A Bayesian Approach to Tracking Learning Detection

Giorgio Gemignani[1], Wongun Choi[2], Alessio Ferone[1],
Alfredo Petrosino[1], and Silvio Savarese[2]

[1] DSA, University of Naples "Parthenope", Napoli, Italy
{giorgio.gemignani,alessio.ferone,alfredo.petrosino}@uniparthenope.it
[2] EECS, University of Michigan, Ann Arbor, USA
silvio@eecs.umich.edu, wgchoi@umich.edu

Abstract. Tracking objects of interest in video sequences, referred in computer vision literature as **video tracking** or **visual tracking**, is an essential task for intelligent machines able to understand and react to the surrounding environment. This work investigates the problem of *robust, long-term visual tracking of unknown objects in unconstrained environments*. Such problem is affected by several challenging difficulties arising from fast camera movements, partial or total object occlusions and temporal disappearance. We describe a novel framework based on *Tracking-Learning-Detection* (*TLD*), that combine bayesian optimal filtering with *pn on-line* learning theory [12] to adapt target visual likelihood during tracking. We designed particle filtering algorithm for parameter inference and propose a solution that enables accurate and efficient tracking. The performance and the long-term stability are demonstrated and evaluated on a set of challenging video sequences usually employed to test tracking algorithms.

Keywords: Visual tracking, MCMC particle filter, Adaptive likelihood.

1 Introduction

Single target tracking, defined as the problem of estimating the state X^t of an object of interest at time t in a sequence of images $I_1, \cdots, I_t$, is a fundamental issue in computer vision since it provides low-level information for a wide range of high level analysis applications, such as visual surveillance, activity analysis, vision-based user interfaces, augmented reality, etc.

Visual trackers rely on an appearance model, i.e. an internal representation of the target appearance, learned by extracting from incoming images high discriminative visual features characterizing the target. This model is then evaluated on candidate image regions, through a set of measurements, in order to estimate the most confident target location in the current frame. In [12], *single target* trackers are classified into two broad categories namely, *short term trackers* (*STT*) and *long term trackers* (*LTT*). The former refers to standard tracking approaches such as [15] that try to find frame to frame correspondences

A. Petrosino (Ed.): ICIAP 2013, Part I, LNCS 8156, pp. 803–812, 2013.

assuming no complete occlusion or disappearance of the tracked object between consecutive frames, while the latter refers to sequences of possibly infinite length, affected by frame cuts, fast camera movements and object temporary disappearance from the scene. In [12] single target tracking is defined as the problem of "*long-term on-line tracking with minimum prior information*" where the tracker learns an appearance model by continuously adapting itself to new observed data and exploiting only information from the past. Minimum prior information underlines that object modeling is formulated as semi-supervised problem where labeled data are provided manually by the user *only at the first frame of the sequence*. Such formulation requires a model able to continuously adapt to changes of appearance and at the same time, robust to wrong measurements generated by failures.

Appearance model adaptation introduces several challenges, such as the need for simultaneous fulfillment of the contradicting goals of rapid learning and stable memory referred in [8] as the *stability-plasticity* dilemma. Furthermore, on-line evaluation of new data samples becomes a critical issue in order to detect and learn changes in pose and scale or varying illumination condition. To cope with the challenges of this task, *Adaptive Appearance Trackers* (*AAT*), [1,12,10,16,24,6,2,22] rely on models able to learn changing imaging conditions. According to the type of the adopted appearance model, adaptive trackers can be grouped into three classes, namely *generative*, *discriminative* and *hybrid trackers*.

