Single target tracking using adaptive clustered decision trees and dynamic multi-level appearance models

Jingjing Xiao
School of EESE
University of Birmingham
shine636363@sina.com

Rustam Stolkin
School of Mechanical Engineering
University of Birmingham
r.stolkin@cs.bham.ac.uk

Aleš Leonardis
School of Computer Science
University of Birmingham
a.leonardis@cs.bham.ac.uk

Abstract

This paper presents a method for single target tracking of arbitrary objects in challenging video sequences. Targets are modeled at three different levels of granularity (pixel level, parts-based level and bounding box level), which are cross-constrained to enable robust model relearning. The main contribution is an adaptive clustered decision tree method which dynamically selects the minimum combination of features necessary to sufficiently represent each target part at each frame, thereby providing robustness with computational efficiency. The adaptive clustered decision tree is implemented in two separate parts of the tracking algorithm: firstly to enable robust matching at the parts-based level between successive frames; and secondly to select the best superpixels for learning new parts of the target. We have tested the tracker using two different tracking benchmarks (VOT2013-2014 and CVPR2013 tracking challenges), based on two different test methodologies, and show it to be significantly more robust than the best state-of-the-art methods from both of those tracking challenges, while also offering competitive tracking precision.

1. Introduction

After several decades of visual tracking research, even the most sophisticated trackers are still prone to failure in challenging scenarios, including clutter and camouflage in one or more feature modalities, rapid and erratic target motion, occlusions, and targets which change their shape and appearance over time. These problematic tracking conditions predominantly lead to failures in three fundamental parts of the tracking algorithm: 1) the model of the target object's visual appearance; 2) the mechanism for matching model parts to image regions at each frame; 3) the mechanism for continuously relearning or updating models of targets which change their appearance over time.

This paper presents a target tracking algorithm which achieves state-of-the-art robustness by addressing each of these three fundamental areas. We propose a flexible target representation which can adaptively exploit an arbitrary number of different image features. Targets are modelled at three different levels of granularity, including the level of individual pixels, the level of parts, and a bounding box level which encodes overall information about the target as a whole. Cross-constraints between these different levels during updates, enable continuous target model relearning which is robust and stable.

The main contribution of the paper is an adaptive clustered decision tree approach which dynamically selects the minimum combination of features necessary to sufficiently represent each target part at each frame, thereby providing robustness without sacrificing computational efficiency. We show how this adaptive clustered decision tree can be utilised in two separate parts of the tracking algorithm: firstly to enable robust matching at the part-based level at every frame; and secondly to select the best superpixels for learning new parts of the target. During matching, the adaptive clustered decision tree is used to search the set of superpixels in the current frame, to find the best match to a target part in the previous frame. During model updating, the decision tree is used to search for the most suitable superpixel from which to model a new target part, in order to replace an old target part which has been detected as drifting.

We have carried out a principled evaluation using the latest benchmarks, and comparing against the best other state-of-the-art trackers. Results show that the proposed method outperforms the best 4 trackers on both VOT 2013 and 2014 benchmark sets. It also significantly outperforms the 7 best available methods on the CVPR2013 dataset wrt robustness, while also achieving competitive tracking accuracy.

The rest of the paper is organized as follows. Related work is discussed in Sec. 2. The multi-level target model, and its initialisation, is introduced in Sec. 3. Sec. 4 explains the adaptive clustered decision tree, and shows how it is used for both target matching and model updating at each successive frame. Sec. 5 presents and discusses the experimental results of testing the tracker on the VOT2013, VOT2014 and CVPR2013 benchmark video datasets. Sec. 6 provides concluding remarks.

2. Related work

In this section, we review recent tracking algorithms in terms of three primary components: target representation, matching mechanism, and model update mechanism.

Choice of target representation is a crucial component of any tracker. Two main streams of research can be distinguished. The first uses holistic (overall) target templates for tracking, e.g. [22] and [24], itself derivative from the fundamental ideas of [10]. However, such methods have difficulty in handling significant appearance changes and deformations of the target. Later work, [27] showed how contextual information can be used to adaptively select in favour of whichever part of the target model is most discriminating in the current frame, but was still limited to a simple holistic target model which did not itself update. Later work, [18] [19] proposed patch-based approaches to provide more flexibility for target matching. However, the choice of geometric constraint for local patch movements remains an open problem, while environmental clutter and camouflage can often distract such local patches and cause them to drift. [33] avoided complex geometric constraints for patch motion, by treating the problem as a classification of foreground and background superpixels. However, since each superpixel is classified independently, this method remains prone to failure with cluttered background scenes. In contrast, our method also makes use of superpixels, but exploits them within a more robust cross-constrained multi-level target model structure. More recent work [30] also combined both global and local patch-based target models together for a more robust representation. However, this work fused multiple features in a homogeneous way, which causes failures under conditions where one or more features become less discriminating than others. In contrast, our method achieves better results by adaptively selecting in favour of whichever feature or feature combination is most discriminating for each target part in each new frame. Other recent work [21] simultaneously tracks, learns and parses targets using a hierarchical and compositional And-Or graph representation. However, the algorithm uses a relatively fixed cell grid to quantize the AOG structure, which may contain nodes with little semantic meaning. In contrast, our method links tree nodes directly to target superpixels, which are more likely to represent homogeneous and meaningful parts of the target object. More recent work is proposed in [17] which conducted tracking based on a tree-structured target appearance model. They estimate the optimal tree using a number of key frames, and employ a semi-supervised manifold alignment technique to construct a tree for all frames. However, this off-line tracker is not suitable for sequential tracking of targets which continuously change their shape and visual appearance over time. In contrast, we propose a highly adaptive decision tree structure, which is relearned for each target part at each new frame, and this enables good results for videos with rapidly deforming targets.

