Kamkar, S., Moghaddam, H. A., Lashgari, R., & Erlhagen, W. (2022).
Brain-inspired multiple-target tracking using Dynamic Neural Fields.
Neural Networks, 151, 121-131.
DOI: 10.1016/j.neunet.2022.03.026

Brain-inspired Multiple-target Tracking Using Dynamic Neural Fields

Shiva Kamkar[1*], Hamid Abrishami Moghaddam[1,2], Reza Lashgari[3] and Wolfram Erlhagen[4]

[1]Machine Vision and Medical Image Processing (MVMIP) Laboratory, Faculty of Electrical and Computer Engineering, K.N. Toosi University of Technology, Tehran, Iran.
[2]Center for International Scientific Studies and Collaboration (CISSC), Tehran, Iran.
[3]Institute of Medical Science and Technology, Shahid Beheshti University, Tehran, Iran.
[4]Research Centre of Mathematics, University of Minho, Guimarães, Portugal.

*Corresponding author(s). E-mail(s): sh.kamkar@email.kntu.ac.ir;
Contributing authors: moghaddam@kntu.ac.ir;
r_lashgari@sbu.ac.ir; wolfram.erlhagen@math.uminho.pt;

Abstract

Despite considerable progress in the field of automatic multi-target tracking, several problems such as data association remained challenging. On the other hand, cognitive studies have reported that humans can robustly track several objects simultaneously. Such circumstances happen regularly in daily life, and humans have evolved to handle the associated problems. Accordingly, using brain-inspired processing principles may contribute to significantly increase the performance of automatic systems able to follow the trajectories of multiple objects. In this paper, we propose a multiple-object tracking algorithm based on dynamic neural field theory which has been proven to provide neuro-plausible processing mechanisms for cognitive functions of the brain. We define several input neural fields responsible for representing previous location and orientation information as well as instantaneous linear and angular speed of the objects in successive video frames. Image processing techniques are applied to extract the critical object features

including target location and orientation. Two prediction fields anticipate the objects' locations and orientations in the upcoming frame after receiving excitatory and inhibitory inputs from the input fields in a feed-forward architecture. This information is used in the data association and labeling process. We tested the proposed algorithm on a zebrafish larvae segmentation and tracking dataset and an ant-tracking dataset containing non-rigid objects with spiky movements and frequently occurring occlusions. The results showed a significant improvement in tracking metrics compared to state-of-the-art algorithms.

Keywords: multiple-object tracking, dynamic field theory, brain-inspired algorithms

1 Introduction

Multiple-target tracking (MTT) is a key part of many applications in different scientific areas. In human surveillance systems, it is necessary to know the exact location of humans in every moment [1–7]. Traffic monitoring systems should continuously determine the vehicles' loci [8–10]. In the analysis of biological systems, the exact location of animals and organisms is required on a fine temporal scale [11–16]. Therefore, developing accurate MTT algorithms is of great interest in many application areas. However, despite the significant progress achieved in the last decade [17, 18], the research for robust and efficient online MTT methods remains a valid goal due to various challenges. Some challenges depend on environmental conditions. For example, background artifacts such as rainfall or snowfall in traffic videos or water impurity and bubbles in microscopic videos should be excluded. Other challenges are related to the objects. For example, the objects in many real-world scenarios are often non-rigid. This makes it hard to correctly extract the fine details and all parts of the objects. Particularly relevant for biological organisms, their movements might be very spiky and therefore difficult to follow. Still other challenges affect the data association process which may reduce the accuracy of the algorithm because of wrong labeling. For example, the objects may go under partial or complete occlusion from time to time.

On the other hand, cognitive MTT studies have revealed that human show good performance in attending and tracking multiple objects simultaneously [19, 20]. This ability emerges very early in childhood [21] suggesting that the basic brain mechanisms supporting its development are already in place at an early age. This is perhaps not surprising given the evolutionary pressure in environments with multiple predators and preys. Nowadays, we drive a car on a highway keeping track of multiple vehicles around us to avoid collision or take care of groups of children on a playground by monitoring their movements. It can be thus expected that a realization of the cognitive tracking capacity in a brain-inspired manner will improve the performance compared to other non-brain inspired algorithms.