Generative trackers formulates target's appearance modeling as an unsupervised learning problem where model adaption is achieved by re-estimating target appearance distribution with new high likely samples [10,16,17,7]. Such approaches ignore discriminative information coming from the surrounding background, resulting in *high sensitivity to cluttered scenes*. On the other hand, *discriminative trackers*, using a classifier that learns a decision boundary between the appearance of the target and that of the surrounding area, w.r.t. background or other moving objects [1,12,18,24,6,2], are more *robust to clutter or resembling objects* lying in the scene. *Hybrid* trackers combine the aforementioned approaches providing more stable and flexible trackers. Authors in [21], propose to switch between discriminative and generative observation models according to targets proximity in a multi-target scenario; in [23] different generative models are aggregated by means of a weighted combination whose values are learned in each frame, by maximizing the distance to the background appearance; in [3] co-training of a short-term discriminative observation model and long-term generative one is exploited; in [14] two generative non-parametric models of target and background appearance are used to train a discriminative tracker in each frame. Authors in [12] decompose the long-term tracking task into three interacting sub-tasks, *Tracking Learning and Detection* (*TLD*), performed by three independent components. The *tracker* is a *STT* component that follows the target exploiting optical flow on local feature points lying on a regular grid generated at each tracking iteration inside the target bounding box. The *detector* localizes all appearances that have been observed so far and if necessary, corrects (re-initialize) the tracker. The *integrator* selects hypothesis coming from the aforementioned components and update the global appearance model defined by a set of patches. During the update stage, it also estimate detector errors and correct it to avoid these errors in the future, by ***pn*** *on-line learning* paradigm [12].

As stated in [19], many of these techniques ([6,22,2,12,18]) are successful in several scenarios and *TLD* is one of the best performing *AAT* paradigm, however some critical aspects of its *tracker* component have been highlighted by theoretical analysis and verified by experimentation where high sensitivity to strong occlusions and resembling background has been revealed.

Inspired by such analysis, we extensively investigated *TLD* architecture revealing a systematic drifting behavior of the *tracker* component that passes wrong hypothesis to the *integrator* component, causing in the worst case, the learning of wrong examples. We argue that such behavior is due to the design choice of tracking points over a fixed grid that is reinitialized at each tracking iteration on the previous estimated location, assuming complete visibility of the target. As it will be explained in section 2, under occlusions, this strategy drifts the *STT* component, that starts to track the occluding object (see fig. 2) until the *detector* provides more confident hypothesis. If the target have high visual similarity with the occluding object, wrong samples could be injected into the appearance model leading to an inconsistent detector and breaking the overall performance of the tracker. The *reinitialization strategy* is a challenging problem in adaptive visual tracking. In [9] a set of simple features (e.g., optical flow features) is used to track individual parts of the object while distances among features are used to add or remove salient points during tracking. Since the set of features is *geometrically unconstrained*, the tracker is likely to get stuck on the background, losing the target. In [24] *harris corner* is used to detect stable regions for tracking and enforcing a single global affine transformation constraint to avoid drifting. However, authors assume that shape of the object can be approximated with an ellipsoid and that the object does not deform, limiting the generality of the tracker.

In this work we approach the aforementioned problems by focusing short term tracking on high textured regions localized around *harris* local maxima and formulating a novel *reinitialization strategy* that is directly encoded into our probabilistic framework. The novel idea behind our *reinitialization strategy* is to add new regions of interest around high confident points tracked from the previous frame and filter out those regions that are not geometrically consistent with the best explanation of target global appearance during the inference process. Inspired by the outlier filtering scheme proposed in [5] for multiple target tracking, we designed an *Markov Chain Monte Carlo* (*MCMC*) particle filter that automatically rejects local features not consistent with the current estimate of the target location and scale. In this way stable regions are geometrically constrained to the estimated target area *without any assumption on its shape.*

We integrated our method into *TLD* approach, resulting in a new efficient and accurate long term tracker, that we named Bayesian Tracking Learning Detection (*BTLD*).

Our contributions Are Three-Fold: we propose (*i*) a tracker that make *TLD* robust to occlusions and resembling background; (*ii*) a novel bayesian model that jointly estimate target location and select new stable feature points for the next tracking iteration, exploiting adaptive visual likelihood provided by *TLD* learning component; (*iii*) quantitative evaluation on a number of challenging video sequences, verifying how our intuition corrects the baseline method respect to underlined critical conditions.

