Tracking multiple high-density homogeneous targets

Fabio Poiesi and Andrea Cavallaro

***Abstract*—We present a framework for multi-target detection and tracking that infers candidate target locations in videos containing a high density of homogeneous targets. We propose a gradient-climbing technique and an isocontour slicing approach for intensity maps to localize targets. The former uses Markov Chain Monte Carlo to iteratively fit a shape model onto the target locations, whereas the latter uses the intensity values at different levels to find consistent object shapes. We generate trajectories by recursively associating detections with a hierarchical graph-based tracker on temporal windows. The solution to the graph is obtained with a greedy algorithm that accounts for false positive associations. The edges of the graph are weighted with a likelihood function based on location information. We evaluate the performance of the proposed framework on challenging datasets containing videos with high density of targets and compare it with six alternative trackers.**

***Index Terms*—High-density targets, crowd, target detection, multi-target tracking.**

I. Introduction

MULTI-TARGET video detection and tracking in scenes with a high density of targets can help in a range of applications, from surveillance to biological studies [1]–[3]. A feature extraction stage generally processes images using prior knowledge (e.g. color, shape, size) and generates estimated target locations using feature values and classification scores (*confidence maps*) [2]. A confidence map is a (noisy) scalar representation of likely target locations [2], [4], [5] that uses sparse [6] or dense [4] confidence values. Sparse values can be obtained with Support Vector Machines (SVM) through sliding windows [6]. Dense values are generated with multi-layer homographies [7] or derived from sparse values by low-pass filtering the confidence map [4]. Candidate target locations are then extracted by thresholding and clustering the sparse values with the highest scores [6], [8]. Without using classifiers, target locations can be extracted by enhancing the target appearance (e.g. by means of color filters) and by localizing the regions with high-intensity values [9]. We refer to enhanced target appearance features as *target-intensity maps*.

Target trajectories can be generated by temporally associating candidate locations with multi-target trackers [4], [10]–[12] or directly from confidence maps [2], [13]. Generally, candidate target locations are generated by applying thresholds, clustering and Non-Maxima Suppression (NMS) to the confidence values [6]. However, weakly detected features of different parts of a target may lead to confidence values with multiple peaks in the target region [8]. Moreover, multi-peak confidence values can be due to adjacent targets and explicit models to separately detect targets are devised. In high-density scenes with targets having the same or similar appearance, multi-target tracking is addressed with complex priors on target motion [14] and appearance [12]. Alternatively, tracking can be performed directly on confidence maps [2] or on detections extracted from target-intensity maps [15].

In this paper, we propose a multi-target detection and tracking approach for videos with a high density of homogeneous undistinguishable targets. Unlike [3], the detection is background independent, suitable for cluttered videos and can deal with multi-peak target-intensities generated by adjacent and overlapping targets. Importantly, the detector does not require learning target appearance models such as colors or textures [15] . The novelty of the detection algorithm is the possibility of automatically localizing targets via local maxima searching by exploiting the 2D gradient inferred from their outline. This allows us to apply the approach to heterogenous targets (by only inputting their size) and to achieve robustness for the localization of targets with irregular outline without introducing temporal dependencies that might lead to drifts when targets overlap [4]. The detector relies also on isocountours applied to target-intensity maps, which improve the alignment of the detections (centering and orientation) with the targets. We track targets with a greedy graph-based method that pair-wise matches short tracks [12] and performs backward validation within temporal windows [16]. The novelty of this greedy solution is to identify and discard (online) false-positive short-tracks during the association process. The number of targets is implicitly inferred by the algorithm. Initialization and termination of tracks are automatically performed in any location of the scene. The proposed tracking algorithm outperforms alternative methods on challenging datasets. The software of the proposed tracking method is available at http://www.eecs.qmul.ac.uk/~andrea/thdt.html.

The paper is organized as follows. Sec. II discusses prior works on multi-target detectors and trackers, whereas Sec. III presents the problem formulation. Sec. IV and Sec. V describe the proposed approach for detection and tracking, respectively. Sec. VI analyzes the computational cost. The results and the comparisons with alternative methods are discussed in Sec. VII. Finally, in Sec. VIII we draw conclusions and present future research directions.

II. State of the art

In this section, we review state-of-the-art methods for target localization that generate candidate locations for tracking and discuss their strengths and limitations.

Target detectors may extract candidate target locations (measurements) using descriptors designed either for specific object

F. Poiesi and A. Cavallaro are with the Centre for Intelligent Sensing, Queen Mary University of London, Mile End Road, UK, E1 4NS, emails: fabio.poiesi@qmul.ac.uk, a.cavallaro@qmul.ac.uk. This work was supported in part by the Artemis JU and in part by the UK Technology Strategy Board through COPCAMS Project under Grant 332913.

representations or for the background [17]. Thresholding followed by clustering can be applied on confidence maps generated with a person detector [8]. The person model embeds features extracted from image patches (e.g. intensity gradient) and is trained with an SVM [6]. The feature extractor uses a sliding window to extract unclassified patches. The patches are provided to the classifier to determine their affiliation to the person or non-person class. The confidence values representing such affiliation in each location of the image are then generated. Next, candidate target locations are extracted by thresholding the confidence map and clustered with Mean-Shift [18]. Negligible overlapping target locations can be further removed using NMS [8]. Linear [9] or morphological [19] image filters can also be used for target localization. The H-dome transform based on morphology can be used to enhance target intensities and to suppress background noise. Gradient information can then be employed to improve the discrimination between clutter and target intensities [9]. Alternatively, wavelet filters (B3-spline and Haar) [20] can be used to segment regions with high-intensity values while discarding low-frequency regions belonging to the background. Target locations are finally generated through pyramidal de-noising based on image cross-correlation. These methods are based on the assumption that confidence values and target intensities are modeled as single-peak distributions for each target. Threshold-based methods followed by clustering and NMS are then applied to extract target locations. Although extensions to multi-peak confidence values exist, they are explicitly designed for human targets only [8].

After the detection stage, tracking temporally associates candidate target locations sequentially (i.e. online), within a temporal buffer (i.e. with a delay) or as a batch process (i.e. offline). *Sequential tracking methods* can be based on Bayesian filters [4], [21] and can operate directly on confidence maps. Markov Chain Monte Carlo (MCMC) can be used in the case of noisy measurements [1]. Simple heuristics such as elliptical shapes along with a method based on sparse least squares are then used to temporally validate and link such measurements [1]. Initialization is performed using the ground truth. Measurements can be associated frame-by-frame either via thresholding or optimum association (e.g. the Hungarian algorithm [22]). The association probability can be inferred through the product of three independent affinities relying on target position, size and appearance [12]. However, in scenes with high-density targets, sequential methods require strong prior knowledge, such as motion models trained on the same scene type [14]. *Buffered trajectories* can be generated with a track-before-detect algorithm based on particle filtering that enables multi-target tracking directly on confidence maps [2]. Particles are spread over the confidence map and the clutter is filtered out through the likelihood function while temporally linking measurements with high confidence. Markov Random Fields is employed within the tracker to keep nearby targets separate whilst tracking. Alternatively, MCMC can be used to reduce the computational complexity with a large number of targets. MCMC is employed to generate target trajectories by confirming track hypotheses within a 100-frame window using Minimum Description Length [10]. The Vessel filter can be used to filter out noisy confidence values during tracking (particle filter) and short temporal intervals of accumulated confidence maps [23]. However, the temporal filtering on short temporal windows might cause the merging of confidence values in the case of nearby targets. Finally, trajectories can be generated *offline* by formulating the problem as a graph [16] or as an energy minimization problem [24]. A graph is composed of nodes representing measurements and edges representing the cost of moving from one node to another (e.g. via log-likelihood [12]). Target trajectories are extracted by finding the minimum-cost paths that connect the nodes of the graph using, for example, the min-cost flow algorithm [25]–[27]. This optimization is an NP-hard problem and, hence, it is necessary to relax the constraints and employ suboptimal solutions. A greedy algorithm developed with dynamic programming can be used to compute approximated graph solutions [16], [26].

Table I summarizes relevant state-of-the-art methods.

TABLE I
COMPARISON OF DETECTION AND TRACKING METHODS. KEY: NMS: NON-MAXIMA SUPPRESSION; MCMC: MARKOV CHAIN MONTE CARLO; N/A: NOT AVAILABLE.

Ref.	Features	Mapping	Tracking	Operation	Supervised
[17]	background subtraction	threshold +clustering	n/a	n/a	no
[8]	gradient	threshold +clustering+nms	n/a	n/a	no
[6]	gradient	threshold +clustering+nms	n/a	n/a	no
[19]	morphology	thresholds	n/a	n/a	no
[20]	wavelet	threshold	n/a	n/a	no
[15]	gradient	threshold	Kalman filter	online	yes
[4]	gradient	no mapping	particle filter	online	no
[1]	color	threshold	MCMC	online	no
[9]	H-dome +gradient	threshold	particle filter	online	no
[10]	gradient +clustering	threshold	MCMC	buffer	no
[12]	gradient	threshold	threshold +Hungarian	online	no
[26]	gradient	threshold +clustering+nms	min-cost flow	offline	no
[27]	background subtraction	minimization	min-cost flow	offline	no
[2]	n/a	no mapping	particle filter	buffer	no

III. PROBLEM FORMULATION

Let $\mathbf{V} = \{V(k)\}_{k=1}^{K}$ be a video composed of K frames $V(k)$. Let $C_i(k) \in \mathbb{R}_{[0,1]}$ be the target intensity (feature) value of the i^{th} pixel with generic coordinates (x_i, y_i). The larger $C_i(k)$, the clearer the target appearance. $\mathcal{C}(k)$ is the target-intensity map extracted from $V(k)$ having $C_i(k)$ as elements with $i = 1, ..., I$ and I the total number of pixels in a frame.

Let $\mathcal{Z}(k) = \{\mathbf{z}_n(k)\}_{n=1}^{N(k)}$ be the set of detections, where $N(k)$ represents the number of detected targets at frame k. The $\mathbf{z}(k)$ of a generic target is represented as

$$\mathbf{z}(k) = [x(k)\ y(k)\ S(k)\ \iota(k)]^T, \tag{1}$$

where $(x(k), y(k))^T$ is the position in the image plane, $S(k)$ is a shape descriptor, and $\iota(k) = \sqrt{\sum_{i \in \mathcal{C}^S(k)} C_i(k)^2}$ is the energy of intensity values $\mathcal{C}^S(k)$ within the region defined by

$S(k)$. T is the transpose operator of a matrix. Without loss of generality, we use an elliptical shape [11], [12] and consider $S(k) = (r_a, r_b, \theta(k))$, where the scalar values r_a and r_b are the major and minor semi-axis, respectively, and $\theta(k)$ is the orientation.

Given $\mathcal{Z}(k)$ for k=1, ..., K, tracking temporally associates detections to generate trajectories. Let $\mathcal{T} = \{\mathsf{T}_a\}_{a=1}^{A}$ be the set of temporally-ordered trajectories, where T_a is the a^{th} trajectory with an arbitrary duration and A is the total number of trajectories. The smaller a, the earlier the starting frame of the trajectory.

IV. Detector

A. Gradient-climbing based detector

The detection of similar targets on high-density videos is performed by exploiting target-intensity maps using the intensity-gradient information. We assume that homogeneous high-intensity values (peaks) of the target-intensity map are measurements and such a map usually contains broad peaks. We use the intensity gradient to localize and discriminate targets since it helps the enhancement of spatial gaps among nearby targets. In the case of partially-overlapping targets, the gradient can also help the fitting of a prior shape model by exploiting visible parts of both the targets involved in the occlusion (the occluding and occluded target). The target localization is achieved by using an iterative algorithm that finds the best fit between the target shape and the target intensity, while disregarding distractors due to nearby targets and multi-peak intensities (e.g. region 1 in Fig. 1g and 1m).