In this paper, we present an approach to MTT which combines image processing techniques for object segmentation and feature extraction with neuro-inspired processing principles for the real-time prediction of the objects' locations (and orientations). The core part is based on the theoretical framework of Dynamic Neural Fields (DNF, [22]) which describes the activation dynamics of neuronal populations selective to continuous input dimensions such as object location or orientation. The DNF model of MTT consists of several input fields encoding the information about the location, orientation and instantaneous linear and angular speed of all moving objects in two successive frames. In addition, two prediction fields integrate excitatory and inhibitory inputs from the connected feature fields. The readout of the peak positions of the evolving activity patterns in the two fields is taken as prediction of the objects' locations and orientations in the next frame. This information is then used in the data association and labeling process, and the field activities continue to evolve with the updated information as inputs.

The rest of the paper is organized as follows: The relevant literature is reviewed in Section 2. Section 3 introduces basic concepts of DNF. The details of the proposed algorithm are explained in Section 4. Section 5 discusses practical aspects of the DNF-based algorithm and compares its performance with state-of-the-art MTT methods. Finally, a critical discussion of results and conclusions are presented in sections 6 and 7.

2 Literature review

Multiple target tracking involves finding the exact location of objects in successive frames. There is a close relationship between MTT and object detection, object-background discrimination, and segmentation. The idea behind most MTT algorithms consists of incorporating the following steps: First, detecting the location of to-be-tracked objects using either an object detection or an object-background discrimination method. Second, using objects' previous motion information such as velocity to predict its location in the next frame. Third, looking for the targets in the previously predicted area when the new frame arrives and label the matches. Forth, updating both objects' motion and appearance information for use in future anticipations. Based on what strategy is applied in each step, different MTT algorithms have been proposed. Due to numerous applications of MTT in research and industry, the literature is rich in this area. Also, there are multiple reviews on MTT algorithms with focus on deep learning [23] or on particle filters [24]. For a thorough review of different aspects of MTT, we refer to [18].

MTT is also a field of study in the area of cognitive neuroscience. A branch of MTT algorithms is inspired by specific brain mechanisms when a human tracks multiple targets simultaneously. For example, experimental studies have shown that attention and memory are two cognitive processes that play fundamental roles during MTT. There are multiple studies on how humans divide attention and use foveal and peripheral vision to handle MTT challenges such

as occlusion and crowding [19, 25, 26]. Accordingly, some brain-inspired MTT algorithms benefit from attention or memory modules. Also, there are several MTT methods based on neural networks reflecting aspects of the neural architecture of the brain. For a review on recent findings in the cognitive sciences of MTT and how these findings help improving MTT algorithms we refer to [17].
Dynamic field theory is a theoretical framework to model brain function. It has been widely employed in the past to model cognitive functions including visual attention, single object tracking and motion extrapolation, working memory or the learning of object pose and identity [27–31].
The theory is also used in cognitive robotics for decision making, action understanding and observational task learning [32–34]. Several hardware implementation studies on this theory have used neuromorphic approaches [35] and FPGA [36, 37].
In the area of MTT, a DNF-based model has been proposed based on our understanding of the visuospatial cognitive system [38, 39]. This model has a three-layer structure mimicking perception, working memory, and a shared layer with inhibitory interneurons. It is applied to a classical MTT task containing multiple solid circles moving randomly on a plain background. Several circles are designated as targets and others as distractors. Subjects are instructed to follow the targets simultaneously and remember their location at each moment. A DNF-based method has been also applied for smooth pursuit tracking of several solid objects with distinct colors in the workspace of a cooperative robotics assistant [40].
These DNF-based approaches to MTT in relatively controlled environments implement a reactive processing mechanism. It ensures that the neural position representation of a target is able to continuously follow the sensory input up to a certain speed limit. However, limitations of this reactive tracking have not been tested with displays in which targets come close to each other, leading to direct spatial interactions of their neural representations [41]. We benefitted from the movements of biological organisms such as zebrafish larvae and ants in real time and under realistic experimental conditions. They are non-rigid objects with continuously changing appearance and spiky movements. Here, we show that the proposed DNF model of MTT implementing predictive processing mechanisms based on past trajectory information is able to cope with the often irregular locomotive characteristics of zebrafish larvae and the occurrence of numerous occlusions when the larvae or ants approach each other. Before explaining the details of the algorithm, a brief introduction to DNF theory is given in the next section.

