Occlusion Handling in Video Segmentation via Predictive Feedback

Jeremie Papon, Alexey Abramov, and Florentin Wörgötter

Bernstein Center for Computational Neuroscience (BCCN)
III Physikalisches Institut - Biophysik, Georg-August University of Göttingen

Abstract. We present a method for unsupervised on-line dense video segmentation which utilizes sequential Bayesian estimation techniques to resolve partial and full occlusions. Consistent labeling through occlusions is vital for applications which move from low-level object labels to high-level semantic knowledge - tasks such as activity recognition or robot control. The proposed method forms a predictive loop between segmentation and tracking, with tracking predictions used to seed the segmentation kernel, and segmentation results used to update tracked models. All segmented labels are tracked, without the use of a-priori models, using parallel color-histogram particle filters. Predictions are combined into a probabilistic representation of image labels, a realization of which is used to seed segmentation. A simulated annealing relaxation process allows the realization to converge to a minimal energy segmented image. Found segments are subsequently used to repopulate the particle sets, closing the loop. Results on the Cranfield benchmark sequence demonstrate that the prediction mechanism allows on-line segmentation to maintain temporally consistent labels through partial & full occlusions, significant appearance changes, and rapid erratic movements. Additionally, we show that tracking performance matches state-of-the art tracking methods on several challenging benchmark sequences.

1 Introduction

Unsupervised image segmentation attempts to cluster pixels into regions which represent the objects present in an image frame without human intervention. Unsupervised video object segmentation (VOS) extends this idea by linking pixels in time as well as space, to generate spatio-temporal clusters. Unfortunately, the addition of the temporal domain brings new challenges; pixels which should be grouped across time may not be continuously visible from frame to frame, as in the case of partial or full occlusions.

To overcome this, we use a novel predictive feedback mechanism which combines Bayesian tracking and VOS to preserve object labels. In this feedback mechanism, multiple particle filters in parallel track object labels, generating a prediction for segmentation, which relaxes this prediction to match the current scene. The relaxed segmentation result is then used to update the particle filters. This loop permits permanence of arbitrary objects through full occlusions.

A. Fusiello et al. (Eds.): ECCV 2012 Ws/Demos, Part III, LNCS 7585, pp. 233–242, 2012.

There are many existing video object segmentation (VOS) methods, but we shall only review here methods which meet three criteria; on-line (the algorithm may only use past data), dense (every pixel is assigned to a spatio-temporal cluster), and unsupervised. Several state-of-the-art segmentation algorithms meet these requirements: Multiple hypothesis video segmentation (MHVS) from superpixel flows [1], Propagation, validation, and aggregation (PVA) of a preceding graph [2], and Matching images under unstable segmentations [3]. Of these methods, none are able to handle full occlusions; in fact only MHVS considers occlusions, and it is only able to handle partial occlusions for a few frames, and does not consider full occlusions. Even state of the art off-line methods such as that of Brendel and Todorovic [4] only handle partial occlusions, claiming that "complete occlusions ... require higher-level reasoning".

Bayesian predictive filters (such as Particle filters) are a broad, well-established field in target tracking [5]. While effective for tracking, these methods generally depend on fixed models with a small dimensional state-space, and are unable to deal with the high-dimensionality of VOS. A recent method [6] uses graph cuts to extract segmentations, and a dynamical model to form predictions which guide successive segmentations. It formally models visible and occluded parts of the tracked objects, and so does not scale well with an increasing number of objects, and thus is better suited to extracting the silhouettes of a few objects than performing a full segmentation. Other methods, such as [7], are limited in that they require pre-computed models which are calibrated to a ground plane in order to resolve occlusions.

The paper is organized as follows. Section 2 presents the proposed algorithm; Section 2.1 gives an overview of the segmentation kernel used, and Section 2.2 discusses the predictive framework. Section 3 consists of experimental results in a specific scenario and comparison to state of the art tracking methods. Finally, Section 4 describes current limitations of the algorithm, discusses future work, and concludes.