The method generates detections over the intensity map while discarding those located near local maxima. This step enables the reduction of the computational complexity of the subsequent steps. Since at this stage there is no knowledge about the orientation of the targets and their location, detections are initialized with a square region for each pixel i (i.e. a single detection is initialized for each $C_i(k)$ where the number of pixels I is large). The square region allows us to start the detection process with a simple dummy shape, which is a computationally effective solution to get rid of a few candidate target locations with low intensity values. This process formulates dummy detections as $\mathbf{d}_i(k) = [x_i(k)\ y_i(k)\ r_a\ C_i(k)]^T$ at frame k, with $\mathcal{D}(k) = \{\mathbf{d}_i(k)\}_{i=1}^{I}$ and r_a is the side of a square region centered at $(x_i(k), y_i(k))$. We use the Non-Maxima Suppression (NMS) algorithm [8], [26] to remove detections that are not part of local maxima from the set $\mathcal{D}(k)$. In this process it is essential to set an overlap value (τ_{nms}) that defines how much overlap can occur within (but not between) subsets of detections. Subsets of detections are then generated using τ_{nms}. NMS outputs the detection with the highest intensity values $C_i(k)$ within (and for) each subset. The set of detections given in output is defined as $\tilde{\mathcal{Z}}_1(k) = \{\tilde{\mathbf{z}}_j^1(k)\}_{j=1}^{J}$, with $J(\ll I)$ the number of detections surviving after NMS.

The survived detections $\tilde{\mathcal{Z}}_1(k)$ are subsequently made to align on the actual target locations with MCMC [13], exploiting the prior shape information $S(k)$. In the case of high densities of targets, MCMC can probabilistically reach equilibrium for a large-state spaces with an unknown distribution.

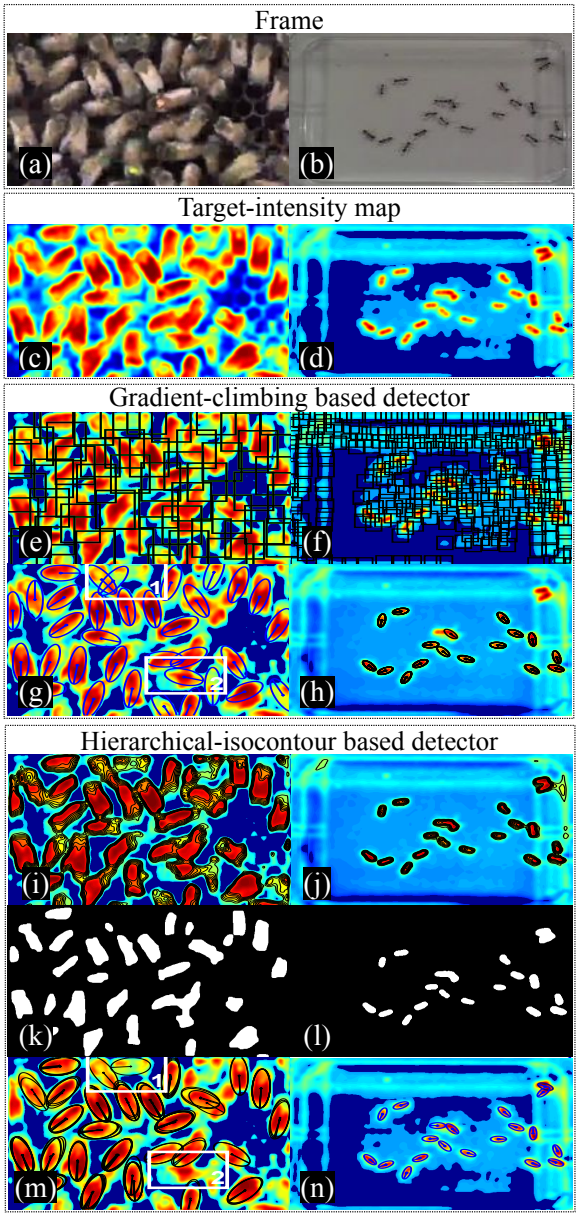

Fig. 1. Detection process using the gradient-climbing based detector and hierarchical-isocontour based morphology: (a,b) input frames; (c,d) maps representing the enhanced target intensities (target-intensity map); (e,f) mid-level step where detections are initialized using the Non-Maxima Suppression algorithm on the target-intensity map; (g,h) resulting detections obtained with the proposed approach based on MCMC. (i,j) Multi-layer isocontours on the target-intensity maps; (k,l) hole filling and erosion followed by dilation applied at intensities with values 0.7 and 0.6, respectively; (m,n) detections obtained evaluating shape properties on each region before Mean-Shift clustering. (g,m) Region 1 and region 2 highlight challenging cases that can be addressed by taking into account the advantages of the two detectors.

Let $\mathcal{Z}_1(k) = \{\mathbf{z}_{1,m}\}_{m=1}^{M_1(k)}$ be the subset of final detections $\mathbf{z}_{1,m}(k)$ generated with MCMC, where $M_1(k)$ is the number of detections at k. $\mathbf{z}_{1,m}(k)$ has the same elements of Eq. 1. Each $\mathbf{z}_{1,m}(k)$ is generated by relying on the matching between its distribution $p(\mathbf{z}_{1,m}(k))$, computed using the intensities of $\mathcal{C}^S(k)$, with that expected by the prior intensity distribution of a single target $\mathcal{P}(\mathbf{z}_{1,m}(k)) = \mathcal{N}(\boldsymbol{\mu}_\mathcal{P}, \boldsymbol{\Sigma}_\mathcal{P})$, where $\boldsymbol{\mu}_\mathcal{P} = (x_{1,m}(k), y_{1,m}(k))$ is the mean location and the covariance $\boldsymbol{\Sigma}_\mathcal{P}$ is a function of $(r_a, r_b, \theta_{1,m}(k))$, which represents the

target extent. Although in our case $\mathcal{P}(\cdot)$ follows a 2D Gaussian distribution, it can be substituted with another function in the case of different intensity distributions.

With MCMC the detections $\tilde{\mathcal{Z}}_1(k)$ tend to align to the actual target locations and orientations according to $S(k)$ and $\mathcal{P}(\mathbf{z}_{1,m}(k))$, so that it is possible to obtain the final set $\mathcal{Z}_1(k)$ such that $p(\mathcal{Z}_1(k))$ is the equilibrium distribution. The goal is to take each detection $\tilde{\mathbf{z}}_{1,j}(k) \in \tilde{\mathcal{Z}}_1(k)$, to propose a move and to validate it using a likelihood function in order to create $\mathcal{Z}_1(k)$. We employ the Metropolis-Hastings (M-H) algorithm, which enables inference of the global distribution $p(\mathcal{Z}_1(k))$ by sampling from an unknown multi-dimensional distribution with multi-dimensional states. Let $\mathbf{z}^h_{1,m}(k)$ define the h^{th} iteration (move) of a proposed detection and $\mathcal{H}$ the total number of iterations. The initialization of M-H, i.e. h=0, is done for each m such that $\mathbf{z}^0_{1,m}(k) = \tilde{\mathbf{z}}_{1,j}(k)$. M-H moves the detection $\mathbf{z}^h_{1,m}(k)$ to a new detection $\mathbf{z}^{h+1}_{1,m}(k)$ using a proposal density $q(\mathbf{z}^{h+1}_{1,m}(k)|\mathbf{z}^h_{1,m}(k), \mathcal{C}(k))$, only if $\gamma \leq \alpha$, where $\gamma \sim \mathcal{U}[0,1]$ and α is the acceptance probability

$$\alpha = \min\left(1, \frac{p(\mathbf{z}^{h+1}_{1,m}(k)|\mathcal{C}(k))}{p(\mathbf{z}^h_{1,m}(k)|\mathcal{C}(k))}\right), \tag{2}$$

with

$$p(\mathbf{z}^{h+1}_{1,m}(k)|\mathcal{C}(k)) = p(\mathcal{C}(k)|\mathbf{z}^{h+1}_{1,m}(k)) q(\mathbf{z}^{h+1}_{1,m}(k)|\mathbf{z}^h_{1,m}(k), \mathcal{C}(k)), \tag{3}$$

where $p(\mathcal{C}(k)|\mathbf{z}^{h+1}_{1,m}(k))$ is the likelihood function.

The proposal density $q(\cdot)$ defines the dynamic model

$$\mathbf{z}^{h+1}_{1,m}(k) = F^h_{1,m}(k)\mathbf{z}^h_{1,m}(k) + w^h_m(k), \tag{4}$$

where $w^h_m(k) \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_w)$, and $F^h_{1,m}(k)$ is a linear transformation dependent on iterations and time

$$F^h_{1,m}(k) = \begin{bmatrix} \begin{bmatrix} 1 + \frac{u^h_{1,m}}{x^h_{1,m}} & 0 \\ 0 & 1 + \frac{v^h_{1,m}}{y^h_{1,m}} \end{bmatrix} & \mathbf{0}_{2\times 4} \\ \mathbf{0}_{4\times 2} & \mathbf{I}_{4\times 4} \end{bmatrix}, \tag{5}$$

where $(u^h_{1,m}, v^h_{1,m})$ is a translation vector. To calculate the translation vector, we firstly compute the normalized cross-correlation [28] between $\mathcal{P}(\mathbf{z}^h_{1,m}(k))$ and a square patch, taken from $\mathcal{C}(k)$, with center $(x^h_{1,m}, y^h_{1,m})$ and extent $r_a \times r_a$. Then, the vector that goes from $(x^h_{1,m}, y^h_{1,m})$ to the point with the maximum intensity in the cross-correlation defines the translation vector. $\mathbf{0}_{\mathfrak{n}'\times\mathfrak{m}'}$ is a matrix of zeros and $\mathbf{I}_{\mathfrak{n}'\times\mathfrak{m}'}$ is the identity matrix with $\mathfrak{n}'$ rows and $\mathfrak{m}'$ columns. The noise $w^h_m(k)$ added to the dynamic model accounts for inaccurate estimations when the translation vector is calculated. The translation vector may perform a coarse shift of a detection from a region with low intensity values to a region with high intensity values. In order to refine the positioning of the detection on the top of a target, we need to add some noise to the linear transformation in order to explore different locations and to find that providing the highest likelihood. Generally, with MCMC, the proposal distribution (which proposes a new state) includes a noise term that enables the exploration of the distribution (that is unknown).

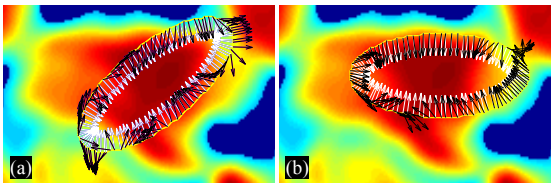

Fig. 2. Example of detection alignment using the error between the vectors of the gradient on the ellipse perimeter (black vectors) and normal vectors to the ellipse perimeter (white vectors). The goal is to minimize the error between the two sets of vectors: (a) case with a larger error due to misalignment; (b) case with a smaller error.