3 Introduction to Dynamic Neural Fields

Dynamic field theory is a brain-inspired modeling language that has been extensively used in the past to explain experimental findings in perception, action, and cognition (for an overview see [22]). The theoretical concepts are based on the theory of nonlinear dynamical systems, emphasizing attractor

states and their bifurcations. The theory explains the emergence of stable representations of continuous-valued information, such as for example the direction of heading during navigation or the position of an object in space, by assuming a distance-dependent neuronal connectivity pattern in feature space. Typically, neurons tuned to similar values of a continuous variable excite, and those tuned to dissimilar values inhibit each other. When the neurons are ordered along a line by their selectivity, the activity pattern, which evolves continuously in time in response to transient inputs, is visualized as a spatially localized activity bump. Since bumps are only neutrally stable, their position can be shifted along the continuous field by a weak external input that overlaps partly with the bump position [42]. This property can be exploited to explain the capacity of the CNS to track the position of a moving object in real time [38, 43].

In the simplest case, a field representing the 2D position (x, y) of an object evolves independently at each site governed by the following equation:

$$\tau \dot{u}(x, y, t) = -u(x, y, t) + h + s(x, y, t) + \varepsilon\xi(t) \tag{1}$$

where $u(x, y, t)$ denotes the activity at time t of a neuron representing the coordinates (x, y). The rate of change, $\dot{u}(x, y, t) = \frac{du(x,y,t)}{dt}$, is a smooth function of the current activation, the localized external input $s(x, y, t)$ and a homogeneous inhibitory input $h < 0$. The parameter τ defines the time scale of the evolution of activation. In the absence of external input, $s(x, y) = 0$, the linear dynamical system has a stable solution at the resting level, $h < 0$, whereas for $s(x, y) > 0$ the attractor shifts to the larger level of activation $u(x, y) = h + s(x, y)$. Both attractor states appear to be perturbed by additive white noise, $\xi(t)$, which is assumed to be weak, $\varepsilon \ll 1$.

The additional term, $cg(v(\tilde{x}, \tilde{y}, t))$, in the following equation

$$\tau \dot{u}(x, y, t) = -u(x, y, t) + h + s(x, y, t) + cg(v(\tilde{x}, \tilde{y}, t)) + \varepsilon\xi(t) \tag{2}$$

represents the integration of the activation at position $(\tilde{x}, \tilde{y})$ from a connected field v with the strength parameter $c > 0$. The classical choice of the sigmoidal nonlinearity

$$g(v) = \frac{1}{1 + \exp\left(-\beta(v - v_0)\right)} \tag{3}$$

with steepness parameter $\beta > 0$ and threshold $v_0 = 0$ ensures that only sufficiently positive levels of activation, $v(\tilde{x}, \tilde{y}) > 0$, have a significant impact on the evolution of activation in the u field.

4 Multiple-target tracking using dynamic neural fields

The proposed MTT method benefits from both computer vision techniques and dynamic field theory. We assume that the input video is recorded using a fixed

camera, and the background remains unchanged. Accordingly, its flowchart is given in Figure 1.

4.1 Background subtraction

Each frame is passed through a background subtraction module to extract the foreground, which consists of the objects. The background of the input video has been estimated previously in an off-line manner. It is created by assigning the most frequent value of each pixel among the first 50 frames of the video. The result of background subtraction is the foreground image which represents the input to the next module.
Since objects in our datasets are in general darker than the background, it is worth-noting that in principle an alternative method to extract foreground based on thresholding could be used. We used this alternative method to process videos from the ant tracking datasets. Since one or more ants may stay stationary for the entire video which challenges most of the background estimation methods. We classify pixels with the intensity lower than 100 as foreground. The intensity of a pixel is processed after converting the RGB image to a gray-level image by eliminating the hue and saturation information while retaining the luminance.

4.2 Blob extraction

Some morphological operations are applied to the binary version of the foreground image. First, we remove all connected components that have fewer than 100 pixels to ignore small noisy parts. Second, closing operation using a disk with 5-pixels radius is applied to remove small holes within the foreground. Finally, we extract connected components and consider each as an object that should be tracked. We assume that no occlusion occurs between the objects in the first frame of the video. Moreover, we consider the number of objects in this frame as the number of to-be tracked objects in the entire video. The centroid of each blob is considered as the location of the corresponding object. In addition to objects' loci, we extract the objects' orientation. For each larva, the orientation is defined as the angle of the line connecting the larva's centroid to its head. To find the larva's head position and reduce noise, we apply the morphological erosion operation to the blob with a disk of 5-pixels radius as structuring element [44]. The centroid of the resulting image is considered as the head. Orientation information is not helpful for the ant-tracking dataset since the angle cannot be determined precisely enough due to the symmetry of the ant's morphology. Therefore, we utilized in the data association module both location and orientation information for the larvae tracking and benefitted from only location information for the ant tracking.