2 Proposed Algorithm

We shall first give an overview of the algorithm (depicted in Figure 1). To begin, segmentation is performed on the first frame $\mathbf{F}_{t_0}$ to generate an initial set of labels $\mathbf{S}_{t_0}$. This is used to generate initial sets of particles, each of which contains a map of an object. Color histogram features are then generated for each object (as in [8]) and particles are initialized with randomly distributed initial velocities. Thus each object k at initial time t_0 from the segmentation is specified by a set of N_k particles $\mathbf{X}_{t_0}^{k,1:N_k}$, each of which contains a representation of the object, specified by a pixel existence map $\mathbf{M}$, a reference color histogram $\hat{q}$ calculated from $\mathbf{F}_{t_0} \cap \mathbf{M}_{t_0}^{k,n}$, a position shift vector $\mathbf{p_{t_0}}$, and a velocity vector $\mathbf{v_{t_0}}$.

Particles are then propagated in time independently, shifting their existence maps to new regions of the image. These shifted maps are used to generate new measured color histograms q_{t_0+1}, which are evaluated to determine particle weights. The set of particles for object k, $\mathbf{X}_{t_0+1}^{k,1:N_k}$, is then combined to create

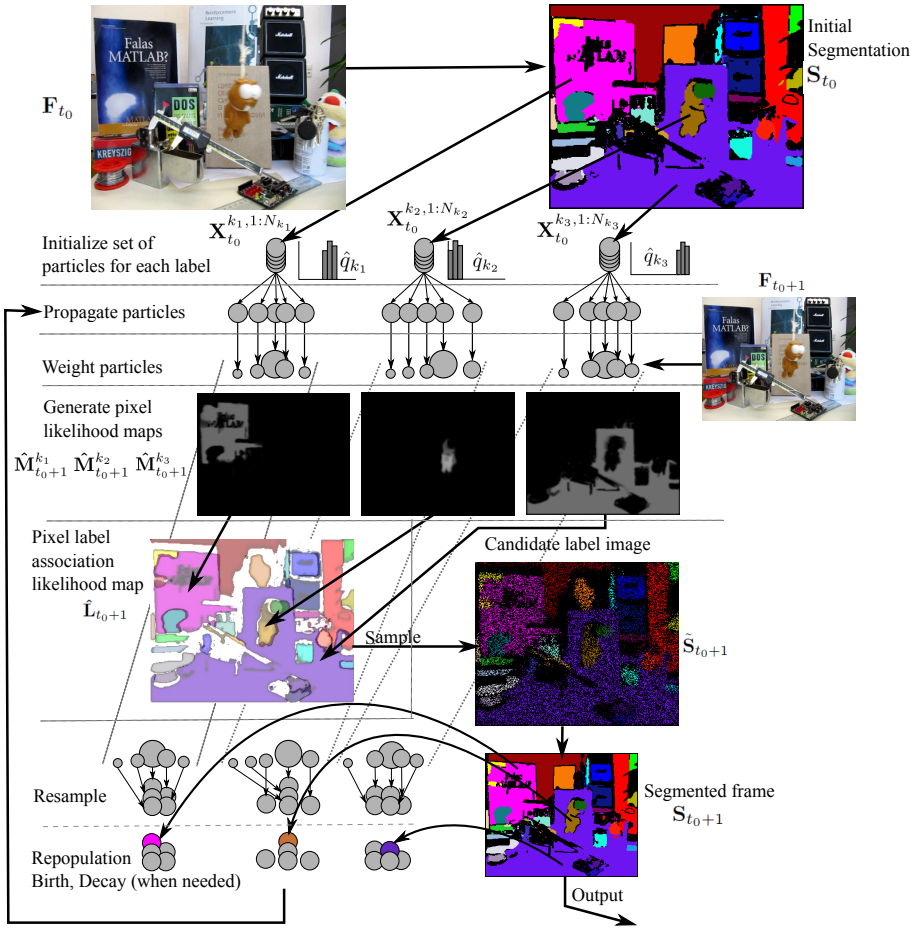

Fig. 1. Flow of algorithm for one time step, shown for three of the labels (k_1, k_2, and k_3). For description see Section 2.

an overall object pixel likelihood map $\hat{\mathbf{M}}_{t_0+1}^{k}$. The pixel likelihood maps for all objects are then used to generate a label association likelihood map $\hat{\mathbf{L}}_{t_0+1}$, where each pixel in the map is a PDF specifying the probability of the pixel belonging to each object k.