The likelihood function $p(\mathcal{C}(k)|\mathbf{z}^{h+1}_{1,m}(k))$ is calculated through Maximum A Posteriori (MAP) by varying the orientation $\theta^{h+1}_{1,m}(k)$ within the interval Θ=$[0,\pi]$ of the translated detection $\mathbf{z}^{h+1}_{1,m}(k)$. $p(\mathcal{C}(k)|\mathbf{z}^{h+1}_{1,m}(k))$ employs (i) the 2D gradient $\nabla\mathcal{C}(k)$ and (ii) Kullback-Leibler (K-L) divergence $d_{\text{K-L}}(\cdot||\cdot)$ [29]. The former is calculated as

$$\nabla\mathcal{C}(k) = \frac{\partial\mathcal{C}(k)}{\partial x}\hat{x} + \frac{\partial\mathcal{C}(k)}{\partial y}\hat{y}, \tag{6}$$

and enables directional alignment of the local vectors of $\nabla\mathcal{C}(k)$ for each target to the normal vectors of the perimeter of $S(k)$ (Fig. 2). The latter enables us to find the orientation within Θ that minimizes the divergence between the local intensity distribution of $\mathcal{C}(k)$ at iteration $h+1$ and the rotated version of $\mathcal{P}(\mathbf{z}^{h+1}_{1,m}(k,\theta))$. In particular, we use the gradient normalized to unit vectors namely $\nabla\bar{\mathcal{C}}(k)$. Specifically, we have

$$p(\mathcal{C}(k)|\mathbf{z}^{h+1}_{1,m}(k)) = \arg\max_{\theta\in\Theta}\left[p(\mathcal{C}(k)|\mathbf{z}^{h+1}_{1,m}(k,\theta))\right] = \arg\max_{\theta\in\Theta}\left[\exp\left(-\frac{1}{2}\left(\frac{E(\nabla\bar{\mathcal{C}}(\mathbf{z}^{h+1}_{1,m}(k,\theta)), \vec{\mathcal{E}}(\theta))}{\sigma_{\mathcal{C}}}\right)^2\right)\cdot \exp\left(-\frac{1}{2}\left(\frac{d_{\text{K-L}}(\mathcal{N}(\boldsymbol{\mu}_L, \mathbf{\Sigma}_L)||\mathcal{P}(\mathbf{z}^{h+1}_{1,m}(k,\theta)))}{\sigma_{\text{K-L}}}\right)^2\right)\right], \tag{7}$$

where, with a simplified notation, the argument θ indicates the rotated version of the state. The step size for θ is $\pi/8$. $\nabla\bar{\mathcal{C}}(\mathbf{z}^{h+1}_{1,m}(k,\theta))$ is the 2D gradient of $\mathcal{C}(k)$ corresponding to the pixels adjacent to the perimeter $\mathcal{E}(\theta)$, $\vec{\mathcal{E}}(\theta)$ are the normal vectors of the θ-rotated ellipse perimeter (Fig. 2), and $\sigma_{\mathcal{C}}$ and $\sigma_{\text{K-L}}$ are constants. $\boldsymbol{\mu}_L$ and $\mathbf{\Sigma}_L$ are the components obtained by fitting a 2D Gaussian [30] in the domain $r_a \times r_a$ of $\mathbf{z}^{h+1}_{1,m}(k)$ on $\mathcal{C}(k)$. $E(\cdot)$ quantifies the orientation error of the ellipse with respect to the direction of the gradient. The goal is to minimize the error between $\nabla\bar{\mathcal{C}}(\mathbf{z}^{h+1}_{1,m}(k,\theta))$ and $\vec{\mathcal{E}}(\theta)$:

$$E(\nabla\bar{\mathcal{C}}(\mathbf{z}^{h+1}_{1,m}(k,\theta)), \vec{\mathcal{E}}(\theta)) = ||\nabla\bar{\mathcal{C}}(\mathbf{z}^{h+1}_{1,m}(k,\theta)) - \vec{\mathcal{E}}(\theta)||_2, \tag{8}$$

where $||\cdot||_2$ is the ℓ-2 norm. When all the iterations are performed by M-H, multiple detections may converge to the same target. In order to suppress these detections, we apply NMS on the converged detections using τ_{nms} on the overlap to obtain $\mathcal{Z}_1(k)$ (Fig. 1g,h).

B. Hierarchical-isocontour based morphology

Targets may appear at different intensity levels within the same frame (e.g. due to illumination changes) and a single intensity level may not be enough to separate all the targets.

In order to distinguish adjacent targets, we "slice" the intensity map at different intensity levels (isocontours). Each level allows inferring shape properties of targets.

Let $\mathcal{Z}_2(k) = \{\mathbf{z}_{2,m}\}_{m=1}^{M_2(k)}$ be the subset of detections $\mathbf{z}_{2,m}(k)$ inferred with this method, where $M_2(k)$ is the number of detections. $\mathbf{z}_{2,m}(k)$ has the same elements of Eq. 1.

Let $\mathcal{I}_{\tau_{iso}}(k)=g_{\tau_{iso}}(\mathcal{C}(k))$, with $\tau_{iso}\in[0,1]$, be the isocontours extracted from the target-intensity map $\mathcal{C}(k)$ at layer τ_{iso}, where the function $g_{\tau_{iso}}(\cdot)$ computes the isocontours [31] on $\mathcal{C}(k)$. $\tau_{iso}\to 0$ might provide large regions encapsulating multiple adjacent targets, whereas $\tau_{iso}\to 1$ might provide small regions with high intensity values, with the possibility of discarding targets with low intensity values. In order to detect targets appearing at different intensity levels, isocontours are computed by ranging τ_{iso} in the interval Ω (multiple layers) and the discretization of the values within Ω can be manually chosen. To separate regions connected by thin segments and to filter out background clutter, each layer $\mathcal{I}_{\tau_{iso}}(k)$ is processed with morphological operators, which include hole filling [32] followed by erosion and dilation [28]. At each τ_{iso}, we select the connected regions considering shape information [33]. We use eccentricity for the elliptic model. We select target regions with an eccentricity equal or greater than 0.75. The selected regions are then used to determine the detections of the initial set $\tilde{\mathcal{Z}}_2(k)$. Each $\tilde{\mathbf{z}}_{2,m}(k)$ has the same elements as those in Eq. 1 and their values are defined according to the following properties: $(\tilde{x}_{2,m}(k), \tilde{y}_{2,m}(k))$ is defined by the region centroid, $\tilde{\theta}_{2,m}$ is the region orientation, $\tilde{\iota}_{2,m}(k)$ is initialized at zero value and r_a, r_b are defined a priori.

Because extracting regions at multiple layers of isocountours may lead to multiple spatially-close detections for each target, we cluster detections in order to remove redundant detections. We use Mean-Shift (MS) [18] to cluster neighboring detections by using the position information of the detection of $\tilde{\mathcal{Z}}_2$, without any prior knowledge on the number of clusters and with a fixed kernel size.

Let the kernel size be r_b and the set of clusters $\Psi(k) = \{\psi_r(k)\}_{r=1}^{\mathcal{R}(k)}$, with $\psi_r(k)$ the generic r^{th} cluster and $\mathcal{R}(k)$ the set of cluster indexes. For each ψ_r, we generate a detection $\mathbf{z}_{2,m}(k)$ whose position $(x_{2,m}(k), y_{2,m}(k))$ coincides with the centroid position of the cluster. The orientation $\theta_{2,m}(k)$ is calculated as the circular median [34] of the states belonging to the cluster and the energy $\tilde{\iota}_{2,m}(k)$ is calculated as in Eq. 1 within the region defined by the ellipse with parameters $(r_a, r_b, \theta_{2,m}(k))$ and centered in $(x_{2,m}(k), y_{2,m}(k))$.

Fig. 1i,k,m and Fig. 1j,l,n show examples of the method on bees and ants, respectively.

C. Pruning

Situations with target intensities having multiple peaks may lead to inaccurate alignments of detections due to gradient variations in the target region. Moreover, adjacent targets may appear as large irregular connected regions that do not fulfill the prior shape constraints. In particular, the gradient-climbing based detector is likely to fail when targets are characterized by multi-peak intensity values. This can be observed in Fig. 1g,m rectangle 1. When a target has multi-peak intensities the local gradient is non-homogeneous in the target region. The gradient direction follows peaks and valleys of the intensity, which leads to different detection configurations due to the low similarity between the gradient in the image and the normal vectors of the ellipse perimeter. The isocontour-based detector seeks regions with connected components throughout different intensity values and the regions dissatisfying the prior shape constraints (e.g. eccentricity) are discarded. In the case of multi-peak intensities, the low intensity values outlining a target may provide regions fulfilling the constraints.

At high intensity values the isocontours enclose multiple regions due to the multiple peaks and it is likely that these regions will be discarded because they do dissatisfy the prior shape constraints. Therefore, only the regions selected at low intensity values are considered as valid target detections. Vice versa, adjacent targets can be effectively detected with the gradient-climbing based detector. This can be observed in Fig. 1g,m rectangle 2. When adjacent targets have intensity values connected, the gradient corresponding to the disconnected parts of each target can be exploited to align the detections. The isocontour-based detector may fail with adjacent targets because it is likely that the isocontours would enclose all the targets together for all the intensity values and the prior shape constraints would not be fulfilled. Therefore, the detections generated using the gradient are accounted for valid target detections.

The selection of valid detections begins by merging the results of gradient-climbing based and hierachical-isocontour based detectors, $\hat{\mathcal{Z}}(k) = \mathcal{Z}_1(k) \cup \mathcal{Z}_2(k) = \{\hat{\mathbf{z}}_n(k)\}_{n=1}^{\hat{N}(k)}$. We then eliminate the remaining false positives and repeated detections of the two methods. As done in Sec. IV-B, we cluster neighboring detections of $\hat{\mathcal{Z}}(k)$ using MS with a kernel of size equal to the minor semi-axis r_b. Let $\hat{\Psi}(k) = \{\hat{\psi}_r(k)\}_{r=1}^{\hat{\mathcal{R}}(k)}$ be the resulting set of clusters, where $\hat{\mathcal{R}}(k)$ is the number of clusters in frame k. For each cluster $\hat{\psi}_r(k)$, a single detection $\mathbf{z}_n(k)$ is selected with the highest $\hat{\iota}(k)$, such that

$$n = \underset{n^*\in\hat{\psi}_r(k)}{\arg\max}(\hat{\iota}_{n^*}(k)), \tag{9}$$

where $\hat{\iota}_{n^*}(k)$ is the $\hat{\iota}(k)$ term defined within each $\hat{\psi}_r(k)$. $\mathcal{Z}(k)$ will therefore be composed of the detections $\hat{\mathbf{z}}_n(k)$ with largest $\hat{\iota}_{n^*}(k)$ from each cluster.

The block diagram depicting the main steps of the target detector is shown in Fig. 3.

V. Greedy-graph based association

A. Tracking formulation

The association process links detections over time, predicts detections in frames with miss-detections and prunes false detections. Short tracks are initially generated by associating only detections with high similarity in order to reduce potential errors while reducing the complexity of the overall problem [12]. We use target-position information for the generation of tracks.

Let the set of short tracks, $\mathfrak{T} = \{\mathfrak{t}_b\}_{b=1}^{B}$, be generated by optimally associating consecutive detections using Munkres (Hungarian) algorithm [22]. We will refer to the

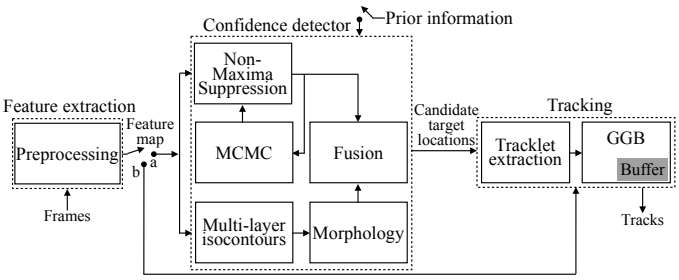

Fig. 3. The main stages for target detection and tracking. Tracking can be either performed using candidate target locations (switch is on "a") or on confidence maps (switch is on "b"). Prior information (user intervention) can be used to improve the confidence detection. The pipeline of the proposed approach has the prior information input off and the switch on "a". Key. MCMC: Monte Carlo Markov Chain; GGB: Greedy-Graph Based.

states belonging to a generic t_b as $\mathbf{z}_b(k)$ for those k where t_b exists. The sequential association is performed while keeping unique identities to the associated detections. Thus for each pair $(\mathcal{Z}(k), \mathcal{Z}(k+1))$ we calculate the cost $\mathfrak{C}_{k,k+1} \in \mathbb{R}^{\bar{N}(k)\times\bar{N}(k+1)}$ using the ℓ-2 norm between each position state in frame k and $k+1$, $\mathfrak{c}_{k,k+1}^{n,n'} = ||(x_n(k), y_n(k))^T - (x_{n'}(k+1), y_{n'}(k+1))^T||_2$, where $\mathfrak{c}_{k,k+1}^{n,n'}$ is the element of $\mathfrak{C}_{k,k+1}$ on the row n and column n'.