To estimate the state of the target, the algorithm must match observations from a candidate image region to the target representation model. A single feature is not sufficient to handle large appearance variations, and recent works [25] [29] [28] [21] increasingly exploit combinations of multiple features. One approach is to compute the likelihood from all features and then multiply all values to estimate the target state, e.g. [30]. However, in such schemes, a poorly performing feature can damage tracking performance, even when other features are highly discriminative. Therefore, instead of treating all features with equal importance, other methods, e.g. [8] [32] [29] [28], attempt to identify and weight in favour of the most discriminative features (or the most discriminative parts of a target model) at each time step. Brasnett et al. [6] propose a scheme for weighting in favour of the best performing features, and updating these weights adaptively at each new frame. However, this method ignores feature saliency from the local background regions. In contrast, recent works [29] and [28] propose two different kinds of adaptive methods which both successfully exploit contextual information for optimally weighting the contributions from each feature online during tracking. However, those methods use only a simple holistic target model which is insufficient to cope with large target deformations and appearance changes. Pernici et al. [25] propose a matching method which uses both the target and context SIFT features. However, the matching indices are obtained directly by a nearest neighbour search, which might perform poorly when the target undergoes rapid and significant deformations and appearance changes. In contrast to the homogenoeus treatment of all features, e.g. [30], our adaptive clustered decision tree approach can adaptively select in favour of the most discriminative features for matching each target part to each new frame. However, this adaptive feature selection is also embedded within a cross-constrained multi-level target representation, which enables much more robust matching and model updating than the simple holistic target models of e.g. [29] or [28].

For robust, general tracking, it is essential to continuously update or relearn the target model to cope with appearance changes. An appropriate target model should enable the tracker to overcome errors in the relearning process which might corrupt the target model, and support long term tracking without drifting [22]. Early methods, such as [24], updated the model at every frame as a simple linear combination of the previous model and the most recent estimation of the target region in the current image (sometimes referred to as an autoregressive update procedure). Without additional methods for precise delineation of the target parts, such update methods are likely to fail, given sufficiently long tracking duration, due to accumulated errors and noise during successive updates. In MIL [4] and other trackers such as OB [12] and SB [13] updating of the target model is performed by an evolving boosting classifier that tracks image patches and learns the object appearance. Interestingly, OB [12] can be regarded as a non-Bayesian approximation to the simpler Bayesian ABC method [27] but which enables continuous relearning of the target model. However, online boosting requires that the data be independent and identically distributed, which is a condition not satisfied in most real video sequences, where data is often temporally correlated [25]. A more robust updating mechanism is achieved by [30], which forms a cross constraint paradigm to stably constrain the relearning of a two-layer target model - global (bounding region) and local (parts based) models are used to constrain (and thereby stabilise) each others' online relearning. However, this method (and most earlier methods e.g. [24]) updates target appearance models at a fixed rate, regardless of the confidence (or lack of) in current target observations. This problem is compounded by the previously described problem, that many methods, e.g. [30], combine the opinion of all features with equal weight, which can lead to tracking failures when one or more feature modalities are poorly discriminative compared to others. Like [30], we also employ a multi-level cross-constraint approach to robustify online target relearning, however we achieve significantly improved performance evaluation results over [30] by adaptively varying the relearning rate, for each target part at each frame, based on a current tracking confidence measure derived from contextual information.

3. Multi-level target model

In principle our method can be used with any combination of features, provided that feature values of pixels in an image region can be associated with a model (e.g. a histogram), and that a suitable likelihood function exists to compare the similarity of such models for two such regions (for proof of principle we use a metric based on Bhattacharyya distance [5] although others are also possible). In the spirit of [2] and [34], our method initialises its target model using only a given bounding box in the first frame. The target is modeled hierarchically at three different levels of granularity: the pixel (bottom) level, parts-based (middle) level, and the bounding box (top) level. Following the logic of the model construction and the order of model updating, this section first introduces the middle level model, since the initial features are extracted from this level. Later, we introduce the top level followed by the bottom level.

The top level (overall target appearance model) is compared against a uniformly sampled set of candidate bounding box locations, and the expectation operator is used to provide a candidate bounding box region where the new **middle level** model can be created. This candidate bounding box is then segmented into superpixels, using the SLIC superpixel segmentation method [3]. These superpixels are then used to identify a set of suitable target parts to form the middle level model consisting of M parts, each associated with N features, i.e. $\{f_k^{i,m}\}_{i=1\ldots N,m=1\ldots M}$ where $f_k^{i,m}$ represents the m^{th} feature of the i^{th} part at k^{th} frame.