4.3 Dynamic neural field model

For each object i, a 2D Gaussian input, $s_{j,i}(x, y, t)$, to the location fields j is defined with the center at the object's position $(p_{x,i,t}, p_{y,i,t})$(Equation 4)

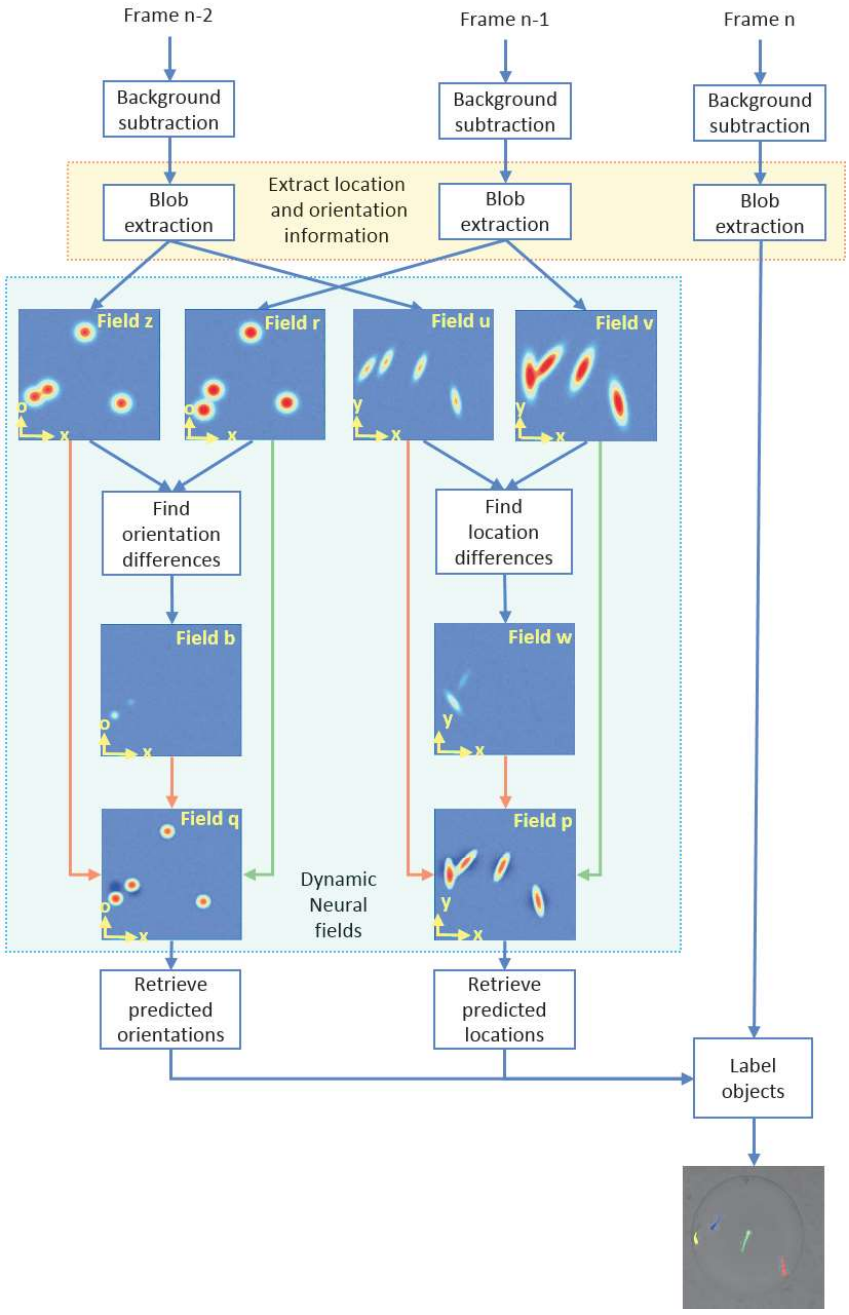

Fig. 1 Flowchart of the proposed MTT method. The green and red arrows represent excitatory and inhibitory inputs, respectively. The blue arrows indicate data flow.

which is found by converting the frame coordinates into the field coordinates. The location information is updated in each frame of the video sequence. The covariance matrix of the Gaussian shows how the object is oriented.