Longer tracks $\mathcal{T} = \{\mathsf{T}_a\}_{a=1}^{A}$ are generated by sequentially linking short tracks as a MAP problem [12],

$$\mathcal{T}^* = \arg\max_{\mathcal{T}} p(\mathcal{T}|\mathfrak{T}), \tag{10}$$

with $\mathcal{T}^*$ the set of tracks with the highest probability. The direct maximization of Eq. 10 is computationally expensive because the number of combinations of the elements of the set $\mathfrak{T}$ is large [12]. We decompose the problem as

$$p(\mathcal{T}|\mathfrak{T}) = p(\mathsf{T}_1|\mathfrak{T}) \cdot p(\mathsf{T}_2|\mathfrak{T} \setminus \mathsf{T}_1) \cdot p(\mathsf{T}_3|\mathfrak{T} \setminus \mathsf{T}_2, \mathsf{T}_1) \cdot \\ \ldots \cdot p(\mathsf{T}_A|\mathfrak{T} \setminus \mathsf{T}_{A-1}, ..., \mathsf{T}_a, ..., \mathsf{T}_1), \tag{11}$$

and we maximize each probability term iteratively with a greedy process. This enables us to perform tracking within a short temporal buffer and, unlike [16] or [12], once a trajectory is computed within the buffer, we do not change the solution afterwards.

B. Greedy-graph solution

Methods exploiting dynamic programming, for example the Viterbi algorithm [35], can find the global optimal solution that maximizes the problem in Eq. 10 by dividing the overall problem into simpler subproblems. The Viterbi algorithm for multi-target tracking assumes that all the nodes (in our case the short tracks) of a graph should be linked to each other and the links should be unmerged. Since the graph may contain many false positive nodes, the use of the Viterbi algorithm would lead to a wrong association of nodes, for example when all the good nodes are connected and only false positive nodes are left. The Viterbi algorithm would also connect these false positive nodes which would result in an increase in false positive tracks. Therefore, we use a greedy graph-based (GGB) method that enables the linkage of short tracks by discarding false positives and by introducing latency.

Algorithm 1 *Greedy graph-based association*

$\mathfrak{T}$: set of temporally-ordered short tracks. $\ell(\mathrm{t}_{b'}|\mathrm{t}_b)$: link probability.
τ_ℓ: threshold for negligible link probabilities. (Ξ, ξ): buffer size and temporal shift.
$\mathfrak{T}_{proc}$: processed short tracks. B_Ξ: number of short tracks within the buffer Ξ.

```
T ← ∅; T_proc ← ∅
for b ← 1 to B_Ξ do
    T_temp ← ∅; t_b ← T
    if t_b ∉ T_proc then
        T_temp ← t_b
        while (1) do
            T_τℓ ← findnodes s.t. {ℓ(t_f^- | t_b^+) > τ_ℓ, t_f ∉ T_proc, t_f ∈ T, f > b}
            if T_τℓ ≠ ∅ then
                while 1 do
                    t_b' = arg max_{t_f ∈ T_τℓ} ℓ(t_f^- | t_b^+)
                    if (arg max_{t_f ∈ T_τℓ} ℓ(t_f^+ | t_b^-) = t_b) || (t_b' = ∅]) then
                        T_temp ← t_b'; t_b ← t_b'
                        break while
                    else
                        T_τℓ ← T_τℓ \ t_b'
                    end if
                end while
            else
                break while
            end if
        end while
        T_proc ← T_temp; T ← g(T_temp)
    else
        T_proc ← t_b; T ← g(t_b)
    end if
end for
```

Let $G = (E, \mathfrak{T})$ be a graph, where E is the set of edges whose weights are calculated via a link probability and $\mathfrak{T}$ are the nodes. Each node is composed of a sink (child) and a source (parent), denoted as t_b^- and t_b^+, respectively. We define a function $g(\cdot)$ that links short tracks by performing a non-linear interpolation of the positions in order to generate detections among linked short tracks and to smooth tracks $\mathcal{T}$. Eq. 11 can be solved by formulating the problem with a graph and using the concept of *parents* and *children*. A *parent* is a short track that ends before the start of another one, which in turn is defined as a *child*. A parent can be associated with a child when the likelihood (weight) from the parent to the child is the biggest for the parent and also the biggest for the child with respect to other competitive (or candidate) parents. We aim to associate parents and children with a forward association and a backward validation, in order to achieve the best association between two short tracks with respect to the competing candidates. Hence, we iteratively and pair-wisely match parents and children over time until there are no alternative pairings in which the single best match is found between each parent and child.

Eq. 11 assumes that each track is temporally dependent on another track conditioned upon their initial state: $s_b \leq s_{b+1} \leq \ldots \leq s_B$, where s_b denotes the frame of the initial state of a generic t_b and likewise for e_b denoting the frame of the last state. In fact, the calculation of T_a depends on $\mathfrak{T}$, except those used for the calculation of T_{a-1}, which in turn depends on $\mathfrak{T}$, except those used for T_{a-2}, and so on. Each probability term of Eq. 11 is maximized via a recursive process using a link probability $\ell_1(\cdot)$ between short-track pairs $(\mathrm{t}_b, \mathrm{t}_{b'})$,

$$\ell_1(\mathrm{t}_{b'}|\mathrm{t}_b) = \begin{cases} \exp -\frac{1}{2}\left[\left(\frac{\beta_{b',b}}{2\sigma_1}\right)^2 + \frac{(s_{b'}-e_b)^2}{2\sigma_2}\right] & \text{if } e_b - s_{b'} < \tau_b \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

where τ_b is the temporal interval permitted between the end

of a short track and the start of another, $\beta_{b',b}$ is a normalized ℓ-2 norm

$$\beta_{b',b} = \frac{1}{r_a}||(x_b(e_b), y_b(e_b))^T - (x_{b'}(s_{b'}), y_{b'}(s_{b'}))^T||_2, \quad (13)$$

and σ_1, σ_2 are constants. We use a normalized distance over the shape model, so that σ_1 is target-size independent.

The linking process is performed within a buffer of duration Ξ frames, implemented as a sliding window approach with ξ overlapping frames and with $\tau_b > 0$. The linkage method is described in Algorithm 1.

Temporally-overlapping tracks for short periods of time can occur when concurrent detections are generated for a single target. This is problem is usually addressed with NMS at the detection stage [8]. However, NMS may fail if the overlap of the detections is small. Hence, by employing this additional analysis in GGB, we can reduce the risk of track fragmentation[1] due to concurrent detection on the same target. Hence, after having generated the set $\mathcal{T}$, we reapply the Algorithm 1 using $\mathfrak{T} = \mathcal{T}$ and the likelihood function

$$\ell_2(\mathfrak{t}_{b'}|\mathfrak{t}_b) = \begin{cases} \exp -\frac{1}{2}\left(\frac{\bar{\beta}_{b',b}}{2\sigma_1}\right)^2 & \text{if } \tau_o \leq s_{b'} - e_b \leq 0 \\ 0 & \text{otherwise} \end{cases} \quad (14)$$

where $\tau_o \leq 0$ is the number of overlapping frames and

$$\bar{\beta}_{b',b} = \frac{1}{|s_{b'} - e_b|}\frac{1}{r_a} \cdot \sum_{\gamma=1}^{|s_{b'}-e_b|-1} ||(x_b(e_b-\gamma), y_b(e_b-\gamma))^T - (x_{b'}(s_{b'}+\gamma), y_{b'}(s_{b'}+\gamma))^T||_2. \quad (15)$$

Finally, we prune tracks within the buffer Ξ with a shorter duration than τ_D.

VI. Computational complexity

We analyze the computational complexity of the main stages of the proposed pipeline (Fig. 3). The proposed pipeline is implemented in Matlab and the code is not optimized. The gradient-climbing based detector involves three main steps: Non-Maxima Suppression with dummy (square) shape, MCMC alignment and Non-Maxima Suppression of the aligned detections. The first step has complexity $O((\frac{I}{(1-\tau_{nms})r_a^2})^2)$ where I is the total number of pixels. The MCMC alignment has complexity $O(\frac{I}{(1-\tau_{nms})r_a^2}\mathcal{H}r_a^2r_b^2)$ where $r_a^2r_b^2$ is due to the normalized cross-correlation between prior and observations, and $\mathcal{H}$ is the number of iterations. The final Non-Maxima Suppression requires $O((\frac{I}{(1-\tau_{nms})r_ar_b})^2)$ since it is performed with elliptical-shape detections. The hierarchical-isocontour based morphology has complexity $O(N_\Omega R)$, where N_Ω is the number of thresholds chosen within $[0, 1]$ and R is the number of regions detected at each layer. Usually R is slightly larger than the number of targets in the frame. We considered filling, erosion and dilation operations as O(1) since they are optimized and embedded in Matlab. The Mean-Shift clustering has complexity $O(N_\Omega R' log(N_\Omega R'))$, where R' is the number of selected regions with eccentricity bigger than the threshold. The optimal association of detections performed with Munkres algorithm has complexity $O(\max(N(k)^3, N(k+1)^3))$, where $N(k)$ is the number of detections at frame k. The greedy-graph based algorithm (Algo. 1) in the worst case scenario (when all nodes are connected) has complexity $O(B_\Xi^2 log(B_\Xi))$ obtained with B_Ξ operations for spanning each node (short track) and multiplied by $B_\Xi log(B_\Xi)$ for the selection of the edge with largest cost that is employed with a sorting algorithm. B_Ξ is the number of short tracks (nodes) within the temporal window Ξ. However, the graph is generally sparse and the average complexity is much lower than the worst case.

VII. Results

A. Experimental setup

1) Methods under comparison: We compare our detector with four detection approaches (D) followed by six alternative trackers (T). D1: threshold based plus Mean-Shift (MS) clustering applied on $\mathcal{C}(k)$ (similar to [3]). D2: D1 followed by Non-Maxima Suppression (NMS). D3: template matching on grayscale frames via normalized cross-correlation using eight target patches at different orientations cropped from the videos. D4: D3 followed by NMS. D5: maximally stable extremal regions (MSER) [36] with MS clustering and NMS. T1: a baseline hierarchical detection association [37] where the detections are associated frame-by-frame with the Hungarian algorithm in order to generate short tracks, which are further globally associated using the nearest neighbor algorithm. T2: a multi-particle tracker that employes Brownian motion as prior on the target motion [38]. The detection association is done by maximizing the probability of finding each target between one frame and the next. T3: based on multiple Kalman filters used to predict and update the locations of the targets [21]. The prediction is performed with a linear motion model, and the association between detections and trackers is performed with the Munkres algorithm. T4: formulated as an energy minimization problem between piecewise polynomials (B-splines) and target trajectories [24]. T5: a multi-target track-before-detect. T6: T5 with postprocessing stage with a buffer of 50 frames [2], as with the proposed approach.

2) Datasets and parameters: We use a bee dataset (B-D), an ant dataset (A-D) and a cell dataset (C-D)[2]. The first two are quantitatively and qualitatively evaluated, the third one is only qualitatively evaluated. B-D is composed of 28400 frames of size 640×350 and recorded at 29.97 frame-per-second (fps) from a moving camera. We use two clips of video footage extracted from B-D to quantitatively evaluate the detector and the tracker, namely B-D1 (frames 500 to 999) and B-D2 (frames 25500 to 25999). B-D1 contains in total 81 targets with an average of 31 targets per frame. B-D2 contains in total 64 targets with an average of 32 targets per frame. Targets interact, occlude each other and move with sudden changes in directions. A-D (10400 frames, size 720×480, 29.97 fps, static camera) contains 20 targets always present and undergoing several occlusions. C-D (131 frames,

[1]This occurs when a track terminates in a frame and restarts with another identity after a few frames.

[2]Tracking results and datasets: http://www.eecs.qmul.ac.uk/~andrea/thdt.html

size 340×240, 29.97 fps, moving camera) has an average of 34 targets/frame moving quickly in unpredictable ways. There are variations of illumination and of camera focus.

