Object Tracking by Oversampling Local Features

Federico Pernici and Alberto Del Bimbo

Abstract—In this paper, we present the ALIEN tracking method that exploits oversampling of local invariant representations to build a robust object/context discriminative classifier. To this end, we use multiple instances of scale invariant local features weakly aligned along the object template. This allows taking into account the 3D shape deviations from planarity and their interactions with shadows, occlusions and sensor quantization for which no invariant representations can be defined. A non parametric learning algorithm based on the transitive matching property discriminates the object from the context and prevents improper object template updating during occlusion. We show that our learning rule has asymptotic stability under mild conditions and confirms the drift-free capability of the method in long term tracking. A real-time implementation of the ALIEN tracker has been evaluated in comparison with the state of the art tracking systems on an extensive set of publicly available video sequences that represent most of the critical conditions occurring in real tracking environments. We have reported superior or equal performance in most of the cases and verified tracking with no drift in very long video sequences.

Index Terms—Visual real-time tracking, long-term tracking, learning from video, local feature invariance, template update.

1 Introduction

Tracking is a fundamental problem in computer vision and has a wide range of application. It is a prerequisite to higher level tasks such as area monitoring, target recognition, trajectory interpretation, behavior analysis and reasoning. Effective methods must provide robust object representation capable to cope with *nuisance factors* that affect the image formation process, i.e. all the information that is not of direct interest for tracking but which needs nevertheless be taken into account. Such factors include invertible nuisances such as contrast and viewpoint variations, as well as non-invertible ones such as occlusions, sensor quantization and general illumination changes [53], [54], [52]. Can these factors be removed so that the only remaining information is that needed to track an object in the scene? This question dates back to J. Gibson [13], who claimed that the crucial information for perception is the information that remains invariant as an observer moves through the environment. Hence, at least for invertible nuisances one can construct *invariant features* that act as a representation for decision tasks that would contain all and only the information that matter to the task.

Unfortunately a similar solution does not exist for non-invertible nuisances. However, when some form of active control of the sensing process can be carried out, then some non-invertible nuisances may become invertible too. Occlusions, for example, can be inverted by moving the camera around the occluder. Scaling/sensor-quantization can be inverted by moving the camera closer to the object. When no active control can be applied, invertibility can only be obtained for planar or locally planar scene structures [53]. A well known case of that is given by the SIFT detector based on Difference of Gaussian by [35]. Instead, if the visual structure detected originates from some phenomena in the scene or from artifacts of the image formation process (for instance a keypoint detected at an occluding contour), invertibility depends on whether the signal is properly sampled.

Our solution for tracking presented in this paper is inspired by these general observations and grounds on solutions that attempt to neutralize both invertible and non invertible nuisance factors that may affect the image formation process. According to this, while on the one hand it assumes local invariant features for object representation, on the other hand, 3D shape deviations from planarity and their interactions with shadows, occlusions and sensor quantization are managed by taking multiple instances of the same features in different conditions, i.e. *oversampling*, after a weak alignment along the object template.

Shape and appearance variations are hence captured by a non parametric classifier that uses the instances of the features to model the complex appearance manifold topology originated by visual artifacts for which no invariants can be computed (Fig 1). Two additional classifiers are used respectively to achieve higher discrimination between the object and its surrounding context and prevent improper object template updating during occlusions.

We will refer to this solution as ALIEN, where the acronym stands for Appearance Learning In Evidential Nuisance. It will be proved that this solution is able to perform continuous tracking even under severe visibility artifacts or critical conditions such as occlusions and shadowing. We will show that our learning rule is asymptotically stable, confirming the

• *F. Pernici is with the Media Integration and Comunication Center (MICC), University Of Florence, Florence, IT, 50134.*
E-mail: see http://www.micc.unifi.it/pernici/

(a) (b)

Fig. 1: Weak alignment of multiple instances of local features. (a): Four frames from the *trellis* sequence [45] with appearance variations in a particular object region (highlighted) due to non-invertible nuisances (self-occlusions and shadows). (b): Region representation after weak alignment. Feature locations describing the 2D shape in the coordinate system of the object template and the appearance descriptors associated (128D).

drift-free capability to the method proposed. We will present an extensive set of experiments on publicly available sequences, showing the superiority of the method with respect to the state of the art.

In the following we review the state of the art of tracking and highlight the distinguishing aspects of the ALIEN tracker with respect to the other approaches. The proposed algorithm is then summarized in Section 2 and detailed in Section 3. In Section 4 we present an extensive comparison with the state of art tracking methods that were published in the literature. Conclusions are finally given in Section 5.

1.1 Related work

Recent surveys of the most notable methods of tracking were published in [33], [69] and comparisons of performances under different conditions appeared in [62], [48], [67]. The very many methods that have been published in the literature differ from each other in the solutions adopted for the object appearance representation, the object shape transformation admitted, whether a motion model is used and whether the learning model is generative or discriminative [62]. Among these methods, a few of them have been recognized for either their performance or their design originality, namely the the CoGD [71], L1 [38], the Predator-TLD [25], the PROST [49], the BLUT [68], the MIL [4], the ConTra [9], the MTT [75], [76] and the LSHT [22]. Many other trackers have been inspired by these solutions with minor distinguishing characteristics.

Different object representations were employed. In the L1 tracker, the authors used the best sparse subspace representation of pixel intensities [65]. Pixel intensities were combined with trivial templates that account for the presence/absence of image patches of the object candidate. Although templates explicitly consider occlusions, the pixel-wise holistic representation is sensitive to partial occlusions and motion blur [63]. These two limitations were addressed respectively in [68] and [39], [27], by adding heuristics to the sparse representation framework. Many other trackers, among them MTT and those in [32], [34], [24] and [77], have adopted the sparse representation. A specific review and comparison on sparse representation methods was published in [74]. The Predator-TLD tracker used Local Binary Patterns of pixel intensities to represent the object appearance. A similar representation was also used in the ConTra method. In the MIL tracker, the authors used Haar-like wavelets as in [61]. The same representation was used in PROST. The LSHT tracker used a locality sensitive intensity histogram computed at each pixel location. Although in all these cases features are local, their evaluation is generally made globally along the object template.

In MIL, Predator-TLD, CoGD, ConTra, ODF [8], ET [3], [41], [36], [71], [73], [18] and many other solutions, context features were used to improve the distinction between the object and its background. A critical problem concerns the accuracy with which foreground image regions are separated from background regions. In most of the literature, an axis aligned bounding box is used although this is likely to treat background regions as part of the foreground. This produces a gradual degradation in object appearance which results in an irreversible drift of the template model. Recently, object segmentation was used to limit this phenomenon [64], [14], but this requires that a significant object segment is visible.

Tracking by detection paradigm was used in most of the tracking systems, namely in MIL, Predator-TLD, CoGD, PROST, ConTra, LSHT, OB[15], ORF[47], FT[1], ODF [8], ET [3], OAB [16], SB [17], BS [56], Struck [20], GTK [51], CoTT [70], DNBS [31], SPT[64], CT [73], BHT [50], DFT [40] and many others. Recursive filtering was employed instead by L1, MTT, BLUT, IVT [45], VTD [28], LGT [7], CT [73], hsvPF [43] and several others. With respect to recursive filtering, the advantage of tracking by detection is in the resilience of the underlying representation of appearance, making it easier recovering from occlusions. It is anyway required that the objects have sufficient visual information.

Template updating is fundamental to maintain a complete object model that makes the tracker capable of resist to factors that might corrupt the object rep-

resentation as well as to support long term tracking with no drifting [37]. In MIL and other trackers like the OB, OAB, SB and BS trackers, template updating is performed by an evolving boosting classifier that tracks image patches and learns the object appearance. However, online boosting requires that data are independent and identically distributed (i.i.d.) [60] which is a condition not satisfied in video sequences, where data is temporally correlated [72]. In the Predator-TLD and the PROST methods the authors combined an optic flow tracker with an online learned random forest. New training samples are collected when detections violate constraints on the object position estimated. In order to control drifting, the new data is not incorporated into the template until a previously confirmed appearance is not retrieved with high confidence. A similar solution was followed in ConTra, where the authors introduced some mechanisms to improve the capability to distinguish between objects of similar appearance. To the best of our knowledge Predator-TLD and ConTra are the only methods capable of effective tracking in long sequences (until 10000 frames) with little drifting. No drifting was also reported recently in [57], in a sequence of about 2600 frames. Object appearance was learned from the selection of trustworthy frames, using HOG features and an SVM classifier. In the L1 tracker, appearance update is performed by keeping the most recent stable templates.

In the ALIEN tracker we adopt a local representation based on SIFT [35] and evaluate object appearance similarity globally in the object template, at distinct and independent stages. Initially, a classifier discriminates between object and context. It exploits bag of features to detect local object appearance similarity according to transitive matching (explained in Subsection 3.1); object features that have been matched are hence voted according to a global shape model based on a similarity transformation. With this approach, two matched features are in principle sufficient to instantiate a global transformation. In the case in which these are spatially close to each other, as for example in the case in which most of the object is occluded, this permits the exploitation of the sole local similarity. On the other hand, when most of the object is visible the global shape model is used. As will be explained in Subsection 3.2, the similarity transformation is combined with a RANSAC-like voting scheme, so accounting also for shape generalization, rather than simply voting to a specific geometric model.

We adopt the tracking by detection paradigm. However, differently from most of the systems, in ALIEN tracking is performed in both scale and rotation space, using an object-oriented bounding box.

Template updating as performed in ALIEN is substantially different from the other approaches. Outliers are maintained in the object template and feature oversampling is used as a potential revealer of novel object structures. Differently from [4], we explicitly consider that the data distribution is time varying and keeps constant only between two consecutive frames. The object template is updated frame by frame in a way that the data distribution is not corrupted and features are removed by uniform sampling. This approach of incremental appearance learning is asymptotically stable and permits tracking with no drifting also in very long sequences.

The voting scheme of our tracker has relation with the Hough voting of local features of [29] and [12]. The works in [18] and [36] have some similarities with our method. They too perform template updating, but reject the outliers. Instead, in ALIEN all the features are retained in the template to cope for non invertible nuisance or potential novel object structures. As in ALIEN, they both use bag of features and discriminate the object with respect to context. Segmentation of superpixel regions is used in [36] to distinguish between object and background. However, in these methods object shape is less considered and the bag of features representation is the most important part of the model.

1.2 Contributions and Improvements

The main contributions of the paper are:

- A novel object representation based on the *weakly aligned multi-instance local features*. We demonstrate that this representation improves on the inherent limit of local features invariance under occlusion, sensor quantization and casting shadow.
- A novel non parametric learning algorithm based on the *transitive property of the matching relationship* between the object and its surrounding context. This property allows building a strong discriminative classifier which also enables to detect occlusions before updating the template, so avoiding improper appearance contamination.