$$s_{j,i}(x,y,t) = a_j \exp\left(-\frac{1}{2}\begin{bmatrix} x - p_{x,i,t} & y - p_{y,i,t}\end{bmatrix} \Sigma_j^{-1} \begin{bmatrix} x - p_{x,i,t} \\ y - p_{y,i,t}\end{bmatrix}\right), i = 1, 2, ..., m \tag{4}$$

where Σ_j^{-1} is the inverse of the covariance matrix and a_j adjusts the strength of the input. Both Σ_j^{-1} and a_j are considered similar for all the Gaussian inputs to field j. We set an initial value $\Sigma_j' = \begin{bmatrix} \sigma_{j,x}'^2 & 0 \\ 0 & \sigma_{j,y}'^2 \end{bmatrix}$ for Σ_j before the start of the algorithm. After processing each frame and extracting the orientation of the object (o), Σ_j' is updated according to Equation 5.

$$\Sigma_j = \begin{bmatrix} \frac{\cos(o)^2}{2\sigma_{j,x}'^2} + \frac{\sin(o)^2}{2\sigma_{j,y}'^2} & -\frac{\sin(2o)}{4\sigma_{j,x}'^2} + \frac{\sin(2o)}{4\sigma_{j,y}'^2} \\ -\frac{\sin(2o)}{4\sigma_{j,x}'^2} + \frac{\sin(2o)}{4\sigma_{j,y}'^2} & \frac{\sin(o)^2}{2\sigma_{j,x}'^2} + \frac{\cos(o)^2}{2\sigma_{j,y}'^2} \end{bmatrix} \tag{5}$$

Since we assume that all objects are already extracted and no occlusion occurs in the first frame, we always know the expected number, m, of objects in the scene. The Gaussian inputs create m localized activation patterns in the input location fields. We distinguish two fields, u and v, representing the locations of all objects in frames n-2 ($s_{u,i}(x,y,t)$) and n-1 ($s_{v,i}(x,y,t)$), respectively. The input driven field dynamics is governed by the following equations:

$$\tau \dot{u}(x,y,t) = -u(x,y,t) + h + \sum_{i=1}^{m} s_{u,i}(x,y,t) + \varepsilon\xi(t) \tag{6}$$

$$\tau \dot{v}(x,y,t) = -v(x,y,t) + h + \sum_{i=1}^{m} s_{v,i}(x,y,t) + \varepsilon\xi(t) \tag{7}$$

In Figure 1, the activation level at each field site is coded in a heat map. The activation is higher for hotter parts and the red circles represent suprathreshold activity.

An additional input field, w, is defined which represents the instantaneous speed of all objects in frame n-1. The speed is given as the Euclidean distance of the object location in frames n-1 and n-2. The activation level in this field is higher for faster moving objects and lower for slower ones.

$$\tau \dot{w}(x,y,t) = -w(x,y,t) + h + \sum_{i=1}^{m} s_{w,i}(x,y,t) + \varepsilon\xi(t) \tag{8}$$

where $s_{w,i}(x,y,t)$ is a Gaussian function similar to $s_{j,i}(x,y,t)$, but with time-dependent amplitude $a_{s_{w,i}}(t)$.

$$s_{w,i}(x,y,t) = a_{s_{w,i}}(t)\exp\left(-\frac{1}{2}\begin{bmatrix} x-p_{x,i,t} & y-p_{y,i,t}\end{bmatrix}\Sigma_{w,i}^{-1}\begin{bmatrix} x-p_{x,i,t} \\ y-p_{y,i,t}\end{bmatrix}\right), i = 1,2,...,m \tag{9}$$

$$a_{s_{w,i}}(t) = d_{i,t}^{\alpha\times\max(0,d_{i,t}^{\alpha})} - 1 \tag{10}$$

where $d_{i,t}$ represents the spatial displacement of object i between frames n-1 and n-2. $\alpha > 0$ is a constant.

Finally, field p is responsible for predicting the future location of objects in frame n. It receives an excitatory input from the v field and inhibitory inputs from the u and w fields. A Gaussian kernel is used for the spatial integration of activity from the connected fields. A sufficiently strong excitatory input at the location of the object in frame n-1 triggers the evolution of a suprathreshold activity pattern whereas the inhibitory input of Gaussian shape from w has a suppressive effect on this location. The inhibitory input from v suppresses the location of the object in the preceding frame n-2. The main effect of the two inhibitory inputs to the prediction field p is that the peak position of the evolving bump does not appear centered at the object location in frame n-1 but shifted in motion direction (Figure 2). To achieve this predictive effect, the spatial width of the input (defined by the standard deviation of the corresponding kernel) from the u field should be sufficiently large to overlap with the object location in frame n-1. The parameter α in Equation 10 defines the strength of the inhibition from the velocity field w. For large α values, the inhibition of the object location might be too strong and no suprathreshold activity pattern will evolve. On the other hand, for smaller α values, the spatial shift of the peak position might not be large enough to make an accurate prediction. For the present application, the parameters of the input kernels are set experimentally in order to minimize the prediction error.