In B-D, the target-intensity map is the equalized red channel (RGB colorspace) of the frames with a Gaussian filter applied to it. In our experiments, the red channel is found to be a reliable feature since most of the information about the color of bees lies on this channel. By observing the size of the objects on the image plane we set the ellipse prior size to r_a=42, r_b=18. The threshold used for NMS is τ_{nms}=0.3. The number of iterations for MCMC is $\mathcal{H}$=10 and the likelihood function used for its computation has variance parameters set as $\sigma_{\mathcal{C}}$=1 and $\sigma_{\text{K-L}}$=0.7; these values of iterations and standard deviation provide an accurate shape alignment. Smaller values of $\sigma_{\mathcal{C}}, \sigma_{\text{K-L}}$ and larger number of iterations do not further improve the fitting accuracy, whereas larger values of $\sigma_{\mathcal{C}}, \sigma_{\text{K-L}}$ and smaller number of iterations might provide less accurate alignments. τ_{iso} values range in the interval Ω=[0.5 0.8] with step size of 0.05. The link probability of Eq. 12 has σ_1=0.3, to penalize detections outside the region of the target, and σ_2=10, to link detections for short temporal gaps. The buffer is Ξ=50 frames with a temporal shift ξ=5 frames, τ_b=10 τ_o=−10 and τ_D=15. In A-D the target-intensity map is the grey-level image and is filtered with a Gaussian function. The parameters are the same as for B-D apart from the ellipse size prior, r_a=16, r_b=7, and τ_D=30. As for A-D, in C-D the target-intensity map is the grey-level image and is filtered with a Gaussian function. The parameters are the same as for B-D apart from the ellipse size prior, r_a=8, r_b=4, and τ_D=5. A sensitivity analysis of the detector and tracker parameters is also performed. The computer used to run the experiments has CPU Intel i5 2.4GHz dual-core with 8GB RAM. We use B-D1 to compute the execution time per frame of detection and tracking algorithms.

B. Evaluation measures

We evaluate the performance of the detector and tracker in terms of Precision (P), Recall (R), F-Score (F) and robustness to ID switches. P and R measure the accuracy of the tracking that quantifies the closeness of agreement between estimated and ground-truth target locations [39]. ID switches quantify the robustness of the tracker in distinguishing targets throughout the sequence. Precision is calculated as $P = \frac{T_p}{T_p+F_p}$ and Recall as $R = \frac{T_p}{T_p+F_n}$, where T_p is the number of true positive tracks of the sequence, F_p the number of false positive tracks and F_n the number of false negative tracks. F-Score is calculated as $F = 2\frac{P \cdot R}{P+R}$. The association between target estimation and ground truth is performed on a frame-by-frame basis. A true positive happens when the distance between the estimated location of a target and its ground truth is smaller than a threshold τ_{T_p}. We use τ_{T_p}=30 pixels for B-D and τ_{T_p}=10 pixels for A-D, which correspond to the 83% and 71% of the target width, respectively.

Moreover, we propose a new measure to quantify the robustness of a tracker to ID switches by using a two-element vector measure IDSR = [Γ Λ]. IDSR enables us to measure the ID switches per frame, Γ, and ID switches per track, Λ.

TABLE II
DETECTION RESULTS. THE THRESHOLD ON THE DISTANCE USED TO DEFINE A DETECTION RESULT AS A TRUE POSITIVE IS 30 PIXELS FOR B-D1 AND B-D2, AND 10 PIXELS FOR A-D. KEY: D: DATASET; P: PRECISION; R: RECALL. D1-D5: ALTERNATIVE DETECTORS.

Target detector	B-D1			B-D2			A-D		
	P	R	F	P	R	F	P	R	F
D1 ([3]+MS)	.63	**.90**	.74	.61	.81	.70	.60	.93	.73
D2 (D1+NMS)	.80	.71	.75	.76	.66	.71	.91	.88	.89
D3 ([28])	.63	.59	.61	.70	.64	.67	.89	.78	.83
D4 (D3+NMS)	.73	.52	.61	.82	.57	.67	.91	.77	.83
D5 ([36]+MS+NMS)	.73	.81	.77	.79	.84	.83	.98	.91	.94
Gradient-based	.80	.89	**.84**	.86	**.90**	.88	.98	**.97**	**.98**
Isocontour-based	**.92**	.73	.81	**.96**	.71	.82	.98	.93	.95
Fusion	.81	.88	**.84**	.89	.89	**.89**	**.99**	**.97**	**.98**

Γ is defined as $\Gamma = \frac{IDS}{K}$, where IDS is the total number of ID switches that occurred in the sequence and K is the total number of frames (see Sec. V). Λ is defined as $\Lambda = \sum_{k=1}^{K} \frac{i(k)}{\zeta(k)}$, where $i(k)$ is the number of ID switches and $\zeta(k)$ is the maximum number of ID switches that can occur at frame k. A small value of τ_{T_p} is more suitable to correctly evaluate IDSR. A large τ_{T_p} may lead to errors in the evaluation procedure when the optimal association between ground-truth and estimated tracks is computed.

C. Target detection and tracking

1) Detection: We compare the results obtained with the proposed method and those obtained with alternative approaches. Table II shows that on average the gradient-climbing based detector (GCD) has a higher Recall (R) than the other methods. Hierarchical-isocontour based morphology (HIM) provides the highest Precision (P) compared to the other methods. The fusion operation provides the highest F-Score (F). By fusing the results we can improve P of GCD, since some of the false positives are discarded by Eq. 9. Morphological operators can be very accurate because they can effectively filter out clutter, but they might not be able to provide reliable detections in the case of adjacent targets. Even if D1 and D2 provide reasonable results in A-D, e.g. D2 reaches F=0.89, which does not contain as challenging situations as B-D, their performance is still lower than those provided by the proposed method. Interestingly, NMS on D2 effectively prunes spurious detections in A-D, but not in B-D1 and B-D2 because NMS suppresses valid detections when the detected regions of adjacent targets overlap. Overall, template-based approaches (D3 and D4) have the lowest performance compared to the other methods, since they are unable to discriminate adjacent targets in high-density videos (e.g. B-D). The same problem occurs for D5, where MSER features cannot generate separate regions with nearby when targets are close to each other.

We also employ the detector proposed in [1] (only for A-D) followed by Mean-Shift clustering in order to obtain a single detection for each target. This method provides good P=0.98 and R=0.94, but the performance remains lower than that of the proposed method, which reaches P=0.99 and R=0.97. Overlapping and adjacent targets are, in fact, difficult to separate with the method proposed in [1].

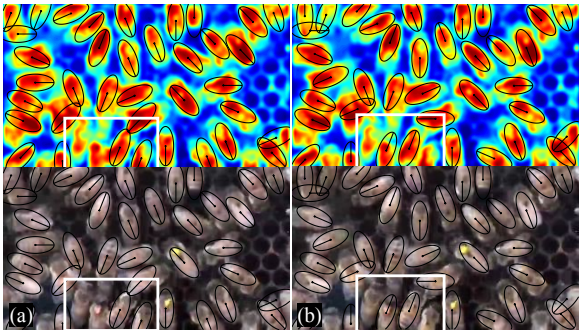

Fig. 4. Sample detections on the bee dataset superimposed on the target-intensity maps and on the respective original frames. Most of the targets are correctly detected: (a) a failure of detected targets can be spotted in the white box; this is due to its weak intensity with respect to the other targets; the target on the bottom is not detected because is partially outside the frame. (b) When the target becomes more visible it gets detected.

TABLE III
TRACKING RESULTS. THE THRESHOLD ON THE DISTANCE USED TO CONSIDER A TRACKING RESULT AS A TRUE POSITIVE IS 30 PIXELS FOR B-D1 AND B-D2, AND 10 PIXELS FOR A-D. THE RESULTS FOR T4 IN A-D ARE NOT PROVIDED DUE TO IMPLEMENTATION LIMITATIONS WITH LONG SEQUENCES. KEY: D: DATASET; P: PRECISION; R: RECALL; IDSR: ID SWITCH RATES. T1-T6: ALTERNATIVE TRACKERS. (*) T5 IS FROM [2] WITH NO POSTPROCESSING.

Trackers	B-D1				B-D2				A-D			
	P	R	F	IDSR	P	R	F	IDSR	P	R	F	IDSR
T1 ([37])	.81	.88	.84	[.60 9.53]	.89	.89	.89	[.43 6.55]	**.98**	.97	**.98**	[.09 44.50]
T2 ([38])	**.83**	.86	.84	[.42 6.67]	**.91**	.88	.89	[.26 3.92]	**.98**	.97	**.98**	[.07 34.65]
T3 ([21])	.59	**.93**	.72	[.80 12.74]	.81	**.93**	.87	[.68 10.50]	.95	**.98**	.96	[.08 42.00]
T4 ([24])	.76	.82	.79	[2.2 35.40]	.87	.90	.88	[1.05 16.16]	-	-	-	-
T5 ([2]*)	**.83**	.84	.84	[1.35 21.68]	.90	.86	.88	[1.15 17.52]	.96	**.98**	.97	[.21 109.55]
T6 ([2])	.81	.85	.83	[.29 4.70]	.90	.87	.88	[.21 3.18]	.97	.97	.97	[.13 66.50]
HA	.81	.88	.84	[1.97 31.53]	.89	.89	.89	[1.69 25.94]	**.98**	.97	**.98**	[.26 135.05]
T5+GGB	**.83**	.87	**.85**	**[.22 3.51]**	.90	.88	.89	[.17 2.61]	**.98**	.97	**.98**	[.06 30.50]
HA+GGB	.82	.89	**.85**	[.22 3.55]	.90	.91	**.91**	**[.14 2.14]**	**.98**	**.98**	**.98**	**[.03 13.40]**

Fig. 4 shows sample results with situations of miss-detected targets (white bounding box). Firstly, the top-left target has a low intensity compared to the others and it is likely that the detection is converged to one of those neighbors. Secondly, for the target at the bottom of the white box, the detection cannot converge and align to it as it is partially occluded by the neighboring target and partially outside the frame. Moreover, the morphological operations does not detect the target since its intensity values are connected with those of the right-hand-side target. When this last target moves upwards and the neighboring one moves slightly farther away, it gets detected (Fig. 4b). In both Fig. 4a and b there is a target (under the top-left corner of the white box) with intensity values more spread out than the others (a bee with open wings) and the orientation of the state inaccurately matches with that of the real target. The incorrect estimated orientation is caused by the fact that we are not using any complex prior knowledge about the targets, unlike [40], other than that they are approximated as elliptical shapes. Additional qualitative results can be observed on C-D from the video of results provided in Sec. VII-A2. Targets are often correctly detected except when their size get much smaller (about three times less) than that defined as prior.

Lastly, the median execution time for both D1 and D2 is 210.8 sec/frame; and for D3 and D4 is 33.0 and 33.5 sec/frame, respectively. D5 has the lowest median execution time per frame (0.3 sec/frame) because the algorithm for MSER is Matlab optimized. The proposed approach has the second lowest execution time: 11.5 for Gradient-based, 3.9 for Isocontour-based and 16.6 for the whole detection pipeline with Fusion included.

2) Tracking: We assess the performance of the proposed tracker and compare it with alternative trackers (Sec. VII-A1). The trackers are tested on B-D, A-D and C-D sequences using detections generated by the proposed detector.

Tracking results on B-D1 and B-D2 are shown in Table III, and we can see that overall the proposed method (HA+GGB) outperforms the other methods: on average F is the highest and ID switches are the lesser. IDSR of the GGB is ten times better than the sequential linking performed with the Hungarian algorithm (HA). T3 has the lowest P in both B-D1, B-D2 and A-D, due to the prediction step of Kalman filter (KF) when no detections are available. Specifically, in situations of abrupt motion changes of the targets, KF is unable to correctly predict the future location, and when no detections are available, the filter uses the predicted state as a valid state (Fig. 5a-c). Then, KF keeps predicting incorrect states until the error covariance becomes big enough to consider the target lost. Following that, the lost target is re-initialized with a new track. Therefore the tracks generated from spurious predictions increase the false positive rate. T2 has a higher P than that of T3 since the prior on the target motion looks closer to the actual movement of the targets (Fig. 5g-i). However, on average T1 has the same F of T2, but T2 is more accurate at correctly distinguishing target identities (lower IDSR). Similarly to T3, T5 uses a linear motion model to predict target locations. However, the update is performed using intensity values instead of detections. This enables more flexibility for the tracker to determine whether detections belong to noise or not; P is higher, while R is lower, which means that correct detections are sometimes considered clutter. The postprocessing applied on T5 (T6) dramatically improves the performance by getting very close to that of HA+GGB, especially in terms of IDSR. We also apply GGB to the short tracks generated with T5 and we can see a considerable improvement when ID switches per frame are slightly lower in the case of T5+GGB. However, since T5 has lower R than HA, T5+GGB does not achieve an R as good as that of HA+GGB. Finally T4 has very poor performance compared to the others.