The algorithm *asympthotic stability* is established using the Multiplicative Ergodic Theorem [42]. This confirms a drift-free capability of our learning procedure.

The method has superior performance with respect to the state of the art tracking methods and is capable of continuous tracking for long periods also in the presence of severe occlusions, in- and out-of-plane-rotations and scale variations, and at the same time is insensitive to blurring. The performance of the solution is demonstrated in Section 4 on several publicly available datasets.

2 Method Overview

In this section, we summarize the principal components of our tracking system and their functional interrelationships.

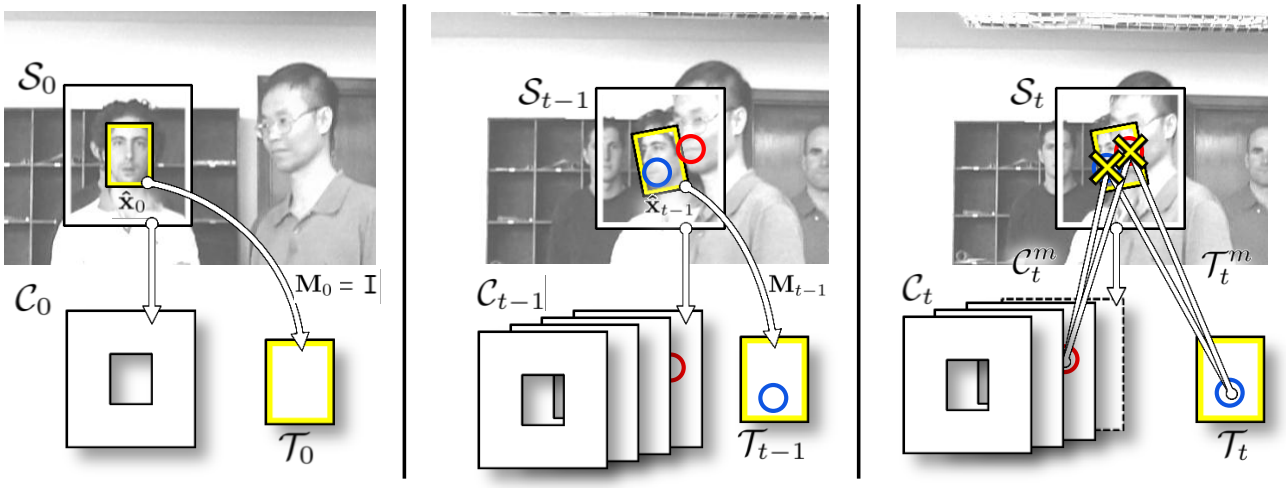


Fig. 2: Overview of tracking by transitive matching with context. *Left*: The object under tracking at time $t = 0$ and the elements involved in our method: object template, context and search area respectively $\mathcal{T}_0$, $\mathcal{C}_0$ and $\mathcal{S}_0$. *Middle*: The object is going to be occluded by another object with similar imaged features (indicated with circles). The contextual space-time region $\mathcal{C}_{t-1}$ captures some of these features which may imminently enter in the bounding box of the object under tracking (only one is shown for clarity). *Right*: Two features (indicated with crosses) extracted from the search area match both with context $\mathcal{C}_t$ and object template $\mathcal{T}_t$ and are therefore classified as not distinctive. This is a particular combined case in which a feature is both an occluding and a non distinctive feature.

Given a bounding box defining an object of interest, discrimination between the object and the rest of the scene is obtained by using two distinct non-parametric Nearest Neighbor classifiers, accounting respectively for the object under tracking and its context. The object classifier $\mathcal{T}_t$ accounts for the object shape and appearance at time t:

$$\mathcal{T}_t = \{(\mathbf{p}_i, \mathbf{d}_i)\}_{i=1}^{N_\mathcal{T}}, \tag{1}$$

where $\mathbf{p} \in \mathbb{R}^2$ is the location of a keypoint referred to the coordinate system of the object template, $\mathbf{d} \in \mathbb{R}^n$ is the appearance descriptor associated (see Fig. 1(b)) and $N_\mathcal{T}$ is the number of object features.

The context classifier $\mathcal{C}_t$ accounts for the appearance of the spatio-temporal context surrounding the object and exploits the standard bag of features representation:

$$\mathcal{C}_t = \{\mathbf{d}_i\}_{i=1}^{N_\mathcal{C}}, \tag{2}$$

where $\mathbf{d} \in \mathbb{R}^d$ is the visual descriptor of a keypoint in the context region and $N_\mathcal{C}$ is the number of features. In the present implementation we have used SIFT features [35], although any scale invariant representation as [5], [6], [46], [30], [2] can be plugged in.

Object tracking (detailed in Subsection **??**) is obtained from the tight interplay between the two classifiers $\mathcal{T}_t$, $\mathcal{C}_t$ and the state of the tracked object $\mathbf{x}_t$. The state $\mathbf{x}_t$ at time t includes the object center location (x_t, y_t), scale s_t and rotation angle θ_t with respect to the bounding box provided at time $t = 0$, and implicitly defines an object oriented bounding box (OBB($\mathbf{x}_t$)). Consequently, it also defines the 2D similarity transformation:

$$\mathbf{M}(\mathbf{p}; \mathbf{x}_t) \iff \mathbf{x}_t = (x_t, y_t, s_t, \theta_t), \tag{3}$$

where $\mathbf{p} \in \mathbb{R}^2$ is the location of a keypoint, $\mathbf{M} : \mathbb{R}^2 \mapsto \mathbb{R}^2$ maps the object detected in the image into the coordinate system of the object template and $\mathbf{x}_t = (x_t, y_t, s_t, \theta_t)$ is the vector of the transformation parameters to be estimated.

Similarly to [4], we do not maintain a distribution of the object state at every frame and assume that, at time t, the tracked object is equally likely to appear within a radius r of the tracker state estimated at time $t - 1$ (in both location and scale):

$$p(\hat{\mathbf{x}}_t | \hat{\mathbf{x}}_{t-1}) = \begin{cases} 1 & \text{if} \quad ||\hat{\mathbf{x}}_t - \hat{\mathbf{x}}_{t-1}||_\infty < r \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

Object detection returns the tracker state and a probability $p(y = 1|\mathcal{S}_t)$ where:

$$\mathcal{S}_t = \{(\mathbf{p}_i, \mathbf{d}_i)\}_{i=1}^{N_\mathcal{S}} \tag{5}$$

is the set of features extracted from an image search area $\mathbf{S}_t$ and y is a binary variable indicating the presence or the absence of the object of interest in that image region. The detector response is evaluated with a greedy strategy (detailed in Subsection 3.2.1).

If the object remains undetected for a certain number of frames n_u, random search is applied (only horizontal and vertical shifts should be generated since SIFT features are invariant to scale variations).

Once the tracker state is estimated, an additional detector (detailed in Subsection 3.2.2) checks if the object is occluded. If not, both the object and the context appearance models are updated. All the local features inside the OBB($\hat{\mathbf{x}}_t$) region are labeled as object features. Instead, the features of the annular region surrounding the object are accumulated over a time window of length l:

$$\mathcal{C}_t = \bigcup_{\tau=t-l}^{t} \{(\mathbf{p}, \mathbf{d}) \,|\, \mathbf{p} \in \mathbf{A}_\tau\}, \tag{6}$$

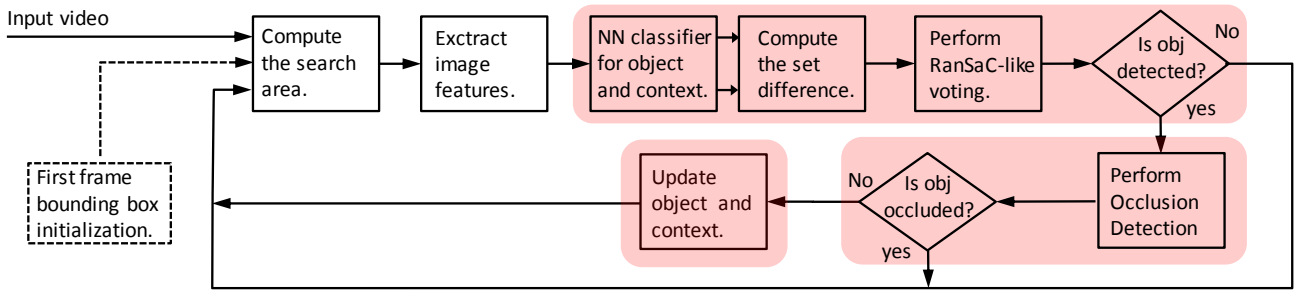


Fig. 3: Block diagram presenting the major work flow and functional component in the proposed tracking algorithm. Shaded area highlights the three components discussed in the paper.

where $\mathbf{A}_t = \mathbf{S}_t \setminus \text{OBB}(\hat{\mathbf{x}}_t)$, and are labeled as context features. Fig. 2-*left* and *middle* illustrates some of the entities described in this section.

3 Tracking Algorithm

In this section, we expose object detection and appearance learning of the ALIEN tracking algorithm. The block diagram of the system and the detailed presentation of the processing steps are shown respectively in Fig. 3 and Alg. 1. In the following subsections, we will discuss the way in which ALIEN discriminates between object and context (Subsection 3.1), how an updated version of the object appearance is continuously learned and the way in which we detect and manage occlusions in order to avoid that spurious features contaminate the object template (Subsection 3.2). In Subsection 3.3 we discuss the asymptotic stability of the learning algorithm.

3.1 Transitive Matching with Context

In order to distinguish between object and context features, we both consider object appearance features and their arrangement with the context. To this aim we exploit the *transitive property*, of the matching relationship (indicated with $\sim$) between the search area feature set $\mathcal{S}_t$, the space-time context feature set $\mathcal{C}_t$ and the object feature set $\mathcal{T}_t$:

$$(\mathcal{T}_t \sim \mathcal{S}_t) \wedge (\mathcal{C}_t \sim \mathcal{S}_t) \Longrightarrow \mathcal{T}_t \sim \mathcal{C}_t. \tag{7}$$

Eq. 7 states that if a feature in $\mathcal{S}_t$ matches both with a feature in $\mathcal{T}_t$ and with a feature in $\mathcal{C}_t$, then this feature may not be sufficiently distinctive in order to discriminate between the object and its context. According to this, all the features satisfying Eq. 7 are removed by performing set-wise difference:

$$\mathcal{F}_t = \mathcal{T}_t^\star \setminus \mathcal{C}_t^\star, \tag{8}$$

where $\mathcal{T}_t^\star$ collects the matching indexes between $\mathcal{T}_t$ and $\mathcal{S}_t$, and $\mathcal{C}_t^\star$ collects the matching indexes between $\mathcal{C}_t$ and $\mathcal{S}_t$, obtained by a Nearest Neighbour search according to the distance ratio criterion of [35] (Alg. 1 lines 3-6). As a byproduct, this strategy also removes the background features which might be included in the object bounding box. In this case, these features both exist in the object and context feature sets and therefore are managed in the same way as the other ambiguous features.