The label association likelihood map is then sampled using a per-pixel selection procedure (as described in Section 2.2) to generate a candidate label image, $\tilde{\mathbf{S}}_{t_0+1}$. This is used as the initialization for the Metropolis-Hastings algorithm with annealing of Abramov et al. [9], which updates the labels iteratively until an equilibrium segmented state is reached. The segmentation result, $\mathbf{S}_{t_0+1}$ is subsequently used to update the set of particles via three mechanisms; birth, decay, and repopulation, which are described in Section 2.2.

2.1 Segmentation

To adjust the candidate label image $\tilde{\mathbf{S}}_t$ to the current frame $\mathbf{F}_t$, we use a real-time image segmentation algorithm based on superparamagnetic clustering of data [10]. This formulates segmentation as a minimization problem which seeks to find the equilibrium states of the energy function in the superparamagnetic phase. In this equilibrium state regions of aligned spins (labels) coexist and correspond to a natural partition of the image data [10]. The equilibrium states are found using a Metropolis algorithm with a simulated annealing, called *relaxation process*, implemented on a GPU [9]. In this work, the relaxation process adjusts the predicted label image to the current frame.

Superparamagnetic clustering of data was chosen as it can use any initialization state; there are no particular requirements to the initial states of spin variables, and the closer the initial states are to the equilibrium, the less time that is needed to converge. This property makes it possible to achieve temporal coherency in the segmentation of temporally adjacent frames by using the sparse label configuration taken from the candidate label image for the spin initialization of the current frame. A final (dense) segmentation result is obtained within a small number of Metropolis updates. Conventional segmentation methods cannot generally turn a sparse segmentation prediction into dense final segments which preserve temporal coherence. Moreover, since the method can directly use sparse predictions as the seed of the segmentation kernel, we can avoid the costly block-matching procedure required to find label correspondences in other work, such as in Brendel and Todorovic [4] or Hedau et al. [3].

2.2 Predicting Object Labels

The goal of the proposed algorithm is to use predictions from Bayesian filtering to inform segmentation of higher-level temporal correspondences. It is well known that sequential Bayesian estimation methods perform well in difficult tracking scenarios [11]. Particle filtering is one such method which has been shown to approximate the optimal tracking solution well, even in complex multi-target scenarios with strong nonlinearities [5]. In this section we describe how particle filtering can be used to predict pixel associations in order to seed segmentation labels.

Parallel Particle Filters. The predictive portion of the method uses multiple Sequential Importance Resampling (SIR) filters in parallel to track multiple objects simultaneously. Objects are assumed independent and interaction between labels is not considered within the filters. Particles are first propagated using a constant velocity dynamic model, and their predicted existence maps $\tilde{\mathbf{M}}^{k,n}$ are used to generate a measured histogram, q_t. Particles are weighted based on the Bhattacharyya distance between the reference histogram $\hat{q}$ for the particle and the measured histogram q_t, and then normalized as a set for each label k. Systematic resampling is used to prevent particle degeneracy, due to its speed and good empirical performance [11].

The resulting distributions from the weighting procedure are used to generate object pixel likelihood maps for each label,$\hat{\mathbf{M}}_{t+1}^k$, which are then combined into the label association likelihood map $\hat{\mathbf{L}}_t$ (as described in the next sections), which can then be relaxed to produce a final segmented output, $\mathbf{S}_t$.

Label Image Generation. The middle portion of Figure 1 depicts how the candidate label image, $\tilde{\mathbf{S}}_t$, is generated. The candidate label image is a summary of the accumulated knowledge of the particle filters; it is a prediction of what the segmented scene should look like. That is to say, it is a pixel-wise realization of the label association likelihood map $\hat{\mathbf{L}}_t$, which is constructed by combining the object pixel likelihood maps $\hat{\mathbf{M}}_t^k$ (which approximate the posteriors of the particle sets). $\tilde{\mathbf{S}}_t$ is the seed of the segmentation kernel, which uses pixel values from $\mathbf{F}_t$ to perform the relaxation process and generate a dense label image.

Object Pixel Likelihood Maps. The object pixel likelihood map for a particular object k is the weighted sum of the pixel existence maps of all of its labels,

$$\hat{\mathbf{M}}_t^k = \sum_{n=1}^{N_k} w_t^{k,n} \mathbf{M}^{k,n}. \tag{1}$$

Because the weights have been normalized, the pixel values in $\hat{\mathbf{M}}_t^k$ will be in the range $[0, 1]$. High pixel values will occur in regions which are present in the existence maps of highly weighted particles, or alternatively, are present in many particles with average weight.