Fig. 6 shows results in presence of poor illumination and low resolution (trajectories are truncated at 50 frames to make the visualization clearer). On the left-hand side of Fig. 6a, where there is a high density of targets and the resolution is low, the frame appears as a dark patch and presents artifacts due to image compression. In this region we can observe a few tracking failures that are recovered in the subsequent frames when targets get farther apart (Fig. 6b,c). The magenta arrow in Fig. 6b points on a situation of ID switch between the red identity and the gray identity. The bee with the red identity is moving from left to right and when she passes on the top of the other bee, the gray identity passes to the red one and the red identity gets lost. When the detections of the

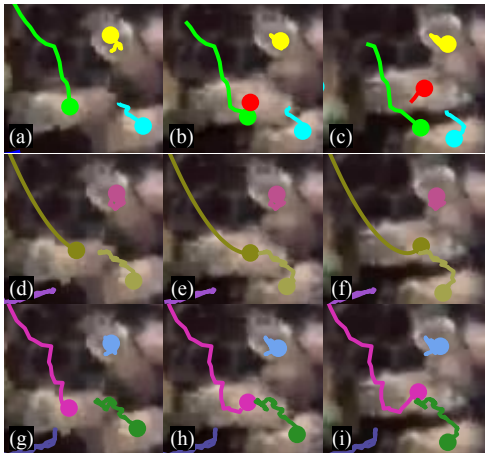

Fig. 5. Example of an abrupt motion change of a target where (a-c) a Kalman filter-based method can fail. The failure occurs on (a) the green track, where the tracked target is moving from top to bottom and (b) suddenly changes direction. The green track will keep going straight on, while the red track is initialized. (c) The track survives for a few steps before being terminated. (d-f) Multi-particle tracker based on Brownian motion and (g-i) the proposed approach can deal with abrupt motion changes. Different rows and different trajectory colors represent different tracker results.

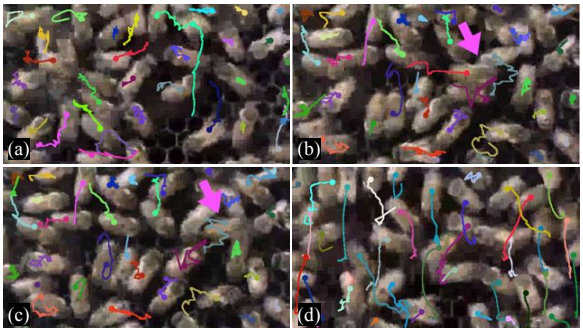

Fig. 6. Sample tracking results on dataset B-D on challenging situations. (a) High density of bees on the left-part of the frame; (b-c) ID switch of the red trajectory in the middle of the frame and gray trajectory on its path (magenta arrow); (d) Robustness of the method to camera movements. The trajectories are truncated to the last 50 frames to make the visualization clearer.

overlapped bee become available, GGB associates those of the flying bee to those of the still bee. Fig. 6d shows a case with an abrupt motion change, that is when the camera is moved by the operator. The targets remain tracked and new tracks of targets in the lower part of the image are initialized. In Fig. 7 we can notice how the central bee is tracked for more than 200 frames. Similarly to the previous case, the camera is moved by the operator and we can spot it by looking at the position of the dark-orange trajectory in Fig. 7a and b.

Results of A-D are quantitatively reported in Table III and qualitative in Fig. 8. Even if the same sequence is already used in [1] and [41], we cannot compare the results since they employ ground-truth information to initialize the target locations and when tracking failures occur. Whereas, we do not use any manual intervention and we let the tracker run throughout the sequence. The results for T4 in A-D are not provided due to its implementation limitations with long sequences [24]. From Table III, HA+GGB has the highest P and R, and lowest IDSR overall. The ID switches of HA+GGB

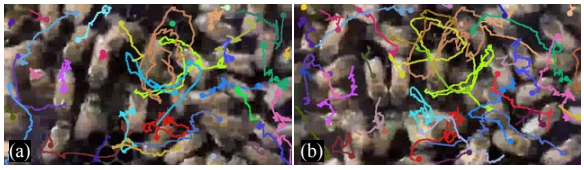

Fig. 7. Sample tracking results on the bee dataset (B-D) with long term visualization. Abrupt motion changes are successfully dealt with the proposed tracker. The trajectories from (a) to (b) are all shifted on the top-right because the camera has been moved by the operator. The trajectories are truncated to the last 200 frames to make the visual representation clearer.

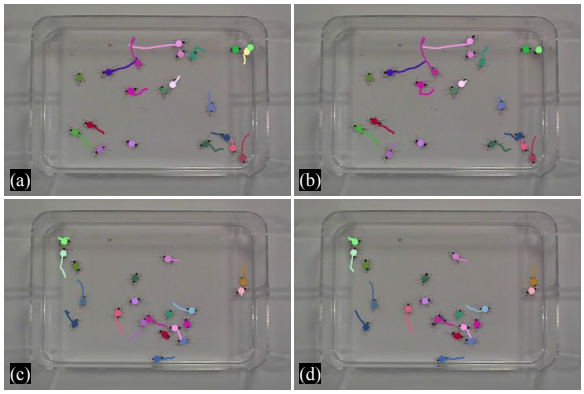

Fig. 8. Sample tracking results on the ant dataset (A-D). ID switches can be due to (a,b) multiple detections on a single target (top-right corner), or (c,d) crossing/overlapping targets (middle). The trajectories are truncated to the last 200 frames to make the visual representation clearer.

are due to two reasons: first, multiple detections are generated on the same targets when they are close to borders (Fig. 8a,b) and it is due to the reflection of the ant in the glass; second, when targets cross each other (Fig. 8c,d) there can be track interruptions, which is mainly due to the fact that occlusions have not been explicitly modeled in GGB. Likewise on B-D, T2 has closer performance to HA+GGB, but the number of ID switches is still about double. Similarly to B-D, the performance of T3 is closer to that of T2 and better than T1. This is due to the different motion of the targets, which can be better approximated with a linear model. Moreover, with the same buffer size as T6 (50 frames), HA+GGB is more robust at keeping the correct identities associated to the targets, even without employing prior dynamics.

Fig. 9 shows that most of the targets are successfully tracked in C-D. However, the position of some trajectories is not accurate due to interpolation in the tracker. Fast variations in direction are smoothed. For additional tracking results see Sec. VII-A2.

Lastly, the median execution time for both T2 and HA is 0.001 sec/frame; and for T1 and T3 is 0.003 and 0.009 sec/frame, respectively. HA+GGB has a higher execution time (0.064 sec/frame) because it allows pruning spurious tracks online during the association process. T5, T6 and T5+GGB have similar median execution time, that is 6.652, 6.848 and 6.701 sec/frame, respectively. T4 has the highest median execution time (73.748 sec/frame).

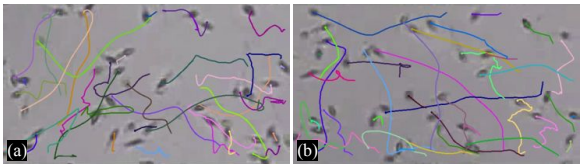

Fig. 9. Sample tracking results on the cell dataset (C-D).

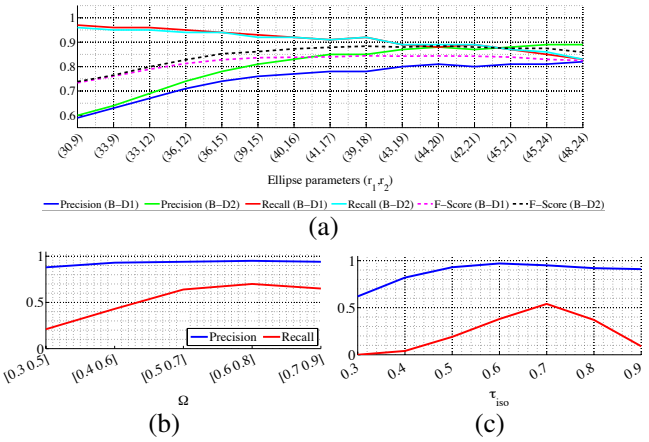

Fig. 10. Sensitivity of the proposed detector (Fusion) by changing (a) the shape parameters (r_a, r_b) on B-D1 and B-D2. Sensitivity of the proposed isocontour-based approach by changing values of (b) the interval Ω and (c) by applying single values of τ_{iso}.

D. Sensitivity analysis

1) Detection: The sensitivity of the detector is assessed by changing the size parameters of the ellipse (r_a, r_b). The experimentation is performed on B-D1 and B-D2 since these are more challenging sequences than A-D. Fig. 10a shows that smaller (r_a, r_b) leads to higher R and lower P. This is due to the multiple detections that are converged on single targets, and since they are not accurately aligned to them, they are not pruned by NMS. While increasing the values of (r_a, r_b), R does not decrease as fast as the increase of P, meaning that the fitting process aligns the shape while enabling an effective pruning with NMS. Small variations of the size, between $r_a = [39\ 44]$ and $r_b = [17\ 21]$ do not affect the performance considerably. Interestingly, R in B-D2 increases faster than that in B-D1 due to the lower density of targets, as the detector is less biased by intensity values of adjacent targets. P follows a similar trend for both cases.

We assess the sensitivity of HIM on B-D1 for different values of Ω and τ_{iso} (Fig. 10b,c). P is the highest for τ_{iso}>0.5, whereas R remains at low levels (below 0.60) throughout all the variations of τ_{iso}. R is greater than zero for τ_{iso}>0.4 and τ_{iso}<0.9, and it reaches the highest value (0.54) for τ_{iso}=0.7. This is the reason we employed a multilayer-isocontour approach with Ω=[0.5 0.8]. Indeed, values outside this range (e.g. [0.3 0.9]) do not further improve the performance. In particular, for τ_{iso}<0.5 HIM would mainly outline clutter and big regions would be discarded by the shape constraint (eccentricity). On the other hand, for τ_{iso}>0.8 isocontours would outline small and negligible regions.

The sensitivity of the likelihood function of GCD is analytically analyzed by changing σ_C of Eq. 7. Fig. 11a is obtained

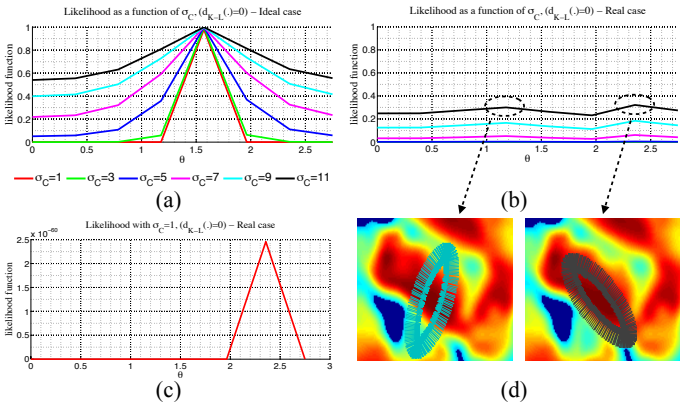

Fig. 11. Behavior of Eq. 7 when varying σ_C with $d_{K-L}(\cdot)$=0 for (a) an ideal case and (b) a real case. (d) Cases of local maxima from (b) on real observations. (c) Case extracted from (b) when σ_C=1.

through a controlled experiment where we generate a 2D Gaussian map distributed as the $\frac{\pi}{2}$-rotated prior $\mathcal{P}(\cdot)$ and we compute its gradient. We then compute the error $E(\cdot)$ between such a gradient and the normal vectors of a θ-rotated ellipse perimeter ($\vec{\mathcal{E}}(\theta)$). The rotation ranges in [0,π), with $\frac{\pi}{8}$-steps. The gradient vectors are normalized, $d_{K-L}(\cdot)$=0 and we vary σ_C in order to observe the effect of the ellipse rotation on an "ideal" observation. Fig. 11a shows that the likelihood function is a concave function where the maximum is located at the $\frac{\pi}{2}$ orientation. Variations of σ_C enhance the localization of maxima. In this case the observation is noiseless, but in a real scenario the observations are often noisy and the likelihood function might have multiple local maxima. An example of real scenario is in Fig. 11b where the cases corresponding to the two maxima are shown in Fig. 11d; the case on the left-hand side provides a local maxima due to the variations of gradient generated by the neighboring targets. Moreover, Fig. 11c shows the trend of the function when σ_C=1, where the peak corresponds to the orientation of the target on right-hand side image of Fig. 11d.