Equation 11 governs the evolution of the prediction field p through time.

$$\begin{aligned}\tau\dot{p}(x,y,t) = -p(x,y,t) + h + g(v(x,y,t)) - \\ \iint_{\Omega} k_{up}(x-x',y-y')g(u(x',y',t))\,dx'\,dy' - \\ \iint_{\Omega} k_{wp}(x-x',y-y')g(w(x',y',t))\,dx'\,dy' + \varepsilon\xi(t)\end{aligned} \tag{11}$$

where the domain $\Omega = [0,M]\times[0,N] \subset \mathbb{R}^2$ satisfies periodic boundary conditions. The domain size covers the frame size with M and N representing the height and width of the input frame, respectively. We define a regular 400×400 spatial discretization grid for the numerical approximation. The step size in horizontal and vertical direction is thus given by $dx = \frac{N}{400}$ and $dy = \frac{M}{400}$, respectively. The choice of dx and dy affects the computational cost and efficiency of the algorithm mainly because of the convolution operator.

$g(.)$ is defined as a ReLU function:

$$g(v) = max(0, v), g(w) = max(0, w) \tag{12}$$

Both $k_{up}(x, y)$ and $k_{wp}(x, y)$ are defined as Gaussian kernels as:

$$k_{jp}(x, y) = c_j \exp(-\frac{1}{2}(\frac{x^2}{\sigma_{j,x}^2} + \frac{y^2}{\sigma_{j,y}^2})), j = v, w \tag{13}$$

The variances $\sigma_{j,x}^2$ and $\sigma_{j,y}^2$ show how the kernel varies alongside the x and y axis, respectively, and c_j controls the kernel amplitude. Figure 2 illustrates for a single object how the integration of the excitatory and inhibitory inputs predicts the object's future location when reading out the peak position of the evolving activity pattern (magenta line). When a certain object remains stationary in two successive frames, the field predicts the same location (left panel). Otherwise, when the object's loci in the two successive frames have moved by a spatial distance, D, the peak position appears to be shifted further ahead in movement direction as indicated by the distance R (right panel).
The amplitude of the Gaussian input from the w field grows as the instantaneous speed of the object increases (blue line). Consequently, the summation of this inhibitory signal and the inhibitory signal from the previous location of the object in frame n-2 (red line) shapes the excitatory activity pattern, causing the shift of the peak position.

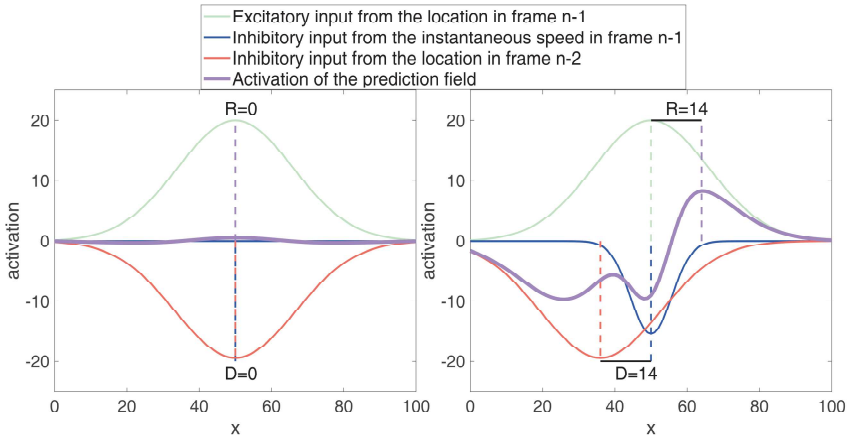

Fig. 2 Prediction using excitatory and inhibitory inputs assuming a single object located in the interval [0 100]. Green, blue, red and magenta lines indicate the contribution of $g(v(x, y, t))$, $\iint_\Omega k_{up}(x - x', y - y')g(u(x', y', t))\, dx'\, dy'$, $\iint_\Omega k_{wp}(x - x', y - y')g(w(x', y', t))\, dx'\, dy'$, and their corresponding summation, respectively. "D" represents object's displacement between frames n-2 and n-1. "R" shows the distance between object's last location (in frame n-1) and its predicted location in frame n. The left panel shows a scenario in which the object has not moved (R=D=0). The right panel represents an object that has displaced about 14 pixels (from $x = 36$ in frame n-2 to $x = 50$ in frame n-1) and the prediction field p anticipates its future location in frame n, 14 pixels ahead (R=14).