Using superpixels as the basis for middle level feature histograms (denoted as $\{h_k^{i,m}\}_{i=1\ldots N,m=1\ldots M}$) offers several advantages over randomly selecting square "patches" as in the previous method [30]. Firstly, a superpixel is much more likely to correspond to a semantically meaningful and homogeneous part of the target. In contrast, randomly (or uniformly) selected patches are likely to contain pixels from two or more dissimilar (e.g. in terms of colour) parts of the target, which can lead to matching problems. Secondly, when patches are randomly (or uniformly) selected from an initial bounding box, many patches are likely to contain information drawn from both target and background pixels, and this is also likely to negatively impact tracking performance. In contrast, due to the homogeneous nature of superpixels, the features extracted from these regions are much more likely to include either purely target pixels, or purely background pixels. Once tracking begins, those patches which erroneously correspond to background regions are rapidly detected and eliminated from the target model, leaving only those patches which truly belong to target region pixels. We use the SLIC superpixel segmentation method [3], because it offers the following advantages: the number of superpixels can be known in advance, the superpixels have uniform size, the compactness can be defined and the algorithm has high computational efficiency. Then, the middle level model ζ_k is a set of N parts (small rectangles), extracted from N superpixels denoted as:

$$\zeta_k = \{c_k^i, h_k^{i,m}\}_{i=1\ldots N,m=1\ldots M} \tag{1}$$

where c_k^i are the image coordinates of the part i at the frame k. Every part is represented by a set of histograms, one histogram for each feature modality. I.e. $\{h_k^{i,m}\}_{i=1\ldots N,m=1\ldots M}$ denotes a set of $M \times N$ histograms which model M different features in each of N different middle level target parts, derived from N superpixels.

The top level of the target model is denoted as $\Im_k$ which includes overall information about the bounding box region:

$$\Im_k = \{C_k, H_k^{B,m}, H_k^{F,m}\}_{m=1\ldots M} \tag{2}$$

where C_k are the bounding box image coordinates (recomputed at each new frame from the distribution of parts in the middle level model) at frame k. The background feature histograms $\{H_k^{B,m}\}_{m=1...M}$ are the histograms for each of M different feature modalities, extracted from a ring-shaped local background region defined by enlarging the target's bounding box by a scale factor. $\{H_k^{F,m}\}_{m=1...M}$ are a set of foreground histograms, in each of M feature modalities which are computed by combining information from the middle level parts models as:

$$H_k^{F,m} = \frac{1}{N} \sum_{i=1}^{N} h_k^{i,m} \tag{3}$$

where $h_k^{i,m}$ indicates the histogram of the m^{th} feature in the i^{th} middle level part.

The bottom level comprises individual pixels, each associated with its RGB value and an assigned likelihood. This likelihood is computed using the top level foreground and background histograms via simple Bayesian reasoning. Let $f^m(x)$ be the value of the m^{th} feature at pixel location x. Furthermore, let $H_k^m(f^m(x))$ denote a special function, which takes as argument the feature value $f^m(x)$ at image location x, and outputs a probability corresponding to the value of the relevant bin in the histogram H_k^m, which denotes a histogram of features of type m in image frame k.

According to the statistics of the m^{th} feature modality, the likelihoods of a pixel corresponding to foreground or background regions are now $P(x|F) = H_k^{F,m}(f^m(x))$ and $P(x|B) = H_k^{B,m}(f^m(x))$ where the symbols F and B (and superscripts H^F and H^B) denote foreground and background information respectively.

In any particular feature modality, the overall probability of a particular pixel representing the tracked object can be computed as

$$P_k^m(F|x) = \frac{\lambda P(x|F)}{\lambda P(x|F) + (1-\lambda) P(x|B)} \tag{4}$$

where λ represents the expected ratio of the size of the target to the size of the image region being searched, in pixels. Typically, the search area and bounding boxes are continually scaled to ensure an approximately constant value of λ. Note that it is trivial to extend the formulation of Eq. 4 to include arbitrarily many feature modalities.

4. Tracker propagation and matching

An overall schematic for the tracker is shown in Fig. 1. The tracker is first propagated by information matching at the top level, which generates a candidate image region for the middle level. Next, this candidate region is segmented by [3] into equally sized superpixels. We next propose a continuously-adaptive clustered tree method, which efficiently finds the best correspondences for matching middle level target model parts from the previous frame, onto newly segmented superpixels in the new frame. The continuously adaptive clustered tree method is efficient in that it adaptively makes use of the minimum number of features for matching each target part at each image frame. Finally, to cope with target deformation and appearance changes, the appearance of old target parts (middle level patches) can be adaptively updated, or severely drifting patches can be temporarily switched off altogether, and replaced by new patches. In such cases, a new kind of adaptively clustered decision tree is used to choose the most suitable superpixels for forming the new parts.

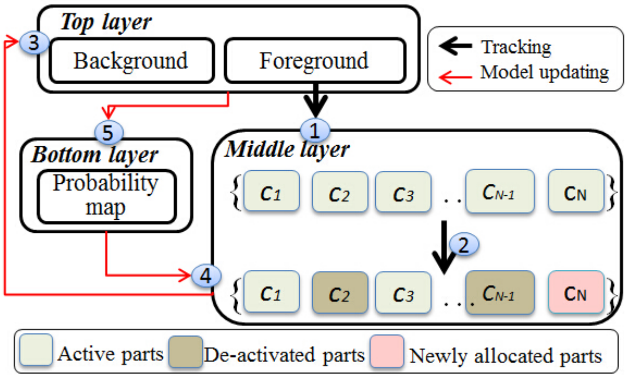

Figure 1. Block diagram of tracking process. 1- tracker propagation by foreground information matching at the top level; 2- middle (super-pixel) level matching with clustered decision tree; 3- update the top level model; 4- feed back the bottom (pixel) level information; 5- re-sample drifting parts at the middle level.