3.2 Weakly Aligned Multiple-Instance Learning of Local Features

3.2.1 *Object state estimation*

The features in $\mathcal{F}_t$ of Eq. 8 are sampled according to a greedy strategy to estimate the object state $\mathbf{x}_t$. The locations of the features are voted with a similarity transformation model $\mathbf{M}(\mathbf{p}; \mathbf{x}_t)$ instantiated at each iteration from two correspondences. Voting is performed according to the MLESAC (Maximum Likelihood SAC) loss function that, differently from RANSAC, has a probabilistic formulation that provides a soft criterion in the evaluation of the error [59]. According to this, the tracker state $\hat{\mathbf{x}}_t$ is estimated as:

$$\hat{\mathbf{x}}_t = \underset{x,y,\theta,s}{\text{argmin}} \Big\{ \sum_{f \in \mathcal{F}_t} L(e_f; \mathbf{M}) \Big\}, \tag{9}$$

where:

$$L(e_f; \mathbf{M}) = -\ln p(e_f|\mathbf{M}), \tag{10}$$

is the MLESAC loss funtion with:

$$p(e_f|\mathbf{M}) = \gamma \frac{1}{\sqrt{2\pi\sigma}} \exp\left(\frac{-2e_f^2}{2\sigma^2}\right) + (1-\gamma)\frac{1}{\nu} \tag{11}$$

where σ is the standard deviation of the keypoint detector, γ is the expected proportion of inliers and ν accounts for some knowledge of the outliers distribution. The quantity e_f is the symmetric transfer error of the matching feature f between the current frame and the object template, computed as:

$$e_f = d(\mathbf{p}, \mathtt{M}^{-1}\mathbf{p}')^2 + d(\mathbf{p}', \mathtt{M}\mathbf{p})^2, \tag{12}$$

being $\mathbf{p}$ and $\mathbf{p}'$ the point locations of a matched feature $f \in \mathcal{F}_t$ in the image and object template respectively, and $\mathtt{M}$ the 3×3 homogeneous matrix representing the similarity transformation $\mathbf{M}(\mathbf{p}; \hat{\mathbf{x}}_t)$. The first term in this equation is the transfer error in the current image and the second term is the transfer error in the object template.

In the evaluation of Eq. 10, the number of iterations is fixed instead of being adapted to a probabilistic confidence level. According to this, object model overfitting is avoided and the local features are only weakly aligned to the object template. As a result, when performing template update this will provide the oversampling mechanism (discussed in

Algorithm 1: ALIEN tracking algorithm.

Input: Initial object bounding box $\mathbf{x}_0$,
Output: Estimated Object State $\hat{\mathbf{x}}_t = (\hat{x}_t, \hat{y}_t, \hat{\theta}_t, \hat{s}_t)$, object shape and appearance $\mathcal{T}_t$ and object context appearance $\mathcal{C}_t$.

1 **repeat**
2 Crop out the search region $\mathbf{S}_t = \{\mathbf{x} \in \mathbb{R}^4 : ||\mathbf{x} - \hat{\mathbf{x}}_{t-1}||_\infty < r\}$ and extract local features $\mathcal{S}_t = \{(\mathbf{p}, \mathbf{d}) | \mathbf{p} \in \mathbf{S}_t, \mathbf{d} \in \mathbb{R}^d\}$.
3 // Compute matching features $\mathcal{T}_t^\star = \{(\mathbf{p}, \mathbf{d}) \in \mathcal{S}_t \mid \frac{||\mathbf{d} - 1NN_{\mathcal{T}}(\mathbf{d})||}{||\mathbf{d} - 2NN_{\mathcal{T}}(\mathbf{d})||} < \lambda_{\mathcal{T}}\}$;
4 $\mathcal{C}_t^\star = \{(\mathbf{p}, \mathbf{d}) \in \mathcal{S}_t \mid \frac{||\mathbf{d} - 1NN_{\mathcal{C}}(\mathbf{d})||}{||\mathbf{d} - 2NN_{\mathcal{C}}(\mathbf{d})||} < \lambda_{\mathcal{C}}\}$;
5 // Transitive matching with context
6 $\mathcal{F}_t = \mathcal{T}_t^\star \setminus \mathcal{C}_t^\star$;
7 $(\hat{x}_t, \hat{y}_t, \hat{\theta}_t, \hat{s}_t) = \underset{x,y,\theta,s}{\operatorname{argmin}} \{ \sum_{f \in \mathcal{F}_t} L(e_f; \mathbf{M}) \}$;
8 **if** $(|\hat{s}_t - \hat{s}_{t-1}| < k_s)$ **and** $(|\hat{\theta}_t - \hat{\theta}_{t-1}| < k_\theta)$ **then**
9 // object detected
10 $\hat{\mathbf{x}}_t = (\hat{x}_t, \hat{y}_t, \hat{\theta}_t, \hat{s}_t)$;
11 $\mathcal{E}_t = \{(\mathbf{p}, \mathbf{d}) \in \mathcal{S}_t \mid \mathbf{p} \in \text{OBB}(\hat{\mathbf{x}}_t)\}$;
12 $\mathcal{O}_t = \{(\mathbf{p}, \mathbf{d}) \in \mathcal{C}_t^\star \mid \mathbf{p} \in \text{OBB}(\hat{\mathbf{x}}_t)\}$;
13 // Occlusion detection
14 **if** $|\mathcal{O}_t| \leq N_{\mathcal{O}}$ **then**
15 // Object non-occluded
16 // Object appearance update $\mathcal{E}_t' = \{(\mathbf{p}', \mathbf{d}) \mid (\mathbf{p}, \mathbf{d}) \in \mathcal{E}_t,\ \mathbf{p}' = \mathbf{M}(\mathbf{p}; \hat{\mathbf{x}}_t)\}$;
17 $\mathcal{T}_t = \mathcal{T}_{t-1} \bigcup \mathcal{E}_t'$;
18 // Context appearance update
19 $\mathcal{D}_t = \mathcal{D}_{t-1} \bigcup \mathcal{O}_t$;
20 $\mathcal{C}_t = \Big(\bigcup_{\tau=t-l}^{t} (\mathcal{S}_\tau \setminus \mathcal{E}_\tau) \Big) \bigcup \mathcal{D}_t$;
21 // Uniform random sampling forgetting features
22 **if** $|\mathcal{T}_t| > N_{\mathcal{T}}$ **then**
23 RandSamp($\mathcal{T}_t$, $N_{\mathcal{T}}$) ;
24 **end**
25 **if** $|\mathcal{D}_t| > N_{\mathcal{D}}$ **then**
26 RandSamp($\mathcal{D}_t$, $N_{\mathcal{D}}$) ;
27 **end**
28 **end**
29 **end**
30 **until** *True*;

Subsection 3.2.3). The detector response is evaluated considering the state parameters $\hat{s}_t$ and $\hat{\theta}_t$ as:

$$p(y = 1|\mathcal{S}_t) = \begin{cases} 1 & \text{if } |\hat{s}_t - \hat{s}_{t-1}| < k_s, |\hat{\theta}_t - \hat{\theta}_{t-1}| < k_\theta, \\ 0 & \text{otherwise.} \end{cases} \tag{13}$$

where k_s and k_θ are two predefined constants which control the maximum allowed speed in scale and rotation between two consecutive frames (Alg. 1 line 8). These two constraints reject false positives due to unreal axial or central symmetry modeled in the $\mathbf{M}$ transformation. It is important to note that, since tracking is performed in scale and rotation space, the image features that are extracted from the detected object are better aligned to those in the object template, so making it easier for the classifier to learn the correct object shape and appearance.

3.2.2 Occlusion Detection

An additional classifier is used in order to prevent improper updates in the presence of persistent occlusions. We consider the set features in $\mathcal{C}_t^\star$ that are also included in the OBB($\hat{\mathbf{x}}_t$) region:

$$\mathcal{O}_t = \{(\mathbf{p}, \mathbf{d}) \in \mathcal{C}_t^\star \mid \mathbf{p} \in \text{OBB}(\hat{\mathbf{x}}_t)\}. \tag{14}$$

The features in $\mathcal{O}_t$ may be: object/context ambiguous features, object/context boundary features, and features of occluding objects. For the first two classes, the cardinality of the features can be approximatively assumed equal to zero, on average. The number of visual features in the third class is instead higher than zero since occluding objects typically consist of a large number of connected regions. Following these considerations, detection of occlusions can be reduced to checking the cardinality $|\mathcal{O}_t|$ of $\mathcal{O}_t$ in Eq. 14 by thresholding (Alg. 1 line 13-14). If an occlusion is detected the object template is not updated. Since the voting strategy is partially robust to outliers (at least for the time interval necessary to remove all the features that may provide correct matches in the template) the specific value of the threshold is not critical for the system performance.

3.2.3 Object/context appearance update

The transformation $\mathbf{M}(\mathbf{p}; \hat{\mathbf{x}}_t)$ naturally accounts for global in-plane rotations and scale variations of the object. Appearance variations due to out-of-plane rotations are accounted by combining the template representation with the information provided by $\mathbf{M}(\mathbf{p}; \hat{\mathbf{x}}_t)$ and the new object evidence in the current image at time t.

All the features in the search area $\mathbf{S}_t$ that are contained in OBB($\hat{\mathbf{x}}_t$) (the set $\mathcal{E}_t$ that represents the new object evidence) are labeled as object features. Their locations $\mathbf{p}'$ in the template are computed as (Alg. 1 line 16):

$$\mathbf{p}' = \mathbf{M}(\mathbf{p}; \hat{\mathbf{x}}_t). \tag{15}$$

This oversampling strategy provides an over-representation of the object with both matched and unmatched features being included in the template. The unmatched features play an important role in the ALIEN tracking system. In fact, they may account for template-aligned samples due to new visual structure determined by out-of-plane rotations or for non-invertible nuisances that cannot be captured by the SIFT invariance. This determines a sort of generalization of the object shape and appearance.

Context must be updated as well. Tracking by transitive matching with context may fail when an occlusion persists for a number of frames greater than the temporal window l used to accumulate the context features (Eq. 6). In this case, the set-wise removal in Eq. 8 is no more effective and the occlusion cannot be detected, as features are discarded after that l frames are elapsed. This effect may be largely reduced by accumulating the features $\mathcal{O}_t$ to the context $\mathcal{C}_t$. In this way, features that are not distinctive for the object are regarded as *hard negatives* and are collected indefinitely (see Alg. 1 line

18–20). Fig. 2 *Middle* and *Right* show an example of this process and its effects.