The paper is organized as follows: section 2 describes *TLD* framework and its weakness; in section 3, the proposed generative tracker and its integration in to the *TLD* framework is presented; in section 4 experimental results are showed comparing the proposed approach with the original *TLD* and other state-of-the arts methods; finally, section 4 summarizes the main contributions and highlights open research challenges.

2 Tracking Learning Detection and Its Limits

As previously introduced, *TLD* is an hybrid long term tracker that performs robust tracking by decoupling object tracking and object detection. It uses a specific object detector, trained on-line with examples found on the trajectory of a short term generative tracker that itself does not depend on the object detector. The system architecture is built on three interacting components namely the *tracker*, *detector* and *integrator*. The *tracker* component is a short term generative tracker that self-learns the appearance model and is based on *median flow tracker* [11] extended with failure detection. At each frame, it tracks by pyramidal Lucas Kanade Tracker (*KLT*) [4], a set of patches centered over points $\mathcal{K}^t = \{K^t_{i,j}\}_{i,j=1\ldots N}$ lying on a regular grid overlapped with the target estimated bounding box. *KLT* failures are controlled by a refinement step, where wrong correspondences are rejected by a median filter computed over measures of visual similarity and motion reversibility (*forward-backward error*) calculated on the set of tracked points $\mathcal{K}^{t+1} = \{K^{t+1}_{i,j}\}_{i,j=1\ldots N}$. Given $\mathcal{K}^t$ and $\mathcal{K}^{t+1}$, visual similarity is computed by normalized cross correlation between patches centered on them. *Forward-backward error* is computed, by backward tracking each point in $\mathcal{K}^{t+1}$ for k frames and measuring the geometric distance between points lying on backward and forward trajectory.

As stated before the adopted *reinitialization strategy* is prone to drift in presence of strong occlusion. In fig. 2 a clarifying example of this critical behavior is verified: the *coke* is characterized by areas of uniform color resembling surrounding background and as it moves behind the leaf the tracker component drifts. Indeed, after reinitialization, stationary local points on the leaf (*blue dots*) are identified by median operator as the correct ones, leading the final solution to drift (*yellow box*). Even if among ensemble's detections (*all other colored boxes*),

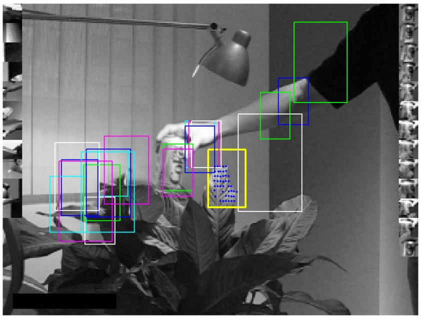

Fig. 1. *TLD* failure on *Coke* sequence. On the right column *positive samples*, on the left *negative samples*. In yellow *TLD* final solution corresponding to *median flow tracker* solution, while other colored boxes are detections provided by *detector*

the correct one is present, its confidence is still too low to activate error correction. The time required to correct the tracker depends on leaf similarity respect to target appearance and on the ability of the *detector* component to recover with an higher confidence the correct target hypothesis that enables error correction trigger. Such *detector* is a cascaded object detector, that analyzes the entire image by a scanning window approach, providing new hypothesis to the *integrator* component in order to correct the aforementioned failure. It performs three main stages: at the beginning, it applies a variance filter rejecting all patches with a variance lower than a given threshold; subsequently it classifies remaining patches by an ensemble of random *ferns* [13]; in the last step it evaluates the confidence of the detections by normalized cross correlation, computed with respect to the nearest *positive* and *negative* patches (respectively on the right and left, in fig. 2) defining the appearance model. The *integrator* component is designed to perform *model adaption* by selecting the highest confident results provided by *tracker* and *detector* to estimate current target location X^t and providing new samples to the learning process. It also bootstraps the ensemble classifier building the object detector, by ***pn**-learning* [12]. Such approach is based on structural constraints namely *P-constraint* and *N-constraint* that identify and relabel miss-classified data samples according to the assumption that all patches highly overlapping with the estimated state X^t should be classified as positive while patches far from it should be classified as negative. Retraining the detector with such strategy realizes a feature selection stage where challenging samples, in *fern* feature space, lying near the decision boundary are continuously corrected, providing a strong classifier, able to re-detect the target during drift.