4.1. Top level propagation

The top level is propagated using a set of uniformly distributed samples to represent the posterior density function of the target, with associated weights. The same as [24], the overall target position is estimated by using the expectation operator over the set of samples, whose weights are computed by the foreground feature matching.

4.2. Middle level matching with adaptive clustered decision tree

After propagating the top level to give a candidate bounding box region, this is further enlarged to form a broader search region, which we then segment into superpixels, [3]. These superpixels now provide a pool of candidate loctions in the new image for matching to middle-level target parts (patches) in the previous image. This matching is performed using our proposed adaptive clustered decision tree method. Typically, a decision tree's structure will be obtained offline by training, which would be unsuitable for tracking targets with dynamic appearance. Instead, we propose a fully adaptive clustered tree which is relearned online for each new target part in each new image, by explicitly considering foreground and background information.

The proposed adaptive clustering decision tree method is illustrated in Fig. 2. The objective of the decision tree is to

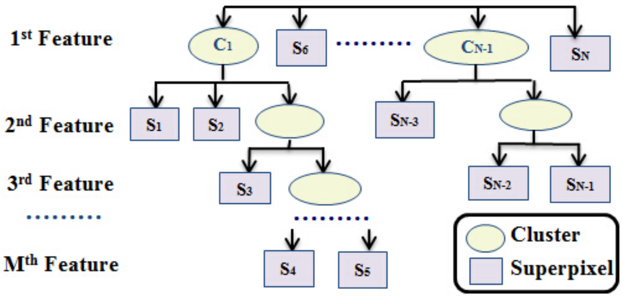

Figure 2. Clustered decision tree. Each tree-level represents a feature modality (e.g. color, motion etc.). The feature values of superpixels form leaves on a tree-level. If any two leaves are sufficiently similar in feature value then they are merged into a cluster. A new tree-level (using the next feature modality) is then used to try to disambiguate the members of such clusters. The tree continues growing (by adding more tree-levels of more features), until all target parts have been assigned to a unique choice of superpixel. Any remaining unmatched parts are assumed to have become occluded and are temporarily switched off.

find the corresponding superpixel which best matches each middle level target part, while adaptively selecting the minimum combination of features needed to do this robustly in each frame. The first tree-level is initialised by selecting a primary feature from the set of all features, and labelling all superpixels as individual leaves. First, each leaf is compared against the part in the middle level target model for which a match is sought.

$$\begin{aligned} arg \max_{(i_P,i_c)} \ & \mathcal{B}(h_k^{i_P,m}, h_k^{i_c,m}) \quad s.t. \\ & \mathcal{B}(h_k^{i_P,m}, h_k^{i_c,m}) > 0 \end{aligned} \tag{5}$$

where i_P indicates the part in the middle level target model, i_c is the candidate leaf and $\mathcal{B}(,)$ is the similarity metric. Next, the remaining candidate leaves are compared to the selected leaf, denoted as i_s, for similarity in the primary feature modality. Similar leaves are grouped into a cluster when:

$$\mathcal{B}(h_k^{i_s,m}, h_k^{i_c,m}) > T_k^m \tag{6}$$

where $\mathcal{B}(h_k^{i_s,m}, h_k^{i_c,m})$ is the similarity metric for m^{th} feature histogram between a selected leaf i_s and the other candidate leaf i_c and T_k^m is a clustering threshold.

Note T_k^m has an important effect on the extent to which each feature is used in the mid-level part matching procedure. High values of T_k^m reduce the amount of clustering on the respective tree-level, ensuring that most superpixels will be represented as individual leaves, i.e., the algorithm will distinguish most superpixels using this feature alone. In contrast, low values of T_k^m make it likely that this tree-level will grow many clusters, with the effect that the overall algorithm will make less use of this particular feature for matching, and will rely on later feature modalities (deeper levels on the tree) to provide discrimination. In other words, the choice of T_k^m can actually be regarded as a measure of confidence in the discriminating ability of a particular feature.

Consequently, we would like to set high values of T_k^m for those features that are highly discriminatory in the current frame, and low values of T_k^m for poorly performing features (e.g., those modalities for which the target is camouflaged against the background in the present frame). We therefore propose a method by which T_k^m is continuously relearned for each feature at each frame, based on evaluating the feature's discriminating ability relative to the current contextual information. T_k^m is computed online as:

$$T_k^m = exp\{\frac{-\mathcal{B}^2(H_k^{F,m}, H_k^{B,m})}{\sigma_m^2}\} \tag{7}$$

where $H_k^{F,m}$ and $H_k^{B,m}$ are the top level foreground and background histograms defined in Eq. 2, and Eq. 7 is therefore a measure of similarity between the target and the local background (i.e. a measure of camouflage) in the respective feature modality. $\mathcal{B}(,)$ is the same similarity metric as in Eq. 5.

If any candidate superpixel is both i) distinct enough from others that it forms its own leaf and does not lie inside a cluster (e.g. S_6 or S_N in Fig. 2) and ii) strongly matches the target part, satisfying Eq. 5, then the decision tree ceases growing and the middle level target part is labeled as matching that candidate superpixel. Once all the middle level superpixels have been matched to new locations in the current image, their distribution is used to output a new bounding box position, and the top level target models are updated accordingly. Occasionally, some parts in the model will fail to find a strongly matching candidate superpixel, even after exhausting all possible image features (corresponding to all possible tree-levels). In such cases, it is inferred that the part is no longer visible and it is switched off. Other methods in the literature (e.g. [30]) remove unmatched patches, and thus permanently lose parts of the model during occlusions, which cannot later be recovered. In contrast, our proposed approach of temporarily switching off the unmatched parts provides a powerful tracking memory that automatically handles occlusion situations without the need for special additional occlusion routines.