Label Association Likelihood Map. The label association likelihood map $\hat{\mathbf{L}}_t$ is a combination of all the object pixel likelihood maps, such that each pixel contains a discrete probability distribution giving the likelihood of the pixel belonging to a certain label. Additionally, a likelihood, p_0, for the pixel belonging to no label is inserted to allow pixels where no label has high likelihood to remain unlabeled in $\tilde{\mathbf{S}}_t$. More formally,

$$\hat{\mathbf{L}}_t = \bigcup_{n=1}^{K} \hat{\mathbf{M}}_t^n + p_0. \tag{2}$$

Each pixel of $\hat{\mathbf{L}}_t$ is then normalized, such that the sum of the discrete probabilities sums to one. The candidate label image can then be generated by taking a realization of $\hat{\mathbf{L}}_t$ to select pixel label values. An example of the result of this process, $\tilde{\mathbf{S}}_t$, can be seen in Figure 1.

Particle Birth, Repopulation, and Decay. A key feature of the method is use of segmentation results $\mathbf{S}_t$ to update the particle sets. This allows the creation of new object labels, adaptation to changing object appearance, and

elimination of objects which are no longer observed. This is accomplished via three mechanisms; birth, repopulation, and decay.

Birth occurs when a label which has not existed previously is found in the segmentation output $\mathbf{S}_t$. It consists of generating a set of particles $\mathbf{X}^k$ for the new label using $\mathbf{S}_t$ to initialize an existence map $\mathbf{M}_t^k$ and $\{\mathbf{F}_t \cap \mathbf{M}_t^k\}$ to calculate a reference color histogram $\hat{q}_t^k$. Decay occurs when a label is not found in the segmentation output, $k \notin \mathbf{S}_t$. Particles are selected from k using random sampling, at a rate determined by the decay rate λ_d, and are pruned; they are no longer considered when filtering k. If the number of active particles for a label falls below a certain threshold, N_{min}, then the set of particles for the label is deleted, and the object is no longer tracked.

Repopulation allows the pixel likelihood map for an object, $\hat{\mathbf{M}}^k$, to adapt over time to the changing appearance of the object. Every iteration, all previously existing object labels which are found in $\mathbf{S}_t$ are repopulated by replacing some particles in the set with particles generated from $\mathbf{S}_t$ and $\mathbf{F}_t$. Particles are chosen for replacement using stratified sampling, at a rate specified by parameter λ_r. The repopulation mechanism gradually modifies the object "model" through the addition of particles which have an updated existence map and color histogram (coming from the segmentation result). Note that there is no explicit model for the objects shape, only a pixel likelihood map generated at each time step by weighting an objects constituent particles using the current image frame.

Occlusion Handling. Occlusion relationships are handled naturally, since foreground objects will tend to have a strong peak in their weight distribution, corresponding to those particles which align properly with $\mathbf{F}_t$. Objects they occlude will have a flat particle weight distribution, since there will exist no shifted existence map which contains a color distribution which matches the reference histogram exactly. This is due to the fact that the occluding objects and objects surrounding the occluded object have color distributions which differ from the occluded object. Let us assume foreground object j is contained by occluded object k, that is

$$\mathbf{M}_t^{j,n} \subset \mathbf{M}_t^{k,n}. \tag{3}$$

We also assume that the number of particles is sufficiently large such that

$$\exists\ \mathbf{M}_t^{j,n} \in \mathbf{M}_t^j : hist(\mathbf{F}_t \cap \mathbf{M}_t^{j,n}) \approx \hat{q}^{j,n}. \tag{4}$$

If the objects have different color distributions then from (3) and (4)

$$\nexists\ \mathbf{M}_t^{k,n} \in \mathbf{M}_t^k : hist(\mathbf{F}_t \cap \mathbf{M}_t^{k,n}) \approx \hat{q}^{k,n} \tag{5}$$

therefore

$$\begin{aligned} &min_{1:N_j}\{\Delta(\hat{q}^{j,n}, hist(\mathbf{F}_t \cap \mathbf{M}_t^{j,n}))\} < \\ &\quad min_{1:N_k}\{\Delta(\hat{q}^{k,n}, hist(\mathbf{F}_t \cap \mathbf{M}_t^{k,n}))\} \end{aligned} \tag{6}$$

and thus

$$max_{1:N_j}\{w_t^{j,n}\} > max_{1:N_k}\{w_t^{k,n}\}. \tag{7}$$

This means that in the label association likelihood map $\hat{\mathbf{L}}_t$, the occluding object will have a higher likelihood then the occluded. The candidate label image, $\tilde{\mathbf{S}}_t$ will therefore tend to favor occluding object labels, which will dominate the occluded object label during the segmentation relaxation process.