3 Proposed Approach

In this work, we focus *KLT* only on salient regions defined around local maximum of *harris* operator. This strategy as stated in [20], reduces *KLT* failures to resembling background since it restricts the tracking on high textured regions, even if further refinement steps are still necessary to remove the remaining erroneous correspondences. Assuming coherent motion of tracked points, we remove those whose motion does not agree with the motion distribution estimated by kernel density estimation, over the set $\mathcal{K}^{t+1}$ of tracked points. This processes removes *KLT* serious failures but reduces drastically the number of local feature points. To guarantee a sufficient number of such salient points, new ones, detected near the remaining points are added. This is the most critical stage, since there is no prior knowledge about the "nature" of such new salient regions. The main idea is to allow the selection of new unconstrained salient points followed by a refining step where not consistent elements are rejected. In this way geometrical constraints can be encoded in our *MCMC* particle filter, where we introduce two competitive likelihood functions: one promotes the maximum number of salient points in $\mathcal{K}^{t+1}$, the other rejects local feature points that are not consistent with the visual model.

3.1 Bayesian Formulation

Following the Sequential Bayesian formulation, the posterior probability of target state X^t a time t is given by

$$\underbrace{p(X^t|\mathcal{O}^t)}_{posterior} \approx \underbrace{p(\mathcal{O}^t|X^t)}_{a} \int \underbrace{p(X^t|X^{t-1})}_{b} \underbrace{p(X^{t-1}|\mathcal{O}^{t-1})}_{c} \, dX^{t-1} \tag{1}$$

where (a), (b) and (c) in Eq. 1 represent the *observation likelihood*, the *motion model* and the *posterior* from previous time, respectively. The hidden state X^t encodes location and scale of the 2D box enclosing the target, resulting in a 4D state space $\mathbf{X}^t = [\ x\ y\ w\ h]$, where x, y, w and h are the coordinates of the center the width and the height of the bounding box, respectively. $\mathcal{O}^t = [\mathcal{K}^t\ \mathcal{A}^t]$ represents the measurement space, where $\mathcal{K}^t = \{\mathbf{K}_i^t \in \mathcal{R}^2\}$ is the set of local points tracked from previous frame and $\mathcal{A}^t$ represent the adaptive global appearance model learned on-line by *TLD*. Assuming $\mathcal{K}^t$ and $\mathcal{A}^t$ independent, the *observation likelihood* $\mathcal{O}$ is factorized by $p(\mathcal{A}^t, \mathcal{K}^t|X^t) = p(\mathcal{A}^t|X^t)p(\mathcal{K}^t|X^t)$. Observation likelihood of $\mathcal{K}^t$ measures the fraction of local feature points lying inside the candidate target state X^t: $p(\mathcal{K}^t|X^t) = \frac{\mathbf{K_i}^t \in \mathbf{X}^t}{|\mathcal{K}^t|}$. Such distribution promotes candidate states containing the maximum number of tracked local features, assuming that they are free of errors. *KLT* failures, are automatically rejected by the global appearance likelihood modeled by *TLD*. It assign low confidence to hypothesis containing local tracked points and not resembling target appearance, assuming an "outliers-rejection" role similar to *RANSAC*. $p(\mathcal{A}^t|X^t)$, measuring the normalized cross-correlation distance respect to target's patches, is given in [12]. We use a linear *dynamic model* defined by a Gaussian distribution over $\mathcal{X}$, centered on previous target X^{t-1} location and scale. Considering the complexity of the given probabilistic formulation, it is extremely challenging to design an analytical inference method for estimating the *MAP* solution. This challenge is due to the presence of the high nonlinearity of observation likelihood functions. We propose to employ a sampling based sequential filtering technique based on the *MCMC* particle filter. At each time step t given a set of N predictions on hidden variable status X^{t-1}, we propagate samples in the particle filtering framework to get an approximation of the final posterior distribution: $p(X^{t-1}|\mathcal{O}^{t-1}) \approx \{X_s^{t-1}\}_{s=1}^N$. Propagating samples through the *motion model*, we generate particles for the *predictive distribution* and approximate the posterior distribution at time t by Monte Carlo integration:

$$p(X^t|\mathcal{Y}^t,\mathcal{K}^t) \propto p(\mathcal{Y}^t|X^t)p(\mathcal{K}^t|X^t)\sum_{s=1}^{N} p(X^t|X_s^{t-1})p(X_s^{t-1}|\mathcal{A}^{t-1},\mathcal{K}^{t-1}) \tag{2}$$

Approximation in eq. 2 is achieved by a Markov chain over the joint space of $\mathcal{X}$ that converges over the posterior distribution $p(X^t|\mathcal{Y}^t,\mathcal{K}^t)$. The whole *Metropolis-Hasting* procedure is sketched in algorithm 1. For *MCMC* sampling to be successful, it is critical to have a good proposal distribution which can explore the hypothesis space efficiently. Our *proposal distribution* generates separate random hypothesis for location and scale subspaces, according to normal deviates from previous accepted hypothesis. Once the sampling method has reached convergence, the maximum a posterior estimate for X^t is analyzed by the *TLD integrator* to establish the final solution.

Algorithm 1. MCMC Particle Filter

```
procedure MCMC PARTICLE FILTER
    Input: $\mathcal{K}^t$ , $\mathcal{Y}^t$ , $X_i^{t-1}$
    Output: $p(X^t|\mathcal{Y}^t,\mathcal{K}^t)$
     Initialize $X_0^t = X^{t-1}$
    while $i < N_{accept}$ do
        Propose $X^* \sim N(X^i|X^{i-1})$
        Evaluate the acceptance probability $\alpha = min(1, \frac{p(X_s^*|\mathcal{A}^t,\mathcal{K}^t)}{p(X_s^{i-1}|\mathcal{A}^t,\mathcal{K}^t)})$
        Accept $X_s^* \rightarrow X_s^{i+1}$ if $\alpha < u \leftarrow$ uniform sample $\in [0\ 1]$
    end while
end procedure
```

4 Experimental Results

We evaluate, quantitatively, *BTLD* using challenging sequences from the *MILBoost dataset*. Each experiment in this section adopts the evaluation protocol proposed in [12]. The tracker is initialized in the first frame of a sequence and tracks the object of interest up to the end. The performance are evaluated by the average percentage of frames for which the overlap between the identified bounding box and the ground-truth bounding box is at least 50%. Authors in [19], identified in *Coke* and *Faceocc2* the most critical sequences for *TLD*.

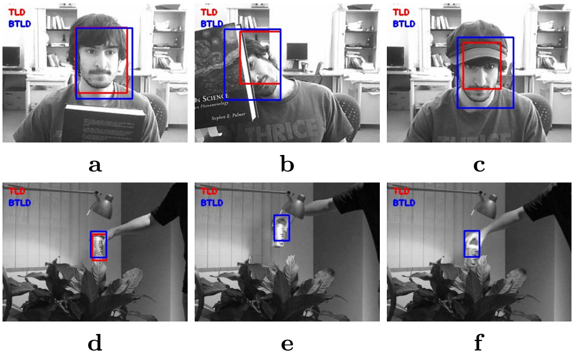

Fig. 2. From top to bottom: *Faceocc2*, *Coke*. In Red *TLD* estimated object state, in blue *BTLD* estimated object state