4.3. Model updating

For robust tracking, it is necessary to continuously update the target model as it changes its appearance with time. The proposed tracker does this via two mechanisms: adapting the old bottom and top level target models, and adding new models of new middle level parts.

At each frame, we examine all middle level parts and detect those which are drifting (moving too far from the target centroid), with a method adapted from [35] which examines the marginal distributions of parts locations. To replace the drifting parts, we select those superpixels in the current frame which are most likely to represent target parts. A second kind of adaptive clustered decision tree is used to perform this selection as follows. We first use the primary feature histogram, $h_k^{i,m}$, to initially rank all unmatched superpixels according to:

$$Rank = sort(s_k^{i,m}, descend) \tag{8}$$

where $s_k^{i,m}$ is a priority score of ith candidate superpixel of feature m computed as:

$$s_k^{i,m} = \sum_{x=1}^{N_p^i} P_k^m(F|x)/N_p^i \tag{9}$$

where $P_k^m(F|x)$ is the likelihood, computed from Eq. 4, of the pixel at image location x, for all N_p^i pixels inside the ith candidate superpixel.

This ranked list can be regarded as the leaves on the first tree-level (first feature) of an adaptive clustered decision tree structure as described in the previous section for matching. Next, leaves (superpixels) with similar priority score are clustered when they satisfy the below constraint:

$$\|s_k^{i,m} - s_k^{j,m}\| < \lambda_{rank}\sigma_{all}^m \tag{10}$$

where λ_{rank} is a pre-defined parameter while σ_{all}^m is the standard deviation of all superpixels' expected likelihoods.

Once again, a cluster on a tree-level suggests that the feature modality for this tree-level is not sufficiently discriminating to achieve a robust ranking. Therefore, a secondary feature is chosen and used to rank all constituent superpixels within the cluster, forming a second tree-level on the tree. The tree is grown (by adding successive tree-levels, using successive features), until a unique ranking has been assigned to all superpixels. Then, the highest ranked n superpixels are chosen in order to initialise n new parts. These new parts replace the n old parts which were identified as drifting.

For the non-drifting parts (those that were matched strongly onto superpixels in the new frame) the target "parts" or "patch" models are updated according to new observations. Note that any kind of target model relearning is potentially dangerous, since even small tracking errors can easily cause background pixels to be learned into the target model, leading to instability with exponentially increasing errors. Early colour particle filter work [24] and recent state-of-the-art patch-based methods [30] perform model relearning at a fixed update speed. In contrast, we continuously recompute individual update speeds for each middle level part at each frame. Our premise is that parts can be relearned rapidly when there is a high confidence in their matching, whereas the relearning rate should be reduced under conditions of uncertainty. We therefore update each part, using a continuously relearned parameter $\mu_k^{i,m}$:

$$h_k^{i,m} = (1 - \mu_k^{i,m})h_{k-1}^{i,m} + \mu_k^{i,m} h_{obs}^{i,m} \tag{11}$$

$$\mu_k^{i,m} = \mathcal{B}(h_{k-1}^{i,m}, h_{obs}^{i,m}) \tag{12}$$

where $h_k^{i,m}$ is the m^{th} feature histogram of the i^{th} part at frame k, while $h_{obs}^{i,m}$ is the m^{th} feature histogram of the matched superpixel in the new frame. Again, $\mathcal{B}(.)$ is a similarity metric for the m^{th} feature as described in Eq. 5.

At each frame, once all middle level parts have been either switched off, updated, or replaced, then the top (global) level target model is updated according to Eq. 3, as described in Sec. 3.

4.4. Handling occlusions

The proposed tracker utilises a memory which memorises the latest tracker state, including all middle level part models. As described in Sec. 4.2, partial occlusion is handled by temporarily switching off poorly matching middle level parts, but retaining these in memory and reacquiring them once occluded target parts reappear in later video frames. If a large proportion of parts (defined by a threshold parameter) remains unmatched after the matching procedure of Sec. 4.2, then the tracker is regarded as being in a special state of full occlusion.

In the full occlusion state, all target model updating (at all model levels) is switched off, and the propagation scope (candidate image region surrounding the estimated target bounding box) is enlarged. The tracker returns to the normal (non-occlusion) state once a sufficient proportion of middle level parts are again matched strongly to candidate superpixels.

5. Experiments

We have tested the performance of our tracker on the sequences from the publicly available datasets VOT 2013, VOT 2014 and CVPR 2013 benchmark dataset [2], [34], which together comprise 70 sequences in total. More details of the datasets can be found from the webpages of [2] and [1].