3 Experimental Results

In order to evaluate performance, we demonstrate occlusion-handling in the context of the Cranfield benchmark, a test scenario used in robotics research to evaluate ability to plan and execute goals. The benchmark consists of building a "widget" consisting of several simple parts such as pegs. In this work, we segment a recording of a human constructing the Cranfield benchmark in order to demonstrate the ability to distill meaningful semantic information from object labels. We emphasize meaningful because the recording contains many occlusions, which cause all other state-of-the-art VOS methods to lose track of labels, spoiling the semantic information contained in the segmentation result (for instance, if a peg changes label when occluded by a hand). In all tests, we employ no learned or a-priori specified models and use 100 particles per label (this runs at 10fps at 640x480 with a GPU implementation).

Figure 2 (see supplementary material for full video) shows the ability of the algorithm to handle full and partial occlusions in the Cranfield sequence. Objects which are temporarily occluded by the hand regain their original labels once they are no longer occluded. Additionally, as objects deform (for instance, as a peg is rotated), tracking successfully maintains their correct labels, while segmentation and repopulation adapt the masks to their changing appearance.

The visual quality of segmentation results are not evaluated here as they have been presented in [9]. Additionally, we do not evaluate other VOS methods, as it is clearly stated in the literature that they fail under partial [2] and full [4,1] occlusions (see supplementary material for an example). Instead, we evaluate the algorithm from a pure tracking standpoint, as these methods are currently able to cope with full occlusions. We compare to the state of the art on several challenging video tracking benchmark sequences which are available online[1]. Results are compared to PROST [12], MilTrack [13], FragTrack [14], and ORF [15]. Details concerning the parameters used for the above algorithms in the benchmarking can be found in [12].

In order to compare with the other methods, we needed to output a tracking rectangle for each frame - we simply used the bounding box of the tracked label. This was compared to ground-truth using two measures; Euclidean distance and the PASCAL-challenge based score proposed in [12]. The latter compares the area of intersection of the ground truth and tracked box with the union of the same. When this is greater than 0.5, the object is considered successfully tracked. Table 1 gives our results and the results for the other methods.

[1] `http://gpu4vision.icg.tugraz.at/index.php?content=subsites/prost/prost.php`

Fig. 2. Output frames from the *Cranfield* sequence, in which objects are completely occluded by an arm, and change in appearance when manipulated. Labels are clearly maintained through partial and full occlusions, as well as through manipulations and changes in appearance. $\mathbf{F}_t$-Original frames. $\mathbf{S}_t$-Segmentation output. $\hat{\mathbf{M}}_t^{arm}$ & $\hat{\mathbf{M}}_t^{widget}$-Object pixel likelihood maps for the arm and widget base plate labels. Intensity represents the sum of the normalized weights of the set of particles.

Table 1. PROST dataset benchmark results. Numbers given are average pixel error (APE) and PASCAL scores, given as APE | PASCAL

Sequence	PROST	MIL	Frag	ORF	Segmenting-PF
Lemming	25.1 | 70.5	14.9 | 83.6	82.8 | 54.9	166.3 | 17.2	19.8 | 73.9
Box	13.0 | 90.6	104.6 | 24.5	57.4 | 61.4	145.4 | 28.3	114.1 | 7.5
Liquor	21.5 | 85.4	165.1 | 20.6	30.7 | 79.9	67.3 | 53.6	25.5 | 54.2
Board	39.0 | 75.0	51.2 | 67.9	90.1 | 67.9	154.5 | 10.0	30.9 | 71.4

Testing showed that, when certain assumptions hold, our algorithm performs on par with, and in some cases outperforms, state of the art tracking algorithms. This is the case for the *liquor*, *lemming*, and *board* sequences. In the *lemming* sequence (shown in supplementary material), our algorithm outperforms the other methods in cases of occlusion, especially when the tracked object is fully occluded. While other methods offer false positives and erroneous tracks, our method decays the label for the object and avoids proposing incorrect tracking solutions. In addition to showing the strengths of our method, a weakness was also highlighted by the benchmark sequences. The *box* sequence demonstrated the limitations of using unsupervised color-based segmentation to initialize the objects to track. In the sequence, the object to track contains strong color differences, which are segmented into different initial regions. As the object moves around, the particles for these regions are attracted to other objects it passes over which have similar color. This will be addressed in future work, which will use a measurement model more heavily weighted on geometric information rather than color.