To further improve the performance of the DNF-based tracking algorithm, we integrate in addition to the location prediction, the prediction of object

orientation. Two 2D input fields, z and r, are defined representing the x-coordinate of the object's spatial position and its orientation, o. Note that being able to represent in a single activity pattern simultaneously position and orientation information of each object solves the binding problem which would occur when the information is stored in separate fields. Using a 3D-field, which represents both spatial coordinates and the orientation, would be in principle possible but would significantly increase the computational cost.
Field z contains several localized activation patterns, each representing the x-coordinate and the orientation of a specific object in frame n-2. Field r contains the same information in frame n-1. The input fields are governed by the following dynamics:

$$\tau \dot{z}(x,o,t) = -z(x,o,t) + h + \sum_{i=1}^{m} s_{z,i}(x,o,t) + \varepsilon\xi(t) \tag{14}$$

$$\tau \dot{r}(x,o,t) = -r(x,o,t) + h + \sum_{i=1}^{m} s_{r,i}(x,o,t) + \varepsilon\xi(t) \tag{15}$$

where $s_{z,i}(x,o,t)$ and $s_{r,i}(x,o,t)$ are Gaussians given in Equation 4. An additional input field b is defined which represents the object's instantaneous angular speed in frame n-1. It is defined as the difference of the object's orientations in frames n-1 and n-2.

$$\tau \dot{b}(x,o,t) = -b(x,o,t) + h + \sum_{i=1}^{m} s_{b,i}(x,o,t) + \varepsilon\xi(t) \tag{16}$$

where $s_{b,i}(x,o,t)$ is a Gaussian function with variable amplitude similar to Equations 9 and 10. We use an orientation prediction field, q to anticipate the future orientation based on the integration of excitatory and inhibitory inputs from fields r, z and b.

$$\begin{aligned} \tau \dot{q}(x,o,t) = -q(x,o,t) + h + g(r(x,o,t)) - \\ \iint_{\Omega} k_{zq}(x-x',o-o')g(z(x',o',t))\,dx'\,do' - \\ \iint_{\Omega} k_{bq}(x-x',o-o')g(b(x',o',t))\,dx'\,do' + \varepsilon\xi(t) \end{aligned} \tag{17}$$

where, $g(z)$ and $g(r)$ are again ReLU functions (Equation 12) and k_{zq} and k_{rq} are Gaussian kernels defined according to Equation 13.
Similar to the prediction mechanism of Equation 11, the information about the change in orientation in two successive frames is used to extrapolate this change to the next frame. Since all fields share the x-dimension, the position and orientation information can be easily combined in the correspondence mapping of the multi-object tracking.
As can be seen in Figure 1, the bumps in the location and orientation fields have the form of an ellipsoid and a circle, respectively. The reason is that the covariance matrix for the Gaussian inputs to the orientation field is the identity

matrix. However, to improve the visualization in the prediction field p, we use the predicted orientation information to update the covariance matrix of the inputs. Accordingly, in the results section, we only depict the activation patterns in the location field p since they visualize both location and orientation information.

4.4 Retrieving predicted location/orientation

The bumps in the prediction fields, p and q, show the predicted location and orientation of the objects. We extract the peak positions of the activity patterns and convert their coordinates to the frame coordinates. This information is used for labeling the objects in the next module.

4.5 Labeling objects

When the new frame, n, arrives, all blobs are extracted by applying the background subtraction and blob extraction modules. We use two criteria to label these blobs. First, the predicted location should be near to the blob's location. Cognitive studies showed that human subjects use proximity information during MTT [45]. Second, the blob's orientation should be similar to the object's predicted orientation. In fact, we assume that the objects change their location and orientation gradually in successive frames. These assumptions are not too restrictive since we solve a linear assignment problem that considers the proximity-orientation criteria for all the objects simultaneously. The cost function used in this matching process considers the weighted sum of the two criteria. Usually, the weight of location is about 1.5 times higher than the weight of orientation. If an occlusion occurs, we apply the inverse ratio. This strategy considerably decreases the occurrence of mismatch errors and improves data association during occlusion. We detect occlusion events by monitoring the number of blobs in the frame. If it is less than the initial number of objects in the video, at least two blobs corresponding to two objects appear to be merged. Reliable information about the location and orientation of the different objects can thus not be extracted anymore and the prediction fields are not updated for the merged objects. Importantly, the activation patterns representing the information from the last frame before occlusion can still be used for object labeling when the blobs start separating again due to the assumed gradual change of the objects' positions and orientations. During occlusion, we consider the centroid of each object before the start of occlusion. A circle with a predetermined radius and the center on the centroid is defined. Then, the intersection of this circle with the blob is reported as the object.
To visualize the output of the DNF model (see Figure 1), we color each object with a distinct color in the output frame. Finally, the new locations and orientations of the objects are updated in the input fields.