3.2.4 *Forgetting Features*

In order to avoid an unbounded growth in the number of features, we adopt a strategy to remove features in the object template. This is based on the assumption that the generated samples of the video sequence are strongly correlated and therefore non i.i.d.. According to this, the generating data distribution is considered as slowly varying and stationary only between consecutive frames. For each frame, when the number of features in the object classifier exceeds the number of features allowed $N_{\mathcal{T}}$ (determined by memory or speed constraints), features are removed from the object template according to uniform random sampling so that the current distribution is not corrupted (Alg. 1 lines 22 and 23). Similarly, when the number of the accumulated occlusion features $\mathcal{D}_t$ in the context exceeds some threshold $N_{\mathcal{D}}$, uniform random sampling for forgetting features in the object context is used (Alg. 1 lines 25 and 26).

3.3 Algorithm Convergence Analysis

As reported in Eq. 15, at each time t, the object template is updated by adding novel feature points. Their new locations are computed using the estimated similarity transformation. Slight imprecisions in the estimation of the similarity transformation might accumulate over time and result in some drifting of the feature point locations. To understand under what conditions this might occur, a stability study is required.

Assumption 1: Because of the temporal coherence between consecutive frames, there is a high probability that the features used to compute the transformation $\mathbf{M}_t$ between the OBB($\mathbf{x}_t$) and the object template at time t are the same as those in the OBB($\mathbf{x}_{t-1}$) computed at time $t-1$. □

According to this, the transformation matrix $\mathbf{M}_t$ can be approximatively computed by the product of frame-to-frame similarity matrices $\{\mathbf{M}'_i\}_{i=0}^{t}$ from $t=0$ until t. In homogeneous coordinate notation:

$$\mathtt{M}_t \approx \mathtt{M}_0\mathtt{M}'_1 \cdots \mathtt{M}'_{t-1}\mathtt{M}'_t. \tag{16}$$

Fig. 4 shows the relationships between the entities involved in Eq. 16. Elements in the matrices $\mathtt{M}'$ of Eq. 16 are random variables because they are computed from measurements trough an estimation process. Therefore their variation depends on both the uncertainty in keypoint localization and their spatial layout [19]. According to [58], [55], [44], during the initial iterations of the evaluation of Eq. 9, the uncertainty due to keypoints spatial layout dominates. As the estimate converges and new correspondences are established, parameters uncertainty due to keypoints spatial layout reduces significantly and the uncertainty in keypoint localization comes to play a major

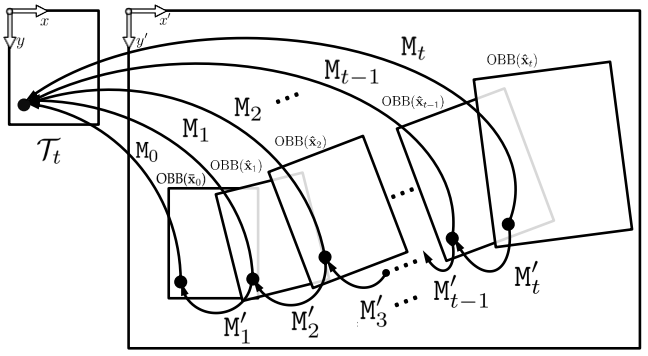


Fig. 4: Object template $\mathcal{T}_t$ and OBBs relationships for a generic trajectory of the object under tracking. Both the template and image coordinate systems are indicated. The transformation $\mathtt{M}_t$ at time t can be expressed as the composition (matrix product) of frame-to-frame transformations at previous time instants.

role. In this case, the variations of the parameters of the similarity transformation can be considered as being subject *only* to errors determined by keypoint localization. This assumes a linearized approximation of the error model in which a first-order model has proved to be sufficient [21].

Assumption 2: According to the considerations above, the estimated state $\hat{\mathbf{x}}_t = (x_t, y_t, s_t, \theta_t)$ can be reasonably assumed, with no loss of generality, to be corrupted by an additive white Gaussian noise $\xi_t^{(\cdot)} \sim \mathcal{N}(0, \sigma_{(\cdot)})$ around a reference value $\bar{\mathbf{x}}_0$ as provided by the initial bounding box:

$$\hat{\mathbf{x}}_t = \bar{\mathbf{x}}_0 + \xi_t = (0,0,1,0) + (\xi_t^x, \xi_t^y, \xi_t^s, \xi_t^\theta). \quad \square \tag{17}$$

The unmatched features violate this normal noise distribution model, but since they are identified before the final estimation is applied, they can be neglected in the convergence analysis. Eq. 16 can be viewed as the infinite product of "perturbed identity matrices" which results after that the detected object features at time t are transformed onto the template coordinate system according to the frame-to-frame similarities.

Asymptotic Stability: According to the Multiplicative Ergodic Theorem [11], [42], it holds:

$$\lim_{t\to\infty} \tfrac{1}{t} \log ||\mathtt{M}_0\mathtt{M}'_1 \cdots \mathtt{M}'_{t-1}\mathtt{M}'_t|| = \gamma \quad \text{with probability 1,} \tag{18}$$

where γ are the Lyapunov Exponents.

Application of transformation $\mathtt{M}$ to the noise component of Eq. 17 can be written in matrix form using homogeneous coordinates as:

$$\mathtt{M}(\xi_t) = \mathtt{T}(\xi_t^x, \xi_t^y)\mathtt{S}(1+\xi_t^s)\mathtt{R}(\xi_t^\theta)\mathtt{T}^{-1}(\xi_t^x, \xi_t^y), \tag{19}$$

where $\mathtt{T}(\xi_t^x, \xi_t^y)$, $\mathtt{S}(1+\xi_t^s)$ and $\mathtt{R}(\xi_t^\theta)$ are 3×3 perturbed matrices of translation, scale and rotation respectively. The stochastic discrete recurrence therefore results as:

$$p_{t+1}^x = p_t^x \xi_t^s cos(\xi_t^\theta) - p_t^y \xi_t^s sin(\xi_t^\theta) - \xi_t^s cos(\xi_t^\theta)\xi_t^x + \xi_t^s sin(\xi_t^\theta)\xi_t^y + \xi_t^x \tag{20}$$

$$p_{t+1}^y = p_t^x \xi_t^s sin(\xi_t^\theta) + p_t^y \xi_t^s cos(\xi_t^\theta) - \xi_t^s sin(\xi_t^\theta)\xi_t^x - \xi_t^s cos(\xi_t^\theta)\xi_t^y + \xi_t^y, \tag{21}$$

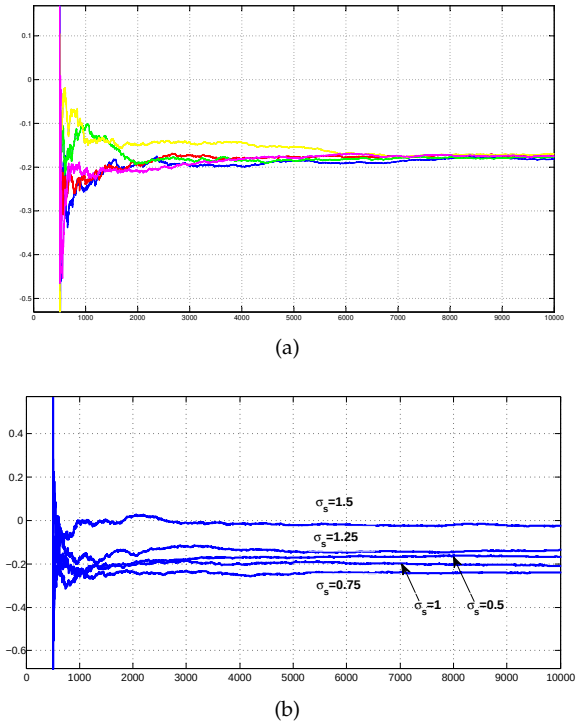


Fig. 5: Asymptotic stability of the Lyapunov Exponents for the stochastic discrete recurrence of Eqs.20 and 21. *(a)*: Convergence trend to a common value for several realizations. *(b)*: Convergence for different values of the standard deviation of the scale parameter σ_s.

where (p_t^x, p_t^y) are the coordinates of a generic location in the object template at time t. The asymptotic stability of Eq.16 is assured when all Lyapunov Exponents γ are negative. □

The robust evaluation of the Lyapunov Exponents can be performed numerically using the algorithm in [10]. Fig. 5(a) shows the temporal convergence of Lyapunov Exponents for several realizations of Eq. 20 and Eq. 21. It can be observed that they all converge to a common negative value. Figure 5(b) shows that temporal convergence still holds for different values of the standard deviation of the scaling parameter σ_s (the most sensitive parameter). In particular, asymptotic stability is confirmed for object detection errors with a standard deviation that is larger up to half the original bounding box size.

4 Experimental Results

We report on a set of quantitative experiments comparing the ALIEN tracker to recent state of the art algorithms. The experiments are evaluated on benchmark sequences that are commonly used in the literature. Since in the ALIEN tracker the object bounding box is oriented, in order to have a consistent comparison with the other methods we have been forced to consider the smallest axis-aligned bounding box that contains the OBB. This results into a slight penalty of the ALIEN performance figures.

4.1 Parameter settings

The parameters of the ALIEN tracker were fixed for all the experiments. The $\lambda_{\mathcal{T}}$ and $\lambda_{\mathcal{C}}$ likelihood ratio

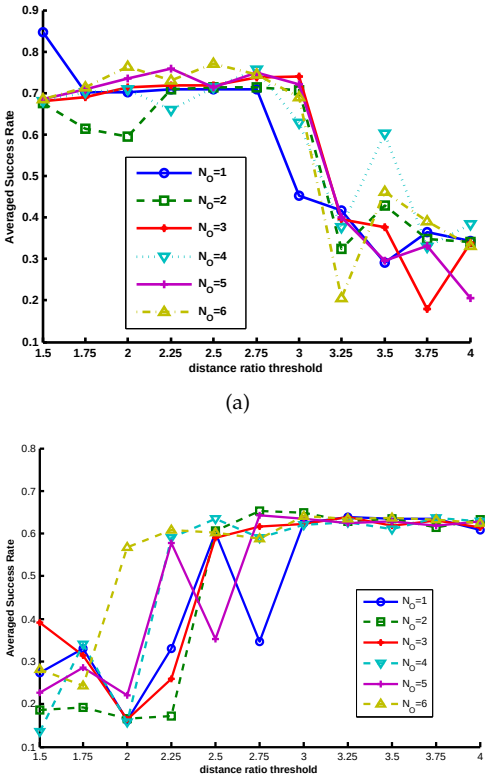


Fig. 6: Average success rate for different values of the likelihood ratio $\lambda_{\mathcal{T}}$ and $\lambda_{\mathcal{C}}$ threshold (both fixed to the same value). *(a)*: Video sequences with no distractors; *(b)*: video sequences with distractors. Curves are plotted for different values of the occlusion threshold $N_{\mathcal{O}}$.

of respectively the object and context classifier, are particularly important and their settings are crucial for the performance in object detection.