2) Tracking: We analyze the sensitivity of the tracking algorithm on B-D1 by varying the buffer duration Ξ in the interval [10 100] frames with step size 10, the temporal interval permitted to merge short tracks τ_b for values $\{5, 10, 20\}$, and $\{\sigma_1, \sigma_2\}$ used for the computation of the likelihood function (Eq. 12) for values $\{0.15, 5\}, \{0.3, 10\}, \{0.45, 15\}$. The performance are shown in Fig. 12. The variation of the buffer duration does not largely affect tracking performance for fixed values of $\{\sigma_1, \sigma_2\}$ (P and R vary within an interval of 0.02). The robustness to ID switches increases with increasing buffer duration and decreases with a shorter buffer durations. A larger buffer enables the algorithm to process more data in order to generate more accurate trajectories. We use a buffer size of 50 frames since it provides a good trade-off between tracking latency and performance. The smaller τ_b, the higher P and the lower R. In fact, many false (true) detections are not linked and get discarded because they generate short (fragmented) trajectories. Fast moving targets are the main cause of fragmented trajectories. This is observed also when values of $\{\sigma_1, \sigma_2\}$ are small, which is expected since the algorithm does not link short tracks with large spatio-temporal gaps between

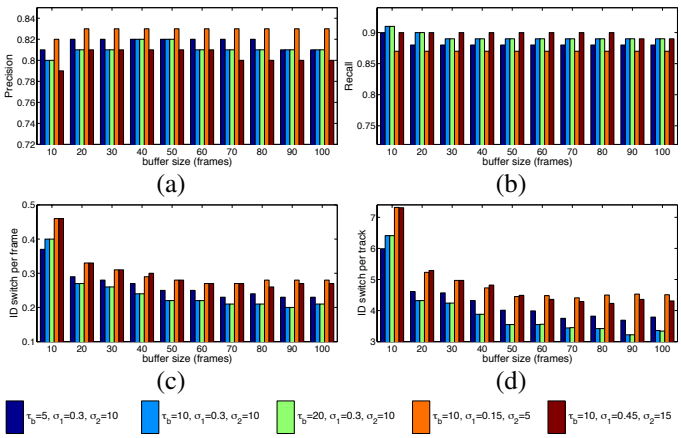

Fig. 12. Tracking results on B-D1 at varying parameters of GGB. (a) Precision, (b) Recall, (c) Γ and (d) Λ are calculated for different buffer durations (horizontal axis), different τ_b values (dark blue, light blue and green bars) and different $\{\sigma_1, \sigma_2\}$ values (light blue, orange and brown bars). Please see Eq. 12 for details about the variables.

an end and a start. Lastly, we assess the robustness to ID switches as a function of $\{\sigma_1, \sigma_2\}$. Fig. 12c,d show the poor performance of the algorithm when $\{\sigma_1, \sigma_2\}$ are either too small or too large. On the one hand, the fragmentation of trajectories leads to continuous track re-initializations when small values of $\{\sigma_1, \sigma_2\}$ are used. On the other hand, larger values of $\{\sigma_1, \sigma_2\}$ provide longer trajectories, but more ID switches between neighboring targets since the kernel for the linkage is larger (a less steep likelihood function).

Finally, we analytically analyze the behavior of Eq. 12 when varying σ_1 and σ_2 (Fig. 13) to test how σ_1 and σ_2 affect the short track linking. The steeper the functions, the more sensitive the tracker in linking short tracks spatio-temporally close to each other. In Fig. 13a we can observe how variations of σ_1 affect the link probability; the horizontal axis indicates the normalized distance, the vertical axis the value of the link probability. Note that the normalized distance takes into account the size (area) of the target (Eq. 13), meaning that, if $\beta_{b',b}>1$, the point we are trying to associate is located outside the target region of the other point. Hence, σ_1 has to be set according to the velocity of the targets in the scene. In Fig. 13b we can observe how the link probability changes while changing the distance between two candidate points to be associated. The value of σ_2 is useful to account for sporadic miss-detections of targets occurring frame-by-frame. Fig. 13b shows also that the lower σ_2, the fewer the frames allowed for miss-detections. From the graph we can also infer that τ_b=10 accounts only for small values of link probability. By setting τ_b=5 (as shown in the analysis of Fig. 12), associations with link probability below 0.5 might be discarded and true associations might be ignored. Hence, τ_b allows associations with small link probability ($\sim$0.1) to be discarded in order to avoid spurious long tracks and to reduce the algorithmic complexity.

E. Application of the tracker on other sequence types

The tracking algorithm (HA+GGB) is further evaluated in multi-person tracking applications. We aim to infer strengths

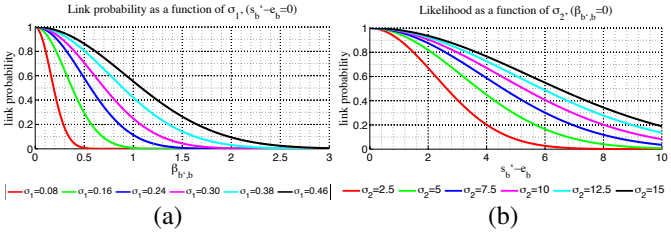

Fig. 13. Behavior of Eq. 12 when varying σ_1 and σ_2.

TABLE IV
COMPARISON AMONG TRACKERS ON MULTI-PERSON TRACKING APPLICATIONS. THE BUFFER IS 25 FRAMES. KEY: P: PRECISION; R: RECALL; IDS: ID SWITCHES.

Tracker	Dataset	MOTA	MOTP	P	R	IDS
CRFBT [11]	ETH-B	**.67**	**.77**	.89	**.76**	**36**
MT-TBD [2]		.56	.75	.81	.75	109
GOG [26]		.55	.75	**.95**	.60	175
HA+GGB		.59	.76	.83	.75	78
CRFBT [11]	ETH-S	.62	.75	.85	.76	**3**
MT-TBD [2]		.61	.73	.82	**.79**	12
GOG [26]		**.65**	**.77**	**.90**	.74	15
HA+GGB		.58	.75	.80	**.79**	21
DCO [42]	PETS-S2L1	.82	.74	-	-	15
DLP [43]		**.91**	.71	-	-	**5**
K-SPO [5]		.80	.58	-	-	28
GAC [44]		.81	.58	-	-	19
HA+GGB		.89	**.88**	.94	.96	44
DCO [42]	TUD	.61	.66	-	-	7
DLP [43]		.79	.74	-	-	**4**
HA+GGB		**.80**	**.85**	.95	.87	27

and weaknesses of HA+GGB with respect to state-of-the-art trackers that employ appearance and velocity information as distinguishing features. We use the following datasets: ETH[3] Bahnhof (ETH-B) and Sunnyday (ETH-S), PETS on the sequence S2L1[4] (PETS-S2L1) and TUD Stadtmitte[5] (TUD). We use detections from [11][6]. We compare HA+GGB with: Conditional Random Field based tracker (CRFBT) [11], multi-target track-before-detect (MT-TBD) [2], globally-optimal greedy algorithm for tracking (GOG) [26], multi-target tracking by continuous energy minimization (DCO) [42], discriminative label propagation based tracker (DLP) [43], K-Shortest path optimization (K-SPO) [5] and multi-target tracker under global appearance constraints (GAC) [44]. We use MOTA, MOTP, Precision, Recall and ID switches (IDS) as evaluation metrics [2], [10] for comparison. Note that HA+GGB, unlike CRFBT, GOG, DCO, DLP, K-SPO and GAC that work offline, performs buffered tracking. Unlike MT-TBD, whose delay is 100 frames, HA+GGB has a 25-frame delay.

Quantitative results are shown in Tab. IV. HA+GGB overall has comparable results in terms of MOTA, MOTP with respect to state-of-the-art trackers, even without using appearance and velocity information. However, due to this, the number of IDS is larger. Below we show how appearance and velocity models are key for the discrimination of targets in people tracking.

[3]http://www.vision.ee.ethz.ch/~aess/dataset/. Accessed: May 2014.
[4]http://www.cvg.rdg.ac.uk/PETS2009/a.html. Accessed: May 2014.
[5]http://www.d2.mpi-inf.mpg.de/node/428/. Accessed: May 2014.
[6]http://iris.usc.edu/people/yangbo/downloads.html. Accessed: May 2014.

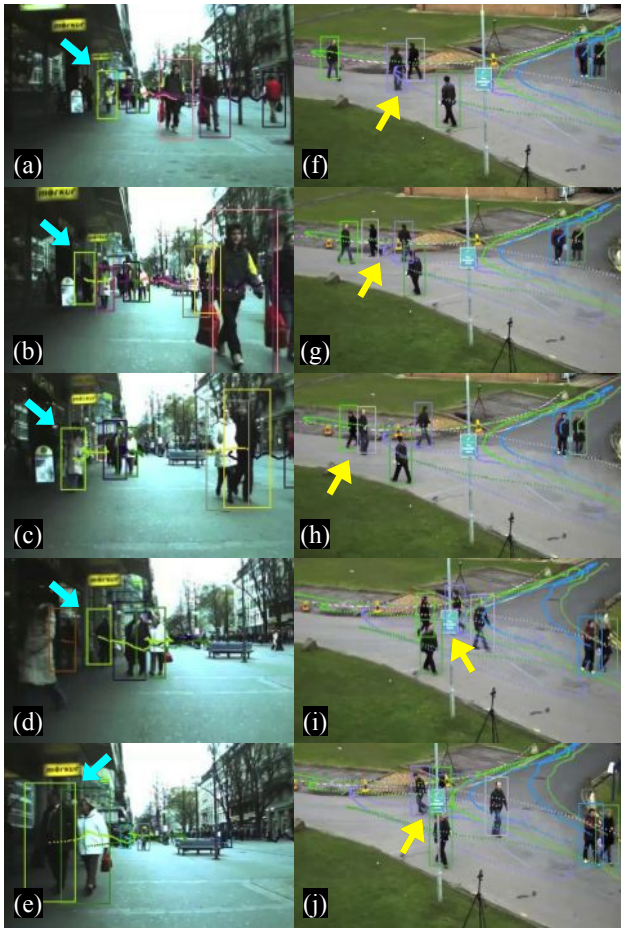

Fig. 14. Sample tracking results of the proposed tracking method on (a-e) ETH Bahnhof and (f-j) PETS-S2L1 datasets.

In ETH-B, MOTA and IDS of HA+GGB are the second best, Recall is higher than GOG which means that HA+GGB correctly tracks a larger number of targets. In ETH-S, HA+GGB has the lowest MOTA value even if it has the best Recall with MT-TBD and comparable Precision values with the other trackers. The low MOTA value is due to the large number of IDS. Fig. 14a-e show cases of identity switches in ETH-B, where the cyan arrow is indicating subsequent IDS due to occlusions. These errors occur due to the absence of the appearance model in HA+GGB. This model can be included in the likelihood function of Eq. 12. However, the use of an appearance model is out of the scope of this paper since it would create ambiguities when objects with the same appearance are tracked. In PETS-S2L1, MOTA of HA+GGB outperforms all the trackers except DLP where its MOTA is 0.02% higher. MOTP of HA+GGB is the highest overall. IDS occur when there are full overlaps between people. In Fig. 14f,g, the yellow arrow indicates situations of successful and unsuccessful associations when occlusions occur. We can observe that the association is reliable in the case of partial occlusions, whereas it fails when full occlusions occur. Unlike in ETH-B, here the target discrimination can be addressed by including the velocity information (i.e. orientation of motion) in the likelihood function (Eq. 12). We did not include the velocity information in HA+GGB since it would have created ambiguities when bees are tracked due to their unpredictable motion variations. Lastly, in TUD, MOTA and MOTP are the best among all the compared approaches.