4 Conclusion

This paper presented a method for performing on-line, dense, unsupervised video segmentation which uses feedback to handle occlusions. Results showed that the method is able to resolve occlusion relations between objects without explicitly modeling them, and by doing so can maintain consistent labels for objects, even through partial or full occlusions. Additionally, the method is able to adapt to rapidly changing appearance of tracked objects, producing consistent segmentations over lengthy video sequences. The combination of these in an unsupervised online algorithm enables new robotics research, as it allows extraction of semantic information directly from segmented labels. This semantic information (how objects interact with eachother) could be used to bootstrap unsupervised learning algorithms and generate plans for complex tasks - such as building the Cranfield benchmark.

Future work will address the limitations of the measurement model by the addition of geometric features extracted from point cloud data. Additionally, movement of labels while occluded is an area of open research; currently, the algorithm will diffuse particles following the most recent velocity vector (before occlusion), and does not associate occluded particles with the motion of the occluder. Finally, we should note that this work shows the need for a standardized video benchmark which evaluates segmentation performance in complex scenarios (such as movement while occluded). In particular, the community needs to select a set of complex scenarios, and come to a consensus as to what constitutes "correct" labeling of them.

Acknowledgments. The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 (Specific Programme Cooperation, Theme 3, Information and Communication Technologies) under grant agreement no. 270273, Xperience and grant agreement no. 269959, Intellact.

References

1. Vazquez-Reina, A., Avidan, S., Pfister, H., Miller, E.: Multiple Hypothesis Video Segmentation from Superpixel Flows. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part V. LNCS, vol. 6315, pp. 268–281. Springer, Heidelberg (2010)
2. Liu, S., Dong, G., Yan, C.H., Ong, S.H.: Video segmentation: Propagation, validation and aggregation of a preceding graph. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR (2008)
3. Hedau, V., Arora, H., Ahuja, N.: Matching images under unstable segmentations. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR (2008)
4. Brendel, W., Todorovic, S.: Video object segmentation by tracking regions. In: IEEE International Conference on Computer Vision, ICCV (2009)
5. Vermaak, J., Godsill, S., Perez, P.: Monte carlo filtering for multi target tracking and data association. IEEE Transactions on Aerospace and Electronic Systems 41, 309–332 (2005)
6. Papadakis, N., Bugeau, A.: Tracking with occlusions via graph cuts. IEEE Transactions on Pattern Analysis and Machine Intelligence 33, 144–157 (2011)
7. Ablavsky, V., Thangali, A., Sclaroff, S.: Layered graphical models for tracking partially-occluded objects. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR (2008)
8. Pérez, P., Hue, C., Vermaak, J., Gangnet, M.: Color-Based Probabilistic Tracking. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002, Part I. LNCS, vol. 2350, pp. 661–675. Springer, Heidelberg (2002)
9. Abramov, A., Pauwels, K., Papon, J., Wörgötter, F., Dellen, B.: Real-time segmentation of stereo videos on a portable system with a mobile gpu. IEEE Transactions on Circuits and Systems for Video Technology (in press)
10. Blatt, M., Wiseman, S., Domany, E.: Superparamagnetic clustering of data. Physical Review Letters 76, 3251–3254 (1996)
11. Doucet, A., De Freitas, N., Gordon, N. (eds.): Sequential Monte Carlo methods in practice (2001)
12. Santner, J., Leistner, C., Saffari, A., Pock, T., Bischof, H.: Prost: Parallel robust online simple tracking. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR (2010)
13. Babenko, B., Yang, M.H., Belongie, S.: Visual tracking with online multiple instance learning. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR (2009)
14. Adam, A., Rivlin, E., Shimshoni, I.: Robust fragments-based tracking using the integral histogram. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR (2006)
15. Saffari, A., Leistner, C., Santner, J., Godec, M., Bischof, H.: On-line random forests. In: IEEE International Conference on Computer Vision Workshops, ICCV Workshops (2009)