In our experiments we have found appropriate to apply the same value for $\lambda_{\mathcal{T}}$ and $\lambda_{\mathcal{C}}$ so to give the same relevance to object and context. Fig. 6 shows the average success rate of the ALIEN tracker as a function of the value of $\lambda_{\mathcal{T}}$ and $\lambda_{\mathcal{C}}$ in two different cases of sequences without and with distractors (objects with similar appearance as the tracked object), respectively. Plots are obtained at different values of $N_{\mathcal{O}}$ (the number of features in $\mathcal{C}_t^\star$ that are also included in the OBB region to assess the presence of occlusion). Fig. 6(a) shows that for scenes with few distractors, the performance of the tracker starts decreasing for values beyond 2.75. Fig. 6(b) shows that values higher than 3 are more suited instead for sequences with distractors. It is also visible that variations of the value of the threshold $N_{\mathcal{O}}$ have little effect on the performance of occlusion detection. In the experiments, the number of features for the object classifier was set to $N_{\mathcal{T}} = 1000$ and the number of features in the context classifier was set to $N_{\mathcal{C}} = 1500$.

The values of the length of the time window l (for feature accumulation in the context classifier in Eq. 6), the max number n_u of frames where the object remains undetected before random search is applied, the maximum allowed speed in scale and

rotation allowed k_s and k_θ (in Eq. 13) have influence on the capability of the ALIEN tracker to resist to critical conditions, such as long occlusions, out of plane rotation of the object, and performance of tracking. We set these parameters with values that correspond to typical conditions observed in most real sequences. Tab. 1 reports the values that we have used

TABLE 1: Parameter values used in the experiments.

$\lambda_\mathcal{T}$	$\lambda_\mathcal{C}$	l	n_u	σ	$N_\mathcal{O}$	k_s	k_θ
2.5	2.5	10	8	1	2	0.5	90

for the experiments reported in this section. We have experimentally observed that little variations around these values have no significant effects on the overall performance.

The number of iterations of the greedy optimization in object state estimation n_R was set to 500. With these settings the current ALIEN implementation runs at 320x240@11 FPS in a Intel i7 CPU quad core @ 2.80 GHz. The overall system is implemented with Matlab except for the SIFT which is based on OpenCV.

4.2 Video Sequences

Verification of the ALIEN performance was carried out on a total number 40 video sequences for a total of 45741 frames that that were used in the literature. Since each sequence has distinguishing aspects that makes it suited to test the tracker in a special critical condition, this permitted the verification of the ALIEN capability to respond to every type of critical conditions as well as its drifting over long term tracking.

We distinguish the following groups of sequences according to their characteristics and use in previously published experiments:

Group 1: the 4 sequences (*Pedestrian1*, *Pedestrian2*, *Pedestrian3*, *Car*) include occlusions (full occlusions in most of the cases) and long intervals where the object disappears from the camera view. These sequences, were used in [71] to compare the performance of their CoGD tracker.

Group 2: in the 8 sequences (*Sylvester*, *Faceocc1*, *Faceocc2*, *Tiger*, *Board*, *Box*, *Lemming*, *Liquor*) objects change their pose frequently and have many short term and partial occlusions; all scenes have also cluttering and some blur. These sequences were used in [49] and [22] to compare the performance of PROST and LSHT, respectively.

Group 3: the 8 sequences (*Jumping*, *Owl*, *Face Body*, *Car1*, *Car2*, *Car3*, *Car4*) include large amounts of blur and abrupt changes of speed and direction of object motion. Most of these sequences were used in [68] to verify the performance of BLUT.

Group 4: the 3 sequences (*Motocross*, *Volkswagen* and *Carchase*) are very long sequences specifically suited to evaluate the long term tracking capability of the tracker. These sequences were used in [25] and [9] (*Motocross* and part of *Carchase* only) for the assessment of Predator-TLD and ConTra, respectively.

Group 5: the 6 sequences (*Animal*, *Clutter*, *Girl*, *ETHPedestrian*, *Multifaces* and *Scale*), include various critical conditions such as cluttering, occlusions, distractors, targets exiting the camera field of view, strong scale changes and out-of-plane rotations. These sequences were used, among the others, in [9] to assess the performance of ConTra.

Group 6: the 11 sequences (*David*, *car4'*, *car11*, *coke11*, *football*, *oneslr*, *shaking*, *singer*, *skating1*, *soccer* and *trellis*) include small objects, objects with homogeneous appearance patterns, shadowing, lighting variation, and articulated object with self-occlusions due to pose changes. They were used for assessment of the the MTT performance in [76].

4.3 Quantitative Results

In the following we compare the performance of the ALIEN tracker with the results of CoGD, PROST, BLUT, Predator-TLD, ConTra, MTT and LSHT and other trackers, for a total of 29 different systems. In order to have a fair comparison, we have reproduced the same tables as presented in [71], [49], [68], [25], [9], [76] and [22] and used the same performance indicators as those originally used by these authors for each group of sequences. For each table, we have indicated the list of sequences that were used for comparison, and for each sequence its sequence group (recalling its characterizing features) and the number of frames. Columns report the name of the trackers that were compared and the performance measures that were published. The last column reports the ALIEN performance. Bold numbers indicate the best performance figures. Second best are underlined.

ALIEN vs CoGD [71]: Table 2(a) compares ALIEN performance against the CoGD tracker and the IVT, ODF, ET and MIL tracker. Most of the sequences belong to Group 1. Performance figures in this table can be assumed as indicators of the capability of the tracker to resist to full occlusions and critical conditions such as exiting/reentering of a target from the camera field of view. The performance was evaluated considering the success rate, defined as the percentage of frames where the target bounding box as an overlap with ground truth larger than 50% (the Pascal Score). ALIEN and CoGD both have largely higher performance than the other methods, with ALIEN scoring the best performance in most of the sequences.

ALIEN vs PROST [49]: Table 2(b) compares against PROST and 4 algorithms: OB, ORF, FT, MIL. All the sequences except the *David* and *Girl* belong to Group 2. The results measure the robustness of the tracker with respect to partial occlusions, pose changes and clutter. Performance was evaluated in terms of both centroid localization error and success rate. ALIEN scores best on all the sequences, outperforming the second best by about 4 times. This performance improvement can be ascribed to the fact that instead of tracking the change of appearance that would results to drift, ALIEN quits tracking until the

object is again visible. This avoids object template degradation and also provides long-term tracking capability as shown in Table 2(e).

ALIEN vs LSHT [22]: Recently the LSHT was proposed to cope with partial occlusions, pose changes and clutter. The authors provided an extensive verification on a subset of Group 2 sequences against 12 trackers: FT, BHT, LGT, L1, SPT, CT, MIL, Struck, VTD, the Predator-TLD, DFT and MTT. Tab. 2(c) shows the ALIEN performance with respect to these trackers in terms of success rate. It can be observed that ALIEN has the higher number of best scores. It has low success rate only on the *Tiger* sequence where the object has very small size in all the frames, so making it difficult to extract keypoints and meaningful local signatures.

ALIEN vs BLUT [68]: Table 2(d) reports the comparison between ALIEN and BLUT and GTK, MIL, OAB, hsvPF, ICTL, VTD, IVT and L1. The sequences analyzed all belong to Group 1. Performance was measured in terms averaged normalized localization error defined as: $\frac{d}{(a+b)/2}$, being d the Euclidean distance between centroids, a and b respectively the height and width of the estimated object bounding box. Both BLUT and ALIEN have one order of magnitude lower errors with respect the other algorithms. All the others fail on these sequences. The BLUT has specific management for blur and scores the best performance. It is anyway important to notice that it has no learning mechanism to adapt the tracker to appearance or shape variations. Due to this, it is likely that its performance degrades in real operating conditions.

ALIEN vs Predator-TLD [25]: The comparison between ALIEN, Predator-TLD and the 5 trackers: OB, SB, BS, MIL and CoGD is reported in Table 2(e) using Precision, Recall and F-measure. Sequences used belong to Group 1 (to verify the robustness to full occlusions) and Group 4 (to verify the long term tracking capability of the tracker). It can be observed that both ALIEN and Predator-TLD both achieve the best performance. This is mainly due to the fact that both trackers manage object reacquisition. Similar results are obtained considering true positives with overlap larger than 25% (rather than 50%) as suggested in [26].

ALIEN vs ConTra [9]: Table 2(f) compares ALIEN against ConTra and 6 others trackers: FT, MIL, CoTT, DNBS, VTD and Predator-TLD. Sequences under test are from Group 5, 4, 2 and 1 and the results indicate the capability of the tracker to adapt to several different critical conditions such as out of plane rotation, scale changes, distractors, total occlusions and targets exiting the camera field of view as well as to perform long term tracking (the *Motocross* and the *Carchase*). In all the sequences, ALIEN scores the best performance with about double accuracy. In particular, in the *Multifaces* sequence, faces are tracked for the entire sequence with no geometric support or explicit tracking of the distractors as in ConTra. ALIEN's much more discriminative capability can be credited to feature oversampling which implicitly provides a local multi-view appearance representation.

ALIEN vs MTT [76]: Table 2(g) reports the comparison between ALIEN, the MTT and the: IVT, L1, MIL, OAB, FT and the VTD. Sequences belong to Group 5, 4 and 2. They show the capability of the tracker to adapt to different conditions and data diversity. All the sequences are short. The average localization error of ALIEN is the lowest in many of the sequences. ALIEN fails tracking in *skating1* where objects have very large self-occlusions. In these conditions features oversampling is almost useless because the visual signal cannot be properly sampled. Failure is also reported in the *shaking* sequence. In this case the face region is mostly uniform and the DoG detector is not effective in localizing keypoints.

4.4 Qualitative Results and Considerations

From the results above it can be noticed that ALIEN is capable of responding to most of the critical conditions that occur in real cases with performance superior or equal to the current state of the art tracking systems. Differently from many other trackers its architecture is not designed to address some specific conditions, but permits instead general adaptivity.

Fig. 7 and Fig. 8 show a few screenshots with conditions that might hamper tracking and the way in which ALIEN responds. Fig. 7(a) shows face tracking under strong occlusion from the *faceocc1* sequence. Crosses represent the features in $\mathcal{O}_t$ of Eq. 14. In this case they are mostly originated from the occluding object. Fig. 7(b) shows a case where the occlusion causes misalignment of the object template which could determine drifting. In this case, the context is capable of intercepting the occluder and avoids learning of misaligned features.