In summary, HA+GGB demonstrated to be effective in discarding false positive detections and being precise in generating trajectories. HA+GGB has achieved average precision, and in some cases higher performance, than state-of-the-art trackers that work offline. On the other hand, since we are not using discriminative features (e.g. color and velocity) the distinguishability of targets is lower. We showed that additional affinities can be directly included in the likelihood function in the case of different applications, such as multi-person tracking.

VIII. Conclusions

We presented a framework for detection and tracking in image sequences with a high density of homogeneous targets. The detection stage relies on gradient information and intensity levels of target-intensity maps to extract candidate target locations. The features are processed using the combination of a method based on Markov Chain Monte Carlo and one based on hierarchical isocontours. Moreover, we use a greedy tracking algorithm that recursively associates detections within a short temporal buffer.

Future work may involve the extension of the first stage of the pipeline by adding a complementary shape-fitting detector or by employing a method for the automatic selection of shape parameters (ellipse) in the case of targets with different sizes [45]. A multi-scale approach [6] could be considered to deal with objects with varying size. Persistent false positive detections could be discarded either at the tracking stage or at post-detection via classification, for example by learning color or texture patterns of the objects of interest.

References

[1] Z. Khan, T. Balch, and F. Dellaert, "MCMC data association and sparse factorization updating for real time multitarget tracking with merged and multiple measurements," *IEEE Trans. on PAMI*, vol. 28, no. 12, pp. 1960–1972, Dec. 2006.

[2] F. Poiesi, R. Mazzon, and A. Cavallaro, "Multi-target tracking on confidence maps: an application to people tracking," *Comp. Vis. Ima. Und.*, vol. 117, no. 10, pp. 1257–1272, 2013.

[3] T. Kimura *et al.*, "Development of a new method to track multiple honey bees with complex behaviors on a flat laboratory arena," *Plos One*, vol. 9, no. 1, pp. 1–12, Jan. 2014.

[4] S. Stalder, H. Grabner, and L. V. Gool, "Cascaded confidence filtering for improved tracking-by-detection," in *Proc. of ECCV*, Crete, Greece, Sep. 2010, pp. 369–382.

[5] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua, "Multiple object tracking using K-Shortest paths optimization," *IEEE Trans. on PAMI*, vol. 33, no. 9, pp. 1806–1819, Sep. 2011.

[6] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. of CVPR*, San Diego, CA, USA, Jan. 2005.

[7] D. Delannay, N. Danhier, and C. D. Vleeschouwer, "Detection and recognition of sports(wo)men from multiple views," in *Proc. of ICDSC*, Como, Italy, Sept. 2009, pp. 1–7.

[8] P. F. Felzenszwalb, R. Girshick, and D. Ramanan, "Object detection with discriminatively trained part based models," *IEEE Trans. on PAMI*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.

[9] S. Razatofighi, R. Hartley, and W. Hughes, "A new approach for spot detection in total internal reflection fluorescence microscopy," in *Proc. of ISBI*, San Francisco, CA, USA, May 2012, pp. 860–863.

[10] B. Benfold and I. Reid, "Stable multi-target tracking in real-time surveillance video," in *Proc. of CVPR*, Colorado Springs, USA, Jun. 2011, pp. 3457–3464.

[11] B. Yang and R. Nevatia, "An online learned CRF model for multi-target tracking," in *Proc. of CVPR*, Providence, RI, USA, Jun. 2012, pp. 2034–2041.
[12] C. Huang, Y. Li, and R. Nevatia, "Multiple target tracking by learning based hierarchical association of detection responses," *IEEE Trans. on PAMI*, vol. 35, no. 4, pp. 898–910, Apr. 2013.
[13] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman filter: particle filters for tracking applications*. Artech House, 2004.
[14] L. Kratz and K. Nishino, "Tracking pedestrians using local spatio-temporal motion patterns in extremely crowded scenes," *IEEE Trans. on PAMI*, vol. 34, no. 5, pp. 987–1002, May 2012.
[15] L. Yang, Z. Qiu, A. Greenaway, and W. Lu, "A new framework for particle detection in low-SNR fluorescence live-cell images and its application for improved particle tracking," *IEEE Trans. on Bio. Eng.*, vol. 59, no. 7, pp. 2040–2050, Jul. 2012.
[16] K. Shafique and M. Shah, "A noniterative greedy algorithm for multi-frame point correspondence," *IEEE Trans. on PAMI*, vol. 27, no. 1, pp. 51–65, Jan. 2005.
[17] E. Bertin and S. Arnouts, "SExtractor: software for source extraction," *Astron. and Astrop. Supp.*, vol. 117, pp. 393–404, May. 1996.
[18] D. Comaniciu and P. Meer, "Distribution free decomposition of multi-variate data," *IEEE Trans. on PAMI*, vol. 2, pp. 22–30, 1999.
[19] Y. Kimori, N. Baba, and N. Morone, "Extended morphological processing: a practical method for automatic spot detection of biological markers from microscopic images," *BMC Bioinf.*, vol. 11, no. 373, pp. 1–13, Jul. 2010.
[20] A. Genovesio *et al.*, "Multiple particle tracking in 3-D+ microscopy: Method and application to the tracking of endocytosed quantum dots," *IEEE Trans. on Ima. Proc.*, vol. 15, no. 5, pp. 1062–1070, May 2006.
[21] D. Herman, "Multi-object tracking algorithm for biological motion using kalman filter and munkres algorithm," http://studentdavestutorials.weebly.com/multi-bugobject-tracking.html, last accessed: May 2014.
[22] H. Kuhn, "The Hungarian method for the assignement problem," *Naval Research Logistics Quarterly*, vol. 2, pp. 843–854, 1955.
[23] A. Frangi, W. Niessen, K. Vincken, and M. Viergever, "Multiscale vessel enhancement filtering," in *Proc. of MICCAI*, Cambridge, MA, USA, Oct. 1998, pp. 130–137.
[24] A. Andriyenko, K. Schindler, and S. Roth, "Discrete-continuous optimization for multi-target tracking," in *Proc. of CVPR*, Providence, RI, Jun. 2012, pp. 1926–1933.
[25] A. Goldberg, "An efficient implementation of a scaling minimum-cost flow algorithms," *Journal of Algo.*, vol. 22, no. 1, pp. 1–29, Jan. 1997.
[26] H. Pirsiavash, D. Ramanan, and C. Fowlkes, "Globally-optimal greedy algorithms for tracking a variable number of objects," in *Proc. of CVPR*, Colorado Springs, USA, Jun. 2011, pp. 1201–1208.
[27] Z. Wu, A. Thangali, S. Sclaroff, and M. Betke, "Coupling detection and data association for multiple object tracking," in *Proc. of CVPR*, Providence, RI, USA, Jun. 2012, pp. 1948–1955.
[28] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Pearson Prentice Hall, 2008.
[29] A. Papoulis and S. Pillai, *Probability, random variables and stochastic processes*. Mc Graw Hill, 2002.
[30] P. Dollár, "Piotr's Image and Video Matlab Toolbox (PMT)," http://vision.ucsd.edu/~pdollar/toolbox/doc/index.html.
[31] D.-J. Kroon, "Segmentation of the mandibular canal in cone-beam CT data," Ph.D. dissertation, University of Twente, 2006.
[32] P. Soille, *Morphological Image Analysis: Principles and Applications*. Springer-Verlag, 1999, pp. 173-174.
[33] Y. Mingqiang, K. Kidiyo, and R. Joseph, "A survey of shape feature extraction techniques," *Patt. Recogn.*, pp. 43–90, Jul. Peng-Yeng Yin (Ed.), 2008.
[34] P. Berens, "Circstat: A matlab toolbox for circular statistics," *Journal of Statistical Software*, vol. 31, no. 10, pp. 1–21, Sept. 2009.
[35] J. Wolf, A. Viterbi, and G. Dixon, "Finding the best set of K paths through a trellis with application to multitarget tracking," *IEEE Trans. Aero. Elec. Sys.*, vol. 25, no. 2, pp. 287–296, Mar. 1989.
[36] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide baseline stereo from maximally stable extremal regions," in *BMVC*, Cardiff, UK, Sep. 2002, pp. 384–396.
[37] https://www.mathworks.com/matlabcentral/fileexchange/34040-simple-tracker. Last accessed: November 2013.
[38] J. Crocker and D. Grier, "Methods of digital video microscopy for colloidal studies," *Journal of Col. and Int. Sci.*, vol. 179, no. 1, pp. 298–310, Apr. 1996.
[39] *Accuracy (trueness and precision) of measurement methods and results - Part 1: General principles and definitions*, International Organization for Standardization Std. ISO 5725-1, Dec. 1994.
[40] S. Oh, J. Rehg, T. Balch, and F. Dellaert, "Learning and inferring motion patterns using parametric segmental switching linear dynamic systems," *Int. Jour. of Comp. Vis.*, vol. 77, no. 1-3, pp. 103–124, May 2008.
[41] N. von Hoyningen-Huene and M. Beetz, "Robust real-time multiple target tracking," in *Proc. of ACCV*, Xi'an, CH, Sep. 2009, pp. 247–256.
[42] A. Andriyenko and K. Schindler, "Multi-target tracking by continuous energy minimization," in *Proc. of CVPR*, Providence, RI, USA, Jun. 2011, pp. 1265–1272.
[43] A. Kumar and C. D. Vleeschouwer, "Discriminative label propagation for multi-object tracking with sporadic appearance features," in *Proc. of ICCV*, Sydney, Australia, Dec. 2013, pp. 2000–2007.
[44] H. Shitrit, J. Berclaz, F. Fleuret, and P. Fua, "Tracking multiple people under global appearance constraints," in *Proc. of ICCV*, Barcelona, ES, Nov. 2011, pp. 137–144.
[45] T. Kobayashi, "BoF meets HOG: Feature extraction based on histograms of oriented p.d.f gradients for image classification," in *Proc. of CVPR*, Portland, OR, USA, Jun. 2013, pp. 747–754.

Fabio Poiesi Fabio Poiesi is a Postdoctoral Research Assistant at Queen Mary University of London (UK) working with Prof. Andrea Cavallaro under the European project ARTEMIS COPCAMS (COgnitive & Perceptive CAMeraS - copcams.eu). He received his Ph.D. in Electronic Engineering and Computer Science from the Queen Mary University of London in 2014, and BSc and MSc degrees in Telecommunication Engineering from the University of Brescia (Italy) in 2007 and 2010, respectively. His research interests are video multi-target tracking in highly-populated scenes, performance evaluation of tracking algorithms and behaviour understanding for the analysis of human interactions in crowds.

Andrea Cavallaro Andrea Cavallaro is Professor of Multimedia Signal Processing and Director of the Centre for Intelligent Sensing at Queen Mary University of London, UK. He received his Ph.D. in Electrical Engineering from the Swiss Federal Institute of Technology (EPFL), Lausanne, in 2002. He was a Research Fellow with British Telecommunications (BT) in 2004/2005 and was awarded the Royal Academy of Engineering teaching Prize in 2007; three student paper awards on target tracking and perceptually sensitive coding at IEEE ICASSP in 2005, 2007 and 2009; and the best paper award at IEEE AVSS 2009. Prof. Cavallaro is Area Editor for the IEEE Signal Processing Magazine and Associate Editor for the IEEE Transactions on Image Processing. He is an elected member of the IEEE Signal Processing Society, Image, Video, and Multidimensional Signal Processing Technical Committee, and chair of its Awards committee. He served as an elected member of the IEEE Signal Processing Society, Multimedia Signal Processing Technical Committee, as Associate Editor for the IEEE Transactions on Multimedia and the IEEE Transactions on Signal Processing, and as Guest Editor for seven international journals. He was General Chair for IEEE/ACM ICDSC 2009, BMVC 2009, M2SFA2 2008, SSPE 2007, and IEEE AVSS 2007. Prof. Cavallaro was Technical Program chair of IEEE AVSS 2011, the European Signal Processing Conference (EUSIPCO 2008) and of WIAMIS 2010. He has published more than 130 journal and conference papers, one monograph on Video tracking (2011,Wiley) and three edited books: Multi-camera networks (2009, Elsevier); Analysis, retrieval and delivery of multimedia content (2012, Springer); and Intelligent multimedia surveillance (2013, Springer).