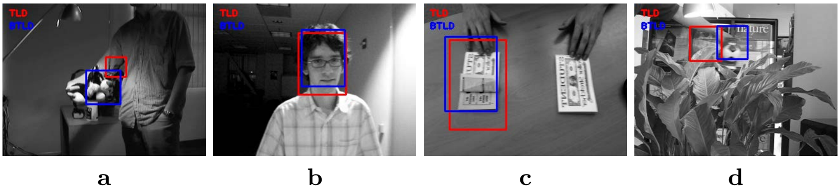

Fig. 3. Sequences *Sylvester*(a), *David*(b), *Dollar*(c),*Tiger2*(d). In Red *TLD* estimated object state, in blue *BTLD* estimated object state

As stated in Section 2, the *Coke* sequence proves sensitivity to occlusion and resembling background. The target is affected by several occlusions at the beginning of the sequence (fig. 2-**d**) resulting in a not effective object detector. Furthermore, coke continuous motion causes drifting of the *median flow* that in few frames loses the target and is unable to be restarted since the ensemble classifier does not detect the target (fig. 2-**e**,**f**). Our method, by tracking only stable points, does not lose the target (fig. 2-**e**,**f**) outperforming the baseline method and other state of the art approaches. In sequence *Faceocc2* sensitivity to occlusions and changes of appearance is analyzed, since a man is continuously occluding his face behind a book (fig. 2-**b**). Moreover during the sequence the man wears a hat (fig. 2-**c**), so that the adaptivity of the tracker to permanent changes of appearance can be evaluated. Reported frames (**a**,**b**,**c**) highlight how *BTLD* produces more accurate detection results since target state estimation is exploited by temporal consistency that controls variation in position and scale over time. Quantitative results reported in table 4 confirm the improvement in accuracy achieved respect to the baseline method. We evaluated our method also on sequences *Sylvester*, *David*, *Tiger1*, *Tiger2* and *Dollar* in order to verify the ability of our method to improve *TLD* results in other scenarios where the baseline method produces accurate tracking itself. In fig. 3 we show some conditions where our method corrects (fig. 2-**a**,**d**) or produces more accurate results (fig. 2-**b,c**) compared to the baseline method. As expected from the theoretical analysis, by fixing short term tracking instability, we increase tracking performance on all the tested sequences. Results reported in table 4 underline the improvement achieved by integrating our component into the *TLD* method. Furthermore, experiments underline how *BTLD* also affects appearance modeling since it provides more stable hypothesis to the learning component.

Table 1. *recall* measuers. The best performance on each video is boldfaced.

Sequence	*frames*	**MIL** [2]	**ORF** [18]	**TLD** [12]	**BTLD**
1. *David*	1200	0.70	0.95	1.00	**1.00**
2. *FaceOcc*	820	0.96	0.70	0.96	**1.00**
3. *Sylvester*	1440	0.93	0.71	0.97	**1.00**
4. *Coke*	292	0.46	0.17	0.60	**0.91**
5. *Tiger1*	353	0.78	0.27	0.88	**0.92**
6. *Tiger2*	364	0.80	0.21	0.85	**0.94**
7. *Dollar*	326	**1.00**	—	0.86	0.93

5 Conclusions

In this paper, we developed *BTLD*, a novel generative tracker that corrects a systematic drifting behavior revealed in the short term tracker of *TLD* approach. We designed a generative model that jointly solve feature selection and resampling exploiting a global adaptive appearance model as outlier removal. A real-time implementation of the *MCMC* particle filter framework has been described in detail and an extensive set of experiments was performed in order to highlight the ability of our approach to increase robustness of *TLD* tracker.