In Fig. 8(a) and Fig. 8(b) we have reported few screenshots of the *car* and *carchase* sequence, respectively. In both sequences, we indicate the bounding boxes of the ALIEN tracker and the other trackers. Fig. 8(a) shows the part of sequence where a vehicle enters and leaves two zones occluded by trees. In both cases, the superior capability to capture the locality of the object appearance makes ALIEN the last to terminate tracking, and the first to re-acquire the object after the full occlusion. Fig. 8(b) shows screenshots after 7000 frames of tracking in the *carchase* sequence, where only ALIEN and Predator-TLD have survived. At this point, the ALIEN continues tracking while the Predator-TLD tracker estimates a wrong object scale and is distracted by the other similar objects. This causes an irreversible drift of the template appearance model.

Finally, Fig. 9(a) shows frames of particular video sequence obtained from the concatenation of 530 unaligned face images of the same subject (former US President G. Bush) taken from the *Labeled Face in*

TABLE 2: ALIEN comparative performance analysis. Bold numbers indicate the best score, underlined numbers indicate the second best.

(a) ALIEN vs CoGD [71] – Success rate (%).

Sequence	Group	Frames	Occ.	IVT [45]	ODF [8]	ET [3]	MIL [4]	CoGD [71]	ALIEN
David	6	761	0	17	-	12	17	**99**	98
Jumping	3	313	0	23	1	14	1	**100**	87
Pedestrian1	1	140	0	7	4	15	72	**100**	**100**
Pedestrian2	1	338	93	10	5	85	26	71	**92**
Pedestrian3	1	184	16	27	2	28	26	83	**90**
Car	1	945	143	83	-	1	4	84	**100**
Mean	-	-	-	27	3	25	24	89	**94**

(b) ALIEN vs PROST [49] – Average localization error/Success rate (%).

Sequence	Group	Frames	OB [15]	ORF [47]	FT [1]	MIL [4]	PROST [49]	ALIEN
Girl	5	452	43.3/24	-	26.5/70	31.6/70	19.0/**89**	**4.514**/66
David	6	502	51.0/23	-	46.0/47	15.6/70	15.3/80	**3.731**/**98**
Sylvester	2	1344	32.9/51	-	11.2/74	9.4/74	10.6/73	**5.213**/**89**
Faceocc1	2	858	49.0/35	-	6.5/**100**	18.4/93	7.0/**100**	**1.357**/99
Faceocc2	2	812	19.6/75	-	45.1/48	14.3/96	17.2/82	**5.949**/**100**
Tiger	2	354	17.9/38	-	39.6/20	8.4/77	7.2/**79**	**3.889**/30
Board	2	698	-	154.5/10	154.5/67.9	51.2/67.9	37.0/75	**10.737**/75
Box	2	1161	-	145.4/28.3	145.4/61.4	104.5/24.5	12.1/91.4	**7.111**/86
Lemming	2	1336	-	166.3/17.2	166.3/54.9	14.9/83.6	25.4/70.5	**9.292/38**
Liquor	2	1741	-	67.3/53.6	67.3/79.9	165.1/20.6	21.6/83.7	**4.039/81**
Mean	-	-	32.9/42.2	133.4/27.3	78.0/58.1	46.1/64.8	18.4/80.4	**5.583**/76.2

(c) ALIEN vs LSHT [22] – Success rate (%).

Sequence	Group	Frames	L1 [38]	SPT [64]	CT [73]	FT [1]	MIL [4]	Struck [20]	VTD [28]	TLD [25]	BHT [50]	LGT [7]	DFT [40]	MTT [76]	LSHT [22]	ALIEN
Board	2	698	3	47	73	82	76	71	81	16	38	5	23	63	**93**	75
Box	2	1161	4	8	33	42	18	**90**	34	60	8	9	37	25	84	86
David	1	761	41	64	46	35	24	67	32	90	7	24	45	92	93	**98**
Faceocc2	2	812	60	22	**100**	80	94	79	77	76	43	8	49	82	**100**	**100**
Sylvester	2	1344	48	34	74	66	70	87	72	76	78	8	54	**96**	89	89
Tiger	2	354	10	3	65	5	**77**	65	17	26	5	2	21	27	66	30
Trellis	6	568	67	72	35	18	34	70	54	31	18	2	45	34	91	**92**
Mean	-	-	29	31	53	41	49	66	45	46	24	7	34	52	77	**71**

(d) ALIEN vs BLUT [68] – Average normalized centroid localization error.

Sequence	Group	Frames	GTK [51]	MIL[4]	OAB[16]	hsvPF[43]	ICTL[66]	VTD[28]	IVT[45]	L1[38]	BLUT[68]	ALIEN
owl	3	631	0.121	0.467	0.544	0.088	0.027	0.683	0.358	0.479	**0.011**	0.042
face	3	493	0.081	0.523	0.960	0.069	0.044	0.123	0.039	0.560	**0.027**	0.039
body	3	334	0.082	0.613	0.244	0.170	0.105	0.388	0.334	0.331	**0.033**	0.049
car1	3	742	0.317	0.626	1.012	0.594	0.193	0.862	0.639	0.622	**0.019**	0.034
car2	3	585	0.697	0.775	0.510	0.272	0.297	0.563	1.679	0.555	**0.023**	0.040
car3	3	357	0.697	0.569	0.345	0.145	0.152	0.328	0.321	0.538	**0.013**	0.032
car4	3	380	0.871	0.974	0.516	0.670	0.304	0.260	0.248	0.452	0.050	**0.028**
Mean	-	-	0.409	0.650	0.590	0.287	0.160	0.458	0.517	0.505	**0.025**	*0.037*

(e) ALIEN vs Predator-TLD [25] – Precision/Recall/F-measure.

Sequence	Group	Frames	OB [15]	SB [17]	BS [56]	MIL [4]	CoGD[71]	TLD [25]	ALIEN
David	6	761	0.01 / 0.01 / 0.01	0.27 / 0.27 / 0.27	0.16 / 0.12 / 0.13	0.06 / 0.06 / 0.06	0.99 / 0.99 / 0.99	**1.00 / 1.00 / 1.00**	0.99 / 0.98 / 0.99
Jumping	3	313	0.41 / 0.04 / 0.08	0.14 / 0.08 / 0.10	0.06 / 0.05 / 0.05	0.37 / 0.37 / 0.37	**1.00 / 0.99 / 1.00**	0.99 / 0.99 / 0.99	0.99 / 0.87 / 0.92
Pedestrian1	1	140	0.36 / 0.09 / 0.14	0.20 / 0.14 / 0.16	0.10 / 0.04 / 0.05	0.42 / 0.42 / 0.42	0.99 / 0.99 / 0.99	**1.00 / 1.00 / 1.00**	**1.00 / 1.00 / 1.00**
Pedestrian2	1	338	0.74 / 0.12 / 0.21	0.55 / 0.46 / 0.50	1.00 / 0.02 / 0.04	0.10 / 0.12 / 0.11	0.71 / 0.90 / 0.79	0.89 / 0.92 / 0.91	**0.93 / 0.92 / 0.93**
Pedestrian3	1	184	1.00 / 0.33 / 0.49	0.41 / 0.33 / 0.36	0.81 / 0.40 / 0.54	0.49 / 0.58 / 0.53	0.84 / 0.99 / 0.91	**0.99 / 1.00 / 0.99**	1.00 / 0.90 / 0.95
Car	1	945	0.89 / 0.57 / 0.69	1.00 / 0.67 / 0.80	0.99 / 0.56 / 0.72	0.11 / 0.12 / 0.11	0.91 / 0.92 / 0.91	0.92 / 0.97 / 0.94	**0.95 / 1.00 / 0.98**
Motocross	4	2665	0.13 / 0.00 / 0.00	0.01 / 0.00 / 0.00	0.14 / 0.00 / 0.00	0.02 / 0.01 / 0.01	0.80 / 0.26 / 0.39	0.67 / 0.58 / 0.62	**0.69 / 0.81 / 0.74**
Volkswagen	4	8576	0.04 / 0.00 / 0.00	0.00 / 0.00 / 0.00	0.00 / 0.00 / 0.00	0.26 / 0.03 / 0.05	0.41 / 0.03 / 0.06	0.54 / 0.64 / 0.59	**0.98 / 0.89 / 0.93**
Carchase	4	9928	0.73 / 0.03 / 0.05	0.79 / 0.04 / 0.08	0.38 / 0.09 / 0.14	0.49 / 0.03 / 0.05	0.87 / 0.04 / 0.08	0.50 / 0.40 / 0.45	**0.73 / 0.68 / 0.70**
mean	-	-	0.40 / 0.04 / 0.06	0.39 / 0.06 / 0.09	0.24 / 0.07 / 0.10	0.32 / 0.04 / 0.06	0.70 / 0.16 / 0.20	0.58 / 0.57 / 0.58	**0.84 / 0.80 / 0.82**

(f) ALIEN vs ConTra [9] – Average localization error.

Sequence	Group	Frames	FT [1]	MIL [4]	CoTT [70]	DNBS [31]	VTD [28]	TLD [25]	ConTra [9]	ALIEN
Animal	5	72	69	9	8	19	6	37	*9*	**3.47**
Carchase	4	5000	lost@355	lost@355	lost@409	lost@364	lost@357	lost@1645	24+	**4.91**
Clutter	5	1528	lost@1081	lost@413	9	6	6	4	6	**3.75**
ETHPedestrian	5	874	lost@95	lost@95	lost@95	lost@635	lost@95	10	*16*	**6.43**
Girl	5	502	lost@248	30	14	39	69	19	*18*	**2.33**
Liquor	2	1407	lost@47	lost@288	30	lost@404	lost@404	21	10	**4.03**
Motocross	4	2665	lost@137	lost@485	lost@591	lost@10	lost@10	10	12	**8.24**
Multifaces	5	1006	lost@64	lost@64	lost@394	lost@64	lost@64	lost@97	26	**11.92**
Scale	5	1911	8	11	6	lost@269	3	6	2	**1.88**
Car	1	946	lost@679	lost@481	9	lost@517	lost@517	8	8	**3.42**
Speed (fps, on 320x240)	-	-	1.6	14	2	7	0.2	12	10	11

+ tracking stopped at frame 5000

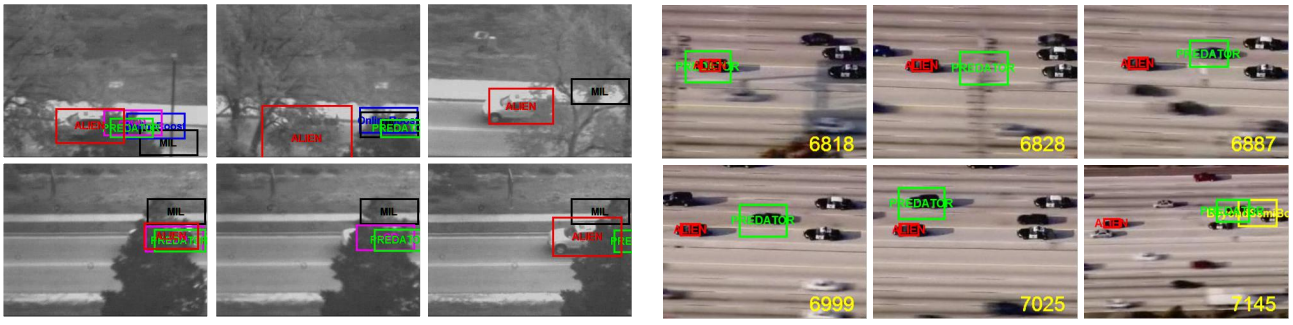
(g) ALIEN vs MTT [76] – Average localization error.