5.1. Implementation

The proposed adaptive clustered decision tree structure is designed to handle, in principle, arbitrarily many features in a robust and efficient manner. For proof of principle, we have implemented a tree with just two tree-levels (exploiting just two feature modalities), however this already delivers strongly competitive performance on benchmark

Table 1. Values of key algorithmic parameters

Section	Equation	Value
Initialization	λ in Eq. 4	0.1
Decision tree	σ_m^2 in Eq. 7	0.05
	λ_{rank} in Eq. 10	0.1
Occlusion	the ratio of unmatched parts	40%

Table 2. VOT challenge results: comparing against best 4 trackers

	VOT 2013 (16 sequences)				
	Ours	PLT13 [15]	LGT++ [35]	EDFT [11]	FoT [31]
Fail.	0	0	1.53	14	22
Acc.	0.59	0.58	0.57	0.58	0.63
	VOT 2014 (25 sequences)				
	Ours	PLT14 [15]	DGT [7]	DSST [9]	SAMF [20]
Fail.	1	4	25	29	32
Acc.	0.52	0.56	0.58	0.62	0.61

test data. For both adaptive clustering tree structures (middle level parts matching Sec. 4.2, and new parts learning Sec. 4.3), we use simple colour histograms as a primary feature histogram, with the commonly used Bhattacharyya metric as a matching likelihood measure. For the secondary feature, we use a simple motion measure, where candidate superpixels are assigned high matching likelihoods if they imply a small frame-to-frame motion for the part being matched. The tracking algorithm has been implemented on an Intel Core i5-3570 CPU, using Matlab code (linked also to some C++ components). This unoptimised implementation, on an old PC, achieves near-to-real-time performance of 8 fps (varying somewhat with different test videos). The key parameters initialised in the first frame are listed in the Tab. 1.

5.2. Evaluation

We first evaluate our tracker using the ICCV2013 and ECCV2014 "VOT challenge" [2] testbeds. Tab. 2 compares the performance of our tracker against the best 4 VOT trackers, out of around 30 trackers that those challenges evaluated. The results are shown in terms of robustness (the total number of failure instances) and accuracy (percentage overlap between trackers output bounding box and ground truth bounding box) averaged over all frames.

For the robustness, our tracker has zero failures in VOT2013 and only one failure in VOT2014. The next best algorithm is PLT which also achieved zero failures in VOT2013. However, note that the version of PLT tested in VOT2013 used a fixed bounding box size. Therefore this algorithm was unable to adapt to targets which change their size during tracking. Since most objects in most test sequences luckily stayed roughly the same size, this rigid constraint helped the algorithm to achieve a high robustness score. For VOT2014, a different version of PLT was submitted, which did enable adaptation to changing target size. In this case, PLT's robustness worsened to four failures.

Note that the accuracy scores can sometimes be misleading. In the VOT testing methodology, ground truth is used to re-initialise trackers (with perfect accuracy) after every tracking failure. Therefore, trackers which fail very often will show high accuracy scores, even if they are not "good" trackers. Hence, the accuracy score is meaningful mainly when comparing two trackers which have the same robustness score. In VOT2013 our accuracy is better than the only other algorithm (PLT) which shares the same robustness, while in VOT2014 no other tracker was able to achieve the same robustness.

For more extensive comparison, we also combine the VOT test sequences with all those from the CVPR 2013 tracking benchmark data set [34]. Using this 70-sequence dataset, we compared our method against the publicly available trackers which have showed strong performance in either the VOT or the CVPR tracking challenges, namely: Struck [14], SCM [36], LGT++ [35], CSK [16], IVT [26], L1 [23], and PF [24]. Since this dataset contains instances of full occlusions, the evaluation is conducted without re-initialization after tracking failures. We show the results as trade-off curves as suggested by [34].

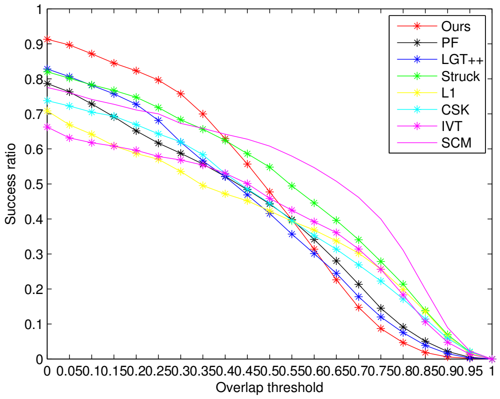

Figure 3. The success ratio versus overlap threshold curve in 70 sequences.

As shown in Fig. 3, our tracker achieves significantly better robustness in accuracy ranges up to 0.4. We assert that robustness is the most important of these metrics: firstly, these methods are intended for highly deformable targets (e.g. people) for which it is hard to meaningfully interpret the "accuracy" of a rectangular bounding box which includes many non-target pixels even during good performance; secondly, provided that a tracker is robust, accuracy can always be further improved by incorporating additional prior knowledge about a specific target [9].

To further evaluate the performance, we also show the trade-off curves for those test videos identified in the benchmark challenges as containing the categories of: significant target deformations, Fig. 4; severe illumination changes, Fig. 5; and occlusions, Fig. 6. Our tracker significantly outperforms the other methods in highly deformed targets tracking. We attribute to the flexibility of the clustered decision tree approach to online model relearning. The tracker also achieves competitive results in illumination change and occlusions scenarios. We attribute the strong performance under illumination changes to the robustness of the cross-constrained multi-level target model. We attribute the results of the occlusion tests to the generality and adaptability of the proposed method. When a method is designed to be robust against dramatic target appearance and shape changes, it may not always be possible to distinguish between appearance changes and occlusions, hence our method sacrifices some accuracy in favour of robustness in such circumstances.

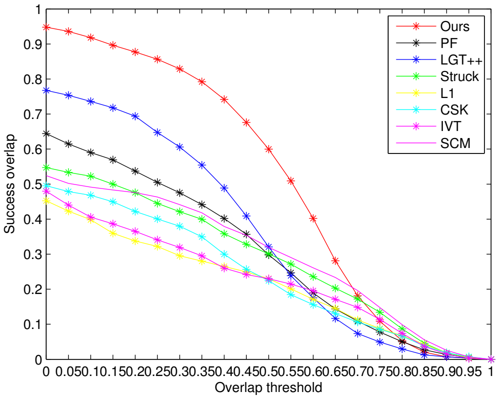

Figure 4. The success ratio versus overlap threshold curve in 19 sequences with deformation.

