28	0.33	68.15
woman	0.61	1.00	68.84	0.61	1.07	73.63
Average	0.49	1.20	96.29	0.47	1.37	104.6

two experiments: Experiment "Baseline" and Experiment "Region Noise". Both the experiments are evaluated with two metrics: (i) accuracy and (ii) failures.

Accuracy is the mean overlap computed only over the valid frames on multiple trials. Failures indicate the number of times the algorithm drifted (i.e. the overlap between the tracker bounding box and the ground truth bounding box is equal to zero).

The overlap ϕ_i, given the ith frame, is defined as:

$$\phi_i = \frac{A^T \cap A^{GT}}{A^T \cup A^{GT}}$$

where A^T and A^{GT} represent the tracker bounding box and the ground truth bounding box.

As show in Table 2 BDF is able to get accuracy values of 0.49 for Baseline and 0.47 for Region Noise, while robustness values in average of 1.20 for Baseline

and 1.37 for Region Noise. We tested our C++ implementation on an Intel i7-920 processor, getting FPS of 96.29 for Baseline and 104.6 for Region Noise.

As an example, Figure 6 illustrates the Best Displacement Flow tracking the object in the *torus* sequence. In the first frame all trackers are initialized with three different patch sizes. The clustering procedure, in the following frames, identifies the *best displacement* vector that is used to: (i) update the bounding box and (ii) filter inlier/outlier trackers. When the number of inlier trackers (represented with red squares) falls under a threshold δ_n, the outlier trackers are reinitialized in their default position with *transition* state (represented with yellow squares). Only the trackers that agree with the current inliers, for at least δ_t frames, are promoted to the inlier state.

Best Displacement Flow is an optical-flow based tracker, hence it fails when the optical-flow estimation doesn't return a good result. The failure cases include: total occlusions and very large displacements between consecutive frames. The failures of *bicycle*, *basketball*, *car*, *fernando*, *fish2*, *jogging* and *woman* are due to total occlusions, whereas the failures of *bolt*, *fish1*, *fish2*, *gymnastics* and *skating* are due to abrupt appearance changes between consecutive frames.

4 Conclusions

In this paper we introduced a new short-term tracking algorithm called Best Displacement Flow (BDF) that tracks an object by robustly combining a set of local tracker estimates. We introduced two main contributions: (i) a clustering procedure to identify the *best displacement* vector and (ii) a *consensus-based reinitialization* to sample new trackers in according to the current inliers using a third state called *transition*.

Our approach reaches state-of-the-art performance and it is more robust than the median filter-based approaches in challenging sequences. Regarding future developments, it would be interesting to extend our approach by adding a re-detector module for handling situations such as: total occlusion and object out of the camera view.

References

1. Babenko, B., Yang, M.H., Belongie, S.: Robust object tracking with online multiple instance learning (2011)
2. Chen, W., Cao, L., Zhang, J., Huang, K.: An adaptive combination of multiple features for robust tracking in real scene. In: 2013 IEEE International Conference on Computer Vision Workshops (ICCVW), pp. 129–136, December 2013
3. Duffner, S., Garcia, C.: PixelTrack: a fast adaptive algorithm for tracking non-rigid objects. In: International Conference on Computer Vision (ICCV 2013), Proceedings of the International Conference on Computer Vision, pp. 2480–2487, December 2013. http://liris.cnrs.fr/publis/?id=6293
4. Felsberg, M.: Enhanced distribution field tracking using channel representations. In: 2013 IEEE International Conference on Computer Vision Workshops (ICCVW), pp. 121–128, December 2013