Sequence	Group	Frames	IVT [45]	L1 [38]	MIL [4]	OAB [16]	FT [1]	VTD [28]	MTT L^*_{21} [76]	MTT $L^*_{\infty 1}$ [76]	MTT L^*_{11} [76]	MTT L_{21} [76]	MTT $L_{\infty 1}$ [76]	MTT L_{11} [76]	ALIEN
car4′	6	659	8.50	6.37	53.76	88.12	127.29	27.01	**1.79**	2.40	3.60	2.25	3.83	3.33	6.23
car11	6	392	19.18	5.42	53.75	5.69	72.71	3.74	**2.01**	2.02	16.73	1.93	**2.01**	23.69	2.69
coke11	6	291	12.09	58.53	13.70	11.33	70.98	62.70	7.34	9.46	7.38	**3.15**	7.39	6.53	5.80
david	6	761	9.87	7.16	25.95	21.15	66.91	58.99	7.55	8.38	6.67	7.88	8.37	9.17	**3.73**
faceocc1	2	858	7.00	9.15	34.35	17.23	7.90	8.73	12.52	18.59	10.92	7.75	23.45	19.15	**1.35**
faceocc2	2	812	15.25	6.55	10.23	20.84	48.22	11.85	6.45	7.04	10.07	8.09	8.06	10.11	**5.94**
football	6	362	15.44	5.25	7.98	53.31	6.35	3.33	4.07	5.19	4.51	4.71	4.87	5.25	**2.37**
girl	5	452	4.98	4.16	12.38	10.99	7.43	11.44	**3.88**	4.61	5.43	4.49	4.33	9.85	4.51
onelsr	6	560	4.69	24.03	23.82	12.46	57.57	44.29	3.39	**2.28**	20.26	3.26	3.63	23.61	4.94
shaking	6	365	37.78	52.23	7.91	100.27	15.27	3.96	9.35	8.62	10.80	8.36	9.54	11.37	9.49[1]
singer1	6	351	5.25	9.79	11.09	62.99	26.92	1.45	1.43	**1.14**	4.53	1.82	4.74	18.88	2.56
skating1	6	400	20.08	74.94	49.18	39.25	63.26	5.02	5.61	9.18	75.51	7.41	82.48	77.89	9.51[2]
soccer	6	392	58.49	97.78	46.34	65.31	41.42	10.50	**10.45**	11.06	17.16	14.26	46.99	16.06	12.95[3]
sylvester	2	1344	14.53	39.37	15.29	10.42	6.78	7.37	4.77	**3.95**	4.01	4.78	4.71	7.56	5.21
trellis	6	569	31.08	54.02	37.32	41.46	55.69	47.76	10.30	17.76	15.36	10.32	9.96	33.14	**4.65**
mean	-	-	17.61	30.32	26.87	37.39	44.98	20.54	6.06	7.44	14.20	6.03	14.96	18.37	**5.46**

[1]Object lost at frame 65; [2]Object lost at frame 55; [3]Object lost at frame 39.

(a) (b)

Fig. 7: Screenshots of ALIENs tracking with the *Face occlusion1* and *Multiplefaces* sequences. *(a)*: In presence of strong partial occlusion. *(b)*: In presence of distractors (other faces). Occluding and ambiguous features are marked with crosses.



(a) (b)

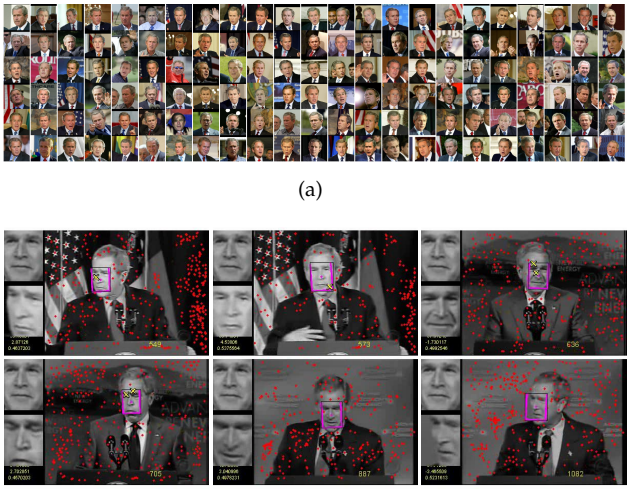
Fig. 8: (a): Full occlusion condition: ALIEN tracks the object until it is fully occluded (1st, 2nd, 4th image) and is the first to recover tracking (3rd and 6th image). *(b)*: Distractors and scale variation in long term tracking: after about 7000 frames. Only ALIEN and Predator-TLD survived with no drift; the presence of distractors and the wrong scale estimates terminate correct tracking of Predator-TLD.

the Wild dataset [23]. The effects of the incremental learning capability of the ALIEN tracker can be observed in Fig. 9(b). The algorithm was run through the dataset several times until the object was detected approximatively in the 90% of the images. This produced about 10000 weakly aligned local features. The learned appearance was then checked by tracking the face in several video clips taken from YouTube (with no template updating). It can be noticed that the tracker is robust against arbitrary viewing conditions and distractors.

Demo sequences are available at the link ALIEN-demo or at the YouTube channel http://www.youtube.com/user/pernixVision.

5 Conclusions

In this paper, we have proposed a method to track an object in long video sequences under complex interactions between illumination, occlusion and object/camera motion. We have presented a novel visual object representation based on weakly aligned multiple instance local features which improves on the inherent limit of local features invariance under occlusion, sensor quantization and casting shadow. A non parametric learning algorithm based on transitive matching exploits discriminative classifiers in order to separate the object from context and detect occlusions. In order to avoid template contamination, when occlusion is detected, the object template is not updated.



(a)

(b)

Fig. 9: (a): A clip of the video sequence artificially obtained from images of the *Labeled face in the wild*[23] used for learning the face appearance. (b): The learned appearance is then used to track face identity in YouTube videos. Red points show features matched with the context.

A real-time implementation of the framework has been evaluated under publicly available datasets with an extensive set of experiments and comparisons with state of the art approaches. Superior or equal tracking performance is reported in most of the cases.

Our analysis shows that the appearance learning algorithm is asymptotically stable even in the presence of large errors in object detection. This result might appear somewhat surprising since one would expect that errors accumulate and the drift grows gradually during tracking. We prove that under mild conditions this will not happen. Hence we believe this is a useful result for the template update problem.

REFERENCES

[1] A. Adam, E. Rivlin, and I. Shimshoni. Robust fragments-based tracking using the integral histogram. In *CVPR*, volume 1, pages 798 – 805, june 2006. 2, 11, 12

[2] M. Ambai and Y. Yoshida. Card: Compact and real-time descriptors. In *ICCV*, pages 97–104, 2011. 4

[3] S. Avidan. Ensemble tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29:261–271, February 2007. 2, 11

[4] B. Babenko, M.-H. Yang, and S. Belongie. Robust object tracking with online multiple instance learning. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(8):1619 –1632, aug. 2011. 2, 3, 4, 11, 12

[5] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (surf). *Comput. Vis. Image Underst.*, 110:346–359, June 2008. 4

[6] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. Brief: Binary robust independent elementary features. In *ECCV (4)*, pages 778–792, 2010. 4

[7] L. Cehovin, M. Kristan, and A. Leonardis. An adaptive coupled-layer visual model for robust visual tracking. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1363–1370. IEEE, 2011. 2, 11

[8] R. Collins and Y. Liu. On-line selection of discriminative tracking features. In *ICCV*, volume 1, pages 346 – 352, October 2003. 2, 11

[9] T. B. Dinh, N. Vo, and G. Medioni. Context tracker: Exploring supporters and distracters in unconstrained environments. In *CVPR*, june 2011. 2, 9, 10, 11

[10] J. P. Eckmann and D. Ruelle. Ergodic theory of chaos and strange attractors. volume 57, pages 617–656. American Physical Society, Jul 1985. 8

[11] H. Furstenberg and H. Kesten. Products of random matrices. *Ann. Math. Statistics*, 31:457–469, 1960. 7

[12] J. Gall, N. Razavi, and L. Van Gool. On-line adaption of class-specific codebooks for instance tracking. In *BMVC*, pages 55.1–55.12, 2010. 3

[13] J. J. Gibson. The ecological approach to visual perception. *LEA*, 1984. 1

[14] M. Godec, P. M. Roth, and H. Bischof. Hough-based tracking of non-rigid objects. *ICCV 2011*, 2011. 2

[15] H. Grabner and H. Bischof. On-line boosting and vision. In *CVPR*, volume 1, pages 260 – 267, june 2006. 2, 11

[16] H. Grabner, M. Grabner, and H. Bischof. Real-time tracking via on-line boosting. In *BMVC*, pages 47–56, 2006. 2, 11, 12

[17] H. Grabner, C. Leistner, and H. Bischof. Semi-supervised on-line boosting for robust tracking. In *ECCV (1)*, pages 234–247, 2008. 2, 11

[18] S. Gu, Y. Zheng, and C. Tomasi. Efficient visual object tracking with online nearest neighbor classifier. In *ACCV (1)*, pages 271–282, 2010. 2, 3

[19] R. M. Haralick. Propagating covariance in computer vision. *International journal of pattern recognition and artificial intelligence*, 10(05):561–572, 1996. 7

[20] S. Hare, A. Saffari, and P. Torr. Struck: Structured output tracking with kernels. *IEEE International Conference on Computer Vision (ICCV 2011)*, 2011. 2, 11

[21] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*, volume 2. Cambridge, 2000. 7

[22] S. He, Q. Yang, R. W. Lau, J. Wang, and M.-H. Yang. Visual tracking via locality sensitive histograms. In *Computer Vision and Pattern Recognition, 2008. CVPR 2013. IEEE Conference on*. IEEE, 2013. 2, 9, 10, 11