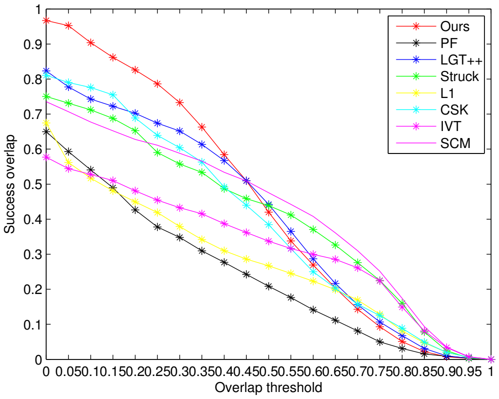

Figure 5. The success ratio versus overlap threshold curve in 18 sequences with illumination change.

Fig. 7 shows examples of our method handling vidoes which feature extremely deforming targets (e.g. gymnast) and very strong clutter (e.g. diver).

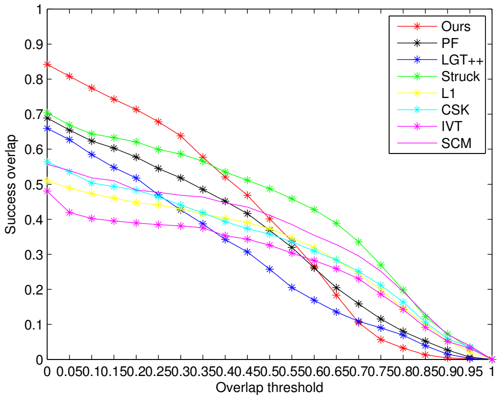

Figure 6. The success ratio versus overlap threshold curve in 22 sequences with occlusion.

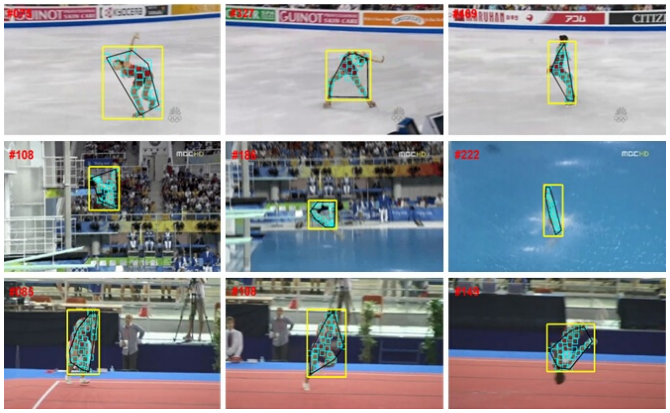

Figure 7. Visualization of results on sequences: *Iceskating*, *Diving*, *Gymnastics*, showing extreme target deformations and significant clutter.

6. Conclusion

In this paper, we have proposed a multi-level target model, and an adaptive clustered decision tree method for both matching and relearning middle level target parts at successive image frames. The resulting tracking algorithm significantly outperforms the best 4 algorithms from each of the VOT2013 and VOT2014 benchmark tracking challenges, and outperforms 7 state-of-the-art trackers within the additional CVPR2013 benchmark tracking data set. The method is particularly robust against challenging tracking conditions of large target deformation, rapid illumination changes, and occlusions. The adaptive decision tree: 1) is generated online, overcoming the overfitting of offline generated classifiers; 2) efficiently exploits only the minimum number of features needed for each target part at each frame; 3) adaptively weights in favour of the most discriminating features, responding dynamically to changing amounts of camouflage in different feature modalities. The future work will evaluate incorporating additional feature modalities, which is expected to improve the performance.

Acknowledgments

J. Xiao is supported by the China Scholarship Council and also by a University of Birmingham school scholarship. This work was supported in part by EU H2020 RoMaNS, 645582, and EPSRC EP/M026477/1.

References

[1] CVPR benchmark dataset. https://sites.google.com/site/trackerbenchmark/benchmarks/v10. 6

[2] The VOT challenge. http://www.votchallenge.net/. 3, 6, 7

[3] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Susstrunk. SLIC superpixels compared to state-of-the-art superpixel methods. *PAMI*, 34(11):2274–2282, 2012. 3, 4

[4] Boris Babenko, Ming-Hsuan Yang, and Serge Belongie. Robust object tracking with online multiple instance learning. *PAMI*, 33(8):1619–1632, 2011. 3

[5] Anil Kumar Bhattacharya On a measure of divergence between two statistical populations defined by their probability distributions. In *Bulletin of the Calcutta Mathematical Society, issue 35*, pages 99–110, 1943. 3

[6] Paul Brasnett, Lyudmila Mihaylova, David Bull, and Nishan Canagarajah. Sequential monte carlo tracking by fusing multiple cues in video sequences. *Image and Vision Computing*, 25(8):1217–1227, 2007. 2

[7] Zhaowei Cai, Longyin Wen, Jianwei Yang, Zhen Lei, and Stan Z Li. Structured visual tracking with dynamic graph. In *ACCV*, pages 86–97. Springer, 2013. 7