References

1. Avidan, S.: Ensemble tracking. In: CVPR, pp. 494–501 (2005)
2. Babenko, B., Yang, M.H., Belongie, S.: Robust object tracking with online multiple instance learning. IEEE Trans. Pattern Anal. Mach. Intell. 33(8), 1619–1632 (2011)
3. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In: Proceedings of the Eleventh Annual Conference on Computational Learning Theory, COLT 1998, pp. 92–100. ACM, New York (1998)
4. Yves Bouguet, J.: Pyramidal implementation of the lucas kanade feature tracker. Intel Corporation, Microprocessor Research Labs (2000)
5. Choi, W., Savarese, S.: Multiple target tracking in world coordinate with single, minimally calibrated camera. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part IV. LNCS, vol. 6314, pp. 553–567. Springer, Heidelberg (2010)
6. Tang, F., Brennan, S., Zhao, Q., Tao, H.: Co-tracking using semi-supervised support vector machines. IEEE Trans. Pattern Anal. Mach. Intell., 1–8 (August 2007)
7. Fan, J., Shen, X., Wu, Y.: Closed-loop adaptation for robust tracking. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part I. LNCS, vol. 6311, pp. 411–424. Springer, Heidelberg (2010)

8. Grossberg, S.: Competitive learning: From interactive activation to adaptive resonance. Cognitive Science 11(1), 23–63 (1987)
9. Hoey, J.: Tracking using flocks of features, with application to assisted handwashing. In: British Machine Vision Conference BMVC (2006)
10. Jepson, A.D., Fleet, D.J., El-maraghi, T.F.: Robust online appearance models for visual tracking, pp. 415–422 (2001)
11. Kalal, Z., Mikolajczyk, K., Matas, J.: Forward-backward error: Automatic detection of tracking failures. In: Proceedings of the 2010 20th International Conference on Pattern Recognition, ICPR 2010, pp. 2756–2759 (2010)
12. Kalal, Z., Mikolajczyk, K., Matas, J.: Tracking-learning-detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 34(7), 1409–1422 (2012)
13. Lepetit, V., Lagger, P., Fua, P.: Randomized trees for real-time keypoint recognition. In: CVPR, pp. 775–781 (2005)
14. Lu, L., Hager, G.D.: A nonparametric treatment for location/segmentation based visual tracking. In: 2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007), Minneapolis, Minnesota, USA, June 18-23, IEEE Computer Society (2007)
15. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: Proceedings of the 7th International Joint Conference on Artificial Intelligence, IJCAI 1981, vol. 2, pp. 674–679. Morgan Kaufmann Publishers Inc., San Francisco (1981)
16. Matthews, I., Ishikawa, T., Baker, S.: The template update problem. IEEE PAMI 26, 810–815 (2003)
17. Ross, D.A., Lim, J., Lin, R.S., Yang, M.H.: Incremental learning for robust visual tracking (2008)
18. Saffari, A., Leistner, C., Santner, J., Godec, M., Bischof, H.: On-line random forests
19. Salti, S., Cavallaro, A., di Stefano, L.: Adaptive appearance modeling for video tracking: Survey and evaluation. IEEE TIP 21(10), 4334–4348 (2012)
20. Shi, J., Tomasi, C.: Good features to track. In: 1994 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 1994), pp. 593–600 (1994)
21. Song, X., Cui, J., Zha, H., Zhao, H.: Vision-based multiple interacting targets tracking via on-line supervised learning. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS, vol. 5304, pp. 642–655. Springer, Heidelberg (2008)
22. Teichman, A., Thrun, S.: Tracking-based semi-supervised learning. Int. J. Rob. Res. 31(7), 804–818 (2012)
23. Yang, M., Lv, F., Xu, W., Gong, Y.: Detection driven adaptive multi-cue integration for multiple human tracking. In: 2009 IEEE 12th International Conference on Computer Vision, pp. 1554–1561. IEEE (2009)
24. Yin, Z., Collins, R.T.: On-the-fly object modeling while tracking. In: CVPR 2007, Minneapolis, Minnesota, USA, June 18-23. IEEE Computer Society (2007)