[23] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller. Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments. In *Workshop on Faces in 'Real-Life' Images: Detection, Alignment, and Recognition*. Erik Learned-Miller and Andras Ferencz and Frédéric Jurie, 2008. 12, 13

[24] X. Jia, H. Lu, and M.-H. Yang. Visual tracking via adaptive structural local sparse appearance model. In *CVPR*, 2012. 2

[25] Z. Kalal, J. Matas, and K. Mikolajczyk. P-n learning: Bootstrapping binary classifiers by structural constraints. In *CVPR*, june 2010. 2, 9, 10, 11

[26] Z. Kalal, J. Matas, and K. Mikolajczyk. Tracking-learning-detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011. 10

[27] S. Kwak, W. Nam, B. Han, and J. H. Han. Learning occlusion with likelihoods for visual tracking. *ICCV 2011*, 2011. 2

[28] J. Kwon and K. M. Lee. Visual tracking decomposition. In *CVPR*, pages 1269–1276, 2010. 2, 11, 12

[29] B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *Proceedings of the Workshop on Statistical Learning in Computer Vision*, Prague, Czech Republic, May 2004. 3

[30] S. Leutenegger, M. Chli, and R. Siegwart. BRISK: Binary Robust Invariant Scalable Keypoints. In *ICCV*, 2011. 4

[31] A. Li, F. Tang, Y. Guo, and H. Tao. Discriminative nonorthogonal binary subspace tracking. ECCV, pages 258–271, 2010. 2, 11

[32] H. Li, C. Shen, and Q. Shi. Real-time visual tracking using compressive sensing. In *CVPR*, pages 1305–1312, 2011. 2

[33] X. Li, W. Hu, C. Shen, Z. Zhang, A. Dick, and A. van den Hengel. A survey of appearance models in visual object tracking. *ACM Transactions on Intelligent Systems and Technology*, 2013. 2

[34] B. Liu, J. Huang, L. Yang, and C. Kulikowsk. Robust tracking using local sparse appearance model and k-selection. In *CVPR*, june 2011. 2

[35] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60:91–110, 2004. 1, 3, 4, 5

[36] L. Lu and G. D. Hager. A nonparametric treatment for location/segmentation based visual tracking. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007. 2, 3

[37] I. Matthews, T. Ishikawa, and S. Baker. The template update problem. In *BMVC*, October 2003. 3

[38] X. Mei and H. Ling. Robust visual tracking and vehicle classification via sparse representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(11):2259 –2272, nov. 2011. 2, 11, 12

[39] X. Mei, H. Ling, Y. Wu, E. Blasch, and L. Bai. Minimum error bounded efficient l1 tracker with occlusion detection. In *CVPR*,

2011, june 2011. 2
[40] M. Narayana, A. Hanson, and E. Learned-Miller. Improvements in joint domain-range modeling for background subtraction. In *Proceedings of the British Machine Vision Conference*, pages 115–1, 2012. 2, 11
[41] H. T. Nguyen and A. W. Smeulders. Robust tracking using foreground-background texture discrimination. *Int. J. Comput. Vision*, 69(3):277–293, Sept. 2006. 2
[42] V. I. Oseledec. A multiplicative ergodic theorem. Liapunov characteristic numbers for dynamical systems. *Trans. Moscow Math. Soc.*, 19:197–221, 1968. 3, 7
[43] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet. Color-based probabilistic tracking. In *ECCV (1)*, pages 661–675, 2002. 2, 11
[44] R. Raguram, J.-M. Frahm, and M. Pollefeys. Exploiting uncertainty in random sample consensus. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 2074–2081. IEEE, 2009. 7
[45] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang. Incremental learning for robust visual tracking. *Int. J. Comput. Vision*, 77:125–141, May 2008. 2, 11, 12
[46] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb: An efficient alternative to sift or surf. In *ICCV*, Barcelona, 11/2011 2011. 4
[47] A. Saffari, C. Leistner, J. Santner, M. Godec, and H. Bischof. On-line random forests. In *in 3rd IEEE ICCV Workshop on On-line Computer Vision.*, 2009. 2, 11
[48] S. Salti, A. Cavallaro, and L. Di Stefano. Adaptive appearance modeling for video tracking: Survey and evaluation. *IEEE Transaction on Image Processing*, 2012. 2
[49] J. Santner, C. Leistner, A. Saffari, T. Pock, and H. Bischof. Prost: Parallel robust online simple tracking. In *CVPR, 2010*, pages 723 –730, june 2010. 2, 9, 11
[50] S. Shahed Nejhum, J. Ho, and M.-H. Yang. Visual tracking with histograms and articulating blocks. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008. 2, 11
[51] C. Shen, J. Kim, and H. Wang. Generalized kernel-based visual tracking. *IEEE Trans. Circuits Syst. Video Techn.*, 20(1):119–130, 2010. 2, 11
[52] S. Soatto. Actionable information in vision. In *ICCV*, October 2009. 1
[53] S. Soatto. Steps towards a theory of visual information: Active perception, signal-to-symbol conversion and the interplay between sensing and control. *arXiv preprint arXiv:1110.2053*, 2011. 1
[54] S. Soatto. Actionable information in vision. In *Machine learning for computer vision*, pages 17–48. Springer, 2013. 1
[55] M. Sofka, G. Yang, and C. V. Stewart. Simultaneous covariance driven correspondence (cdc) and transformation estimation in the expectation maximization framework. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007. 7
[56] S. Stalder, H. Grabner, and L. V. Gool. Beyond semi-supervised tracking: Tracking should be as simple as detection, but not simpler than recognition. In *OLCV 09: 3rd On-line learning for Computer Vision Workshop*, 2009. 2, 11
[57] Supancic and D. Ramanan. Self-paced learning for long-term tracking. *Computer Vision and Pattern Recognition (CVPR)*, 2013. 3
[58] B. Tordoff and R. Cipolla. Uncertain ransac. In *Proceedings of the IAPR Workshop on Machine Vision Applications (MVA), Tsukaba, Japan*, 2005. 7
[59] B. Tordoff and D. W. Murray. Guided sampling and consensus for motion estimation. In *ECCV '02*, pages 82–98, 2002. 5
[60] A. Tsymbal. The problem of concept drift: definitions and related work. Technical Report TCD-CS-2004-15, The University of Dublin, Trinity College, Department of Computer Science, 2004. 3
[61] P. Viola and M. J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, 2004. 2
[62] Q. Wang, F. Chen, W. Xu, and M.-H. Yang. An experimental comparison of online object tracking algorithms. *Proceedings of SPIE: Image and Signal Processing Track*, 2011. 2
[63] Q. Wang, F. Chen, W. Xu, and M.-H. Yang. Online discriminative object tracking with local sparse representation. In *Proceedings of IEEE Workshop on the Applications of Computer Vision (WACV)*, January 2012. 2
[64] S. Wang, H. Lu, F. Yang, and M.-H. Yang. Superpixel tracking. *ICCV*, 2011. 2, 11
[65] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(2):210 –227, feb. 2009. 2
[66] Y. Wu, J. Cheng, J. Wang, and H. Lu. Real-time visual tracking via incremental covariance tensor learning. In *ICCV*, pages 1631–1638, 2009. 11
[67] Y. Wu, J. Lim, , and M.-H. Yang. Online object tracking: A benchmark. *To appear in Computer Vision and Pattern Recognition (CVPR)*, 2013. 2
[68] Y. Wu, H. Ling, J. Yu, F. Li, X. Mei, and E. Cheng. Blurred target tracking by blur-driven tracker. *ICCV*, 2011. 2, 9, 10, 11
[69] H. Yang, L. Shao, F. Zheng, L. Wang, and Z. Song. Recent advances and trends in visual tracking: A review. *Neurocomputing*, 74(18):3823–3831, 2011. 2
[70] M. Yang, Y. Wu, and G. Hua. Context-aware visual tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(7):1195–1209, July 2009. 2, 11
[71] Q. Yu, T. B. Dinh, and G. G. Medioni. Online tracking and reacquisition using co-trained generative and discriminative trackers. In *ECCV (2)*, pages 678–691, 2008. 2, 9, 11
[72] B. Zeisl, C. Leistner, A. Saffari, and H. Bischof. On-line Semi-supervised Multiple-Instance Boosting. In *CVPR*, 2010. 3
[73] K. Zhang, L. Zhang, and M.-H. Yang. Real-time compressive tracking. In *ECCV 2012*, pages 864–877. Springer, 2012. 2, 11
[74] S. Zhang, H. Yao, X. Sun, and X. Lu. Sparse coding based visual tracking: Review and experimental comparison. *Pattern Recognition*, 2012. 2
[75] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja. Robust visual tracking via multi-task sparse learning. In *CVPR*, june 2012. 2
[76] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja. Robust visual tracking via structured multi-task sparse learning. *International Journal of Computer Vision*, 101(2):367–383, 2013. 2, 9, 10, 11, 12
[77] W. Zhong, H. Lu, , and M.-H. Yang. Robust object tracking via sparsity-based collaborative model. In *CVPR*, 2012. 2

Federico Pernici received the laurea degree in information engineering in 2002, the postlaurea degree in internet engineering 2003, and the PhD degree in information and telecommunications engineering in 2005, from the University of Florence, Italy. Since 2002, he has been a research professor at the same University. He is an associate editor of Machine Vision and Application. His scientific interests are pattern recognition and computer vision with focus on different aspects of visual tracking.

Alberto Del Bimbo Alberto Del Bimbo is a full professor of computer engineering, the director of the Master in Multimedia and the director of the Media Integration and Communication Center at the University of Florence, where he was the deputy rector for Research and Innovation Transfer from 2000 to 2006 and the president of the Foundation for Research and Innovation. His scientific interests are computer vision, image and video analysis and multimedia information retrieval. He has published more than 250 publications in some of the most distinguished scientific journals and international conferences, and is the author of the monograph Visual Information Retrieval. He is a member of the IEEE Computer Society. From 1996 to 2000, he was the president of the IAPR Italian Chapter, and from 1998 to 2000, a member at large of the IEEE Publications Board. He was the General Chair of IAPR ICIAP 1997, IEEE ICMCS 1999 and the program co-chair of ACM Multimedia 2008. He was the General Co-chair of ACM Multimedia 2010 and ECCV 2012. He is an IAPR fellow and associate editor of Multimedia Tools and Applications, Pattern Analysis and Applications, Journal of Visual Languages and Computing, and International Journal of Image and Video Processing, and was an associate editor of Pattern Recognition, IEEE Transactions on Multimedia, and IEEE Transactions on Pattern Analysis and Machine Intelligence.