[8] Robert T Collins, Yanxi Liu, and Marius Leordeanu. Online selection of discriminative tracking features. *PAMI*, 27(10):1631–1643, 2005. 2

[9] Martin Danelljan, Gustav Häger, Fahad Shahbaz Khan, and Michael Felsberg. Accurate scale estimation for robust visual tracking. In *BMVC*, 2014. 7

[10] François Ennesser and Gérard Medioni. Finding waldo, or focus of attention using local color information. *PAMI*, 17(8):805–809, 1995. 2

[11] Michael Felsberg. Enhanced distribution field tracking using channel representations. In *ICCV visual object tracking workshop*, pages 121–128. IEEE, 2013. 7

[12] Helmut Grabner and Horst Bischof. On-line boosting and vision. In *CVPR*, volume 1, pages 260–267. IEEE, 2006. 3

[13] Helmut Grabner, Christian Leistner, and Horst Bischof. Semi-supervised on-line boosting for robust tracking. In *ECCV*, pages 234–247. Springer, 2008. 3

[14] Sam Hare, Amir Saffari, and Philip HS Torr. Struck: Structured output tracking with kernels. In *ICCV*, pages 263–270. IEEE, 2011. 7

[15] Cher Keng Heng, Sumio Yokomitsu, Yuichi Matsumoto, and Hajime Tamura. Shrink boost for selecting multi-lbp histogram features in object detection. In *CVPR*, pages 3250–3257. IEEE, 2012. 7

[16] Joao F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. Exploiting the circulant structure of tracking-by-detection with kernels. In *ECCV*, pages 702–715. Springer, 2012. 7

[17] Seunghoon Hong and Bohyung Han. Visual tracking by sampling tree-structured graphical models. In *ECCV*, pages 1–16. Springer, 2014. 2

[18] Xu Jia, Huchuan Lu, and Ming-Hsuan Yang. Visual tracking via adaptive structural local sparse appearance model. In *CVPR*, pages 1822–1829. IEEE, 2012. 2

[19] Junseok Kwon and Kyoung Mu Lee. Highly nonrigid object tracking via patch-based dynamic appearance modeling. *PAMI*, 35(10):2427–2441, 2013. 2

[20] Yang Li and Jianke Zhu. A scale adaptive kernel correlation filter tracker with feature integration. In *ECCV visual object tracking workshop*, pages 254–265. 2014. 7

[21] Yang Lu, Tianfu Wu, and Song-Chun Zhu. Online object tracking, learning and parsing with and-or graphs. In *CVPR*, pages 3462 – 3469. IEEE, 2014. 2

[22] Iain Matthews, Takahiro Ishikawa, and Simon Baker. The template update problem. *PAMI*, 26(6):810–815, 2004. 2

[23] Xue Mei and Haibin Ling. Robust visual tracking using l1 minimization. In *ICCV*, pages 1436–1443. IEEE, 2009. 7

[24] Katja Nummiaro, Esther Koller-Meier, and Luc Van Gool. An adaptive color-based particle filter. *Image and vision computing*, 21(1):99–110, 2003. 2, 3, 4, 6, 7

[25] Federico Pernici and Alberto Del Bimbo. Object tracking by oversampling local features. *PAMI*, 36:2538 – 2551, 2013. 2, 3

[26] David A Ross, Jongwoo Lim, Ruei-Sung Lin, and Ming-Hsuan Yang. Incremental learning for robust visual tracking. *IJCV*, 77(1-3):125–141, 2008. 7

[27] Rustam Stolkin, Ionut Florescu, and George Kamberov. An adaptive background model for camshift tracking with a moving camera. In *6th International Conference on Advances in Pattern Recognition*, pages 147–151. Citeseer, 2007. 2, 3

[28] Rustam Stolkin, David Rees, Mohammed Talha, and Ionut Florescu. Bayesian fusion of thermal and visible spectra camera data for region based tracking with rapid background adaptation. In *IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 192–199. IEEE, 2012. 2

[29] Rustam Stolkin and Mohammed Talha. Particle filter tracking of camouflaged targets by adaptive fusion of thermal and visible spectra camera data. *IEEE Sensors*, 2014. 2

[30] Luka Čehovin, Matej Kristan, and Aleš Leonardis. Robust visual tracking using an adaptive coupled-layer visual model. *PAMI*, pages 941 – 953, Apr 2013. 2, 3, 5, 6

[31] Tomáš Vojíř and Jiří Matas. Robustifying the flock of trackers. In *16th Computer Vision Winter Workshop*, pages 91–97, 2011. 7

[32] Qing Wang, Feng Chen, Wenli Xu, and Ming-Hsuan Yang. Online discriminative object tracking with local sparse representation. In *WACV*, pages 425–432. IEEE, 2012. 2

[33] Shu Wang, Huchuan Lu, Fan Yang, and Ming-Hsuan Yang. Superpixel tracking. In *ICCV*, pages 1323–1330. IEEE, 2011. 2

[34] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Online object tracking: A benchmark. In *CVPR*, pages 2411–2418. IEEE, 2013. 3, 6, 7

[35] Jingjing Xiao, Rustam Stolkin, and Aleš Leonardis. An enhanced adaptive coupled-layer LGTracker++. In *ICCV visual object tracking workshop*, volume 2, page 5. 2013. 6, 7

[36] Wei Zhong, Huchuan Lu, and Ming-Hsuan Yang. Robust object tracking via sparsity-based collaborative model. In *CVPR*, pages 1838–1845. IEEE, 2012. 7