5. Gao, J., Xing, J., Hu, W., Zhang, X.: Graph embedding based semi-supervised discriminative tracker. In: 2013 IEEE International Conference on Computer Vision Workshops (ICCVW), pp. 145–152, December 2013
6. Godec, M., Roth, P.M., Bischof, H.: Hough-based tracking of non-rigid objects. In: Proc. Int. Conf. on Computer Vision (2011)
7. Heng, C.K., Yokomitsu, S., Matsumoto, Y., Tamura, H.: Shrink boost for selecting multi-lbp histogram features in object detection. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3250–3257, June 2012
8. Kalal, Z., Mikolajczyk, K., Matas, J.: Forward-backward error: Automatic detection of tracking failures. In: 2010 20th International Conference on Pattern Recognition (ICPR), pp. 2756–2759, August 2010
9. Kolsch, M., Turk, M.: Fast 2d hand tracking with flocks of features and multi-cue integration. In: Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW 2004) vol. 10, pp. 158. IEEE Computer Society, Washington, DC (2004). http://dl.acm.org/citation.cfm?id=1032641.1033046
10. Kristan, M., Pflugfelder, R., Leonardis, A., Matas, J., Porikli, F., Cehovin, L., Nebehay, G., Fernandez, G., Vojir, T., Gatt, A., Khajenezhad, A., Salahledin, A., Soltani-Farani, A., Zarezade, A., Petrosino, A., Milton, A., Bozorgtabar, B., Li, B., Chan, C.S., Heng, C., Ward, D., Kearney, D., Monekosso, D., Karaimer, H., Rabiee, H., Zhu, J., Gao, J., Xiao, J., Zhang, J., Xing, J., Huang, K., Lebeda, K., Cao, L., Maresca, M., Lim, M.K., El Helw, M., Felsberg, M., Remagnino, P., Bowden, R., Goecke, R., Stolkin, R., Lim, S., Maher, S., Poullot, S., Wong, S., Satoh, S., Chen, W., Hu, W., Zhang, X., Li, Y., Niu, Z.: The visual object tracking vot2013 challenge results. In: 2013 IEEE International Conference on Computer Vision Workshops (ICCVW), pp. 98–111, December 2013
11. Lebeda, K., Hadfield, S., Matas, J., Bowden, R.: Long-term tracking through failure cases. In: Proceeedings of the IEEE workshop on Visual Object Tracking Challenge at ICCV 2013. IEEE, Sydney, December 2, 2013
12. Lim, M.K., Chan, C.S., Monekosso, D.N., Remagnino, P.: Swatrack: A swarm intelligence-based abrupt motion tracker. In: MVA. pp. 37–40 (2013)
13. Maresca, M.E., Petrosino, A.: MATRIOSKA: A Multi-level Approach to Fast Tracking by Learning. In: Petrosino, A. (ed.) ICIAP 2013, Part II. LNCS, vol. 8157, pp. 419–428. Springer, Heidelberg (2013)
14. Nebehay, G., Pflugfelder, R.: Consensus-based matching and tracking of keypoints for object tracking. In: Winter Conference on Applications of Computer Vision. IEEE, March 2014
15. Ross, D.A., Lim, J., Lin, R.S., Yang, M.H.: Incremental learning for robust visual tracking. Int. J. Comput. Vision 77(1–3), 125–141 (2008). http://dx.doi.org/10.1007/s11263-007-0075-7
16. Salaheldin, A., Maher, S., El Helw, M.: Robust real-time tracking with diverse ensembles and random projections. In: 2013 IEEE International Conference on Computer Vision Workshops (ICCVW), pp. 112–120, December 2013
17. Sevilla-Lara, L.: Distribution fields for tracking. In: Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2012, pp. 1910–1917. IEEE Computer Society, Washington, DC (2012)
18. Smeulders, A.W.M., Chu, D.M., Cucchiara, R., Calderara, S., Dehghan, A., Shah, M.: Visual tracking: An experimental survey. IEEE Transactions on Pattern Analysis and Machine Intelligence 99(PrePrints), 1 (2013)
19. Čehovin, L., Kristan, M., Leonardis, A.: Robust visual tracking using an adaptive coupled-layer visual model. IEEE Computer Society, April 2013

20. Vojíř, T., Matas, J.: Robustifying the flock of trackers. In: Wendel, A., Sternig, S., Godec, M. (eds.) CVWW 2011: Proceedings of the 16th Computer Vision Winter Workshop, pp. 91–97. Graz University of Technology, Inffeldgasse 16/II, Graz (2011)
21. Vojíř, T., Matas, J.: The Enhanced Flock of Trackers. In: Cipolla, R., Battiato, S., Farinella, G.M. (eds.) Registration and Recognition in Images and Video. SCI, vol. 532, pp. 111–138. Springer, Heidelberg (2014)
22. Wu, C., Zhu, J., Zhang, J., Chen, C., Cai, D.: A Convolutional Treelets Binary Feature Approach to Fast Keypoint Recognition. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part V. LNCS, vol. 7576, pp. 368–382. Springer, Heidelberg (2012)
23. Wu, Y., Lim, J., Yang, M.H.: Online object tracking: A benchmark. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2411–2418, June 2013
24. Wu, Y., Shen, B., Ling, H.: Online robust image alignment via iterative convex optimization. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1808–1814, June 2012
25. Xiao, J., Stolkin, R., Leonardis, A.: An enhanced adaptive coupled-layer lgtracker++. In: Proceedings of the 2013 IEEE International Conference on Computer Vision Workshops, ICCVW 2013, pp. 137–144. IEEE Computer Society, Washington, DC (2013). http://dx.doi.org/10.1109/ICCVW.2013.24
26. Zhang, T., Ghanem, B., Liu, S., Ahuja, N.: Low-Rank Sparse Learning for Robust Visual Tracking. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part VI. LNCS, vol. 7577, pp. 470–484. Springer, Heidelberg (2012)