5 Evaluation

We implemented the proposed algorithm in MATLAB R2019a using the COSIVINA software[1] which is designed to facilitate the development of DNF architectures. We used a computer with Windows 10 operating system, Intel Core(TM) i7-9700K processor and 16-GB RAM. The evaluation of the model performance was done on two different datasets. First, we compared the DNF-model with several state-of-the art MTT algorithms on all 10 videos of a zebrafish larvae dataset [15] available online[2]. The videos recorded using a fixed camera show multiple larvae moving naturally in a round or squared container (Figure 3). Tracking these larvae is challenging in different aspects. Larvae are non-rigid objects, and their shape changes continuously. Their movement is also unpredictable and spiky with non-constant acceleration. They occlude each other multiple times, and their similar appearance makes the data association process error-prone. Second, the proposed DNF method is applied on 4 videos (videos 1,3,4 and 5) from an ant-tracking dataset and the performance is compared with a recent MTT method [12]. This dataset is also publicly available[3] and contains ants with similar morphology but different sizes (workers and queen). As fundamentally social animals, close body contact causing severe occlusion is a frequent phenomenon (Figure 3).

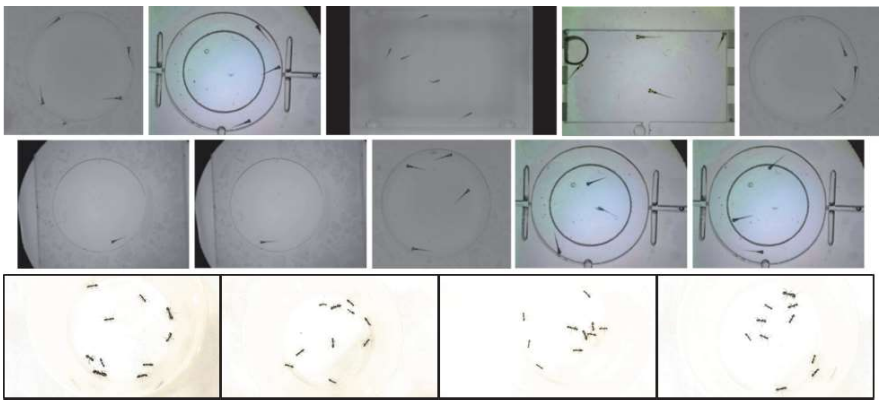

Fig. 3 The first two lines show a sample frame from each video of the zebrafish larvae segmentation and tracking dataset. The third line shows a sample frame from the videos 1, 3, 4 and 5 of the ant-tracking dataset

As performance measures for the comparison, we used MOTA and MOTP as two common metrics to evaluate tracking accuracy and precision . MOTA considers all types of correspondence errors made by the tracking algorithm:

[1] https://github.com/sschneegans/cosivina
[2] https://github.com/Xiao-ying/moving-zebrafish-larvae-segmentation-and-tracking-dataset-
[3] https://data.mendeley.com/datasets/9ws98g4npw/1

$$MOTA = 1 - \frac{\sum_t (FN_t + FP_t + IDS_t)}{\sum_t N_t} \tag{18}$$

where FN_t, FP_t and IDS_t represent in this order the number of false negatives (FN), false positives (FP) and mismatch or identity switches in frame t and N_t indicates the total number of objects present in this frame. MOTP shows how precisely the algorithm determines the position of objects:

$$MOTP = \frac{\sum_{i,t} \mid D_{i,t} - GT_{i,t} \mid}{\sum_t N_t} \tag{19}$$

where $D_{i,t}$ indicates the position of object D as the i-th object in frame t and $GT_{i,t}$ is the ground-truth. $\mid D_{i,t} - GT_{i,t} \mid$ measures the Euclidean distance between the centroids of the blobs of $D_{i,t}$ and $GT_{i,t}$ [15]. A high MOTA score and a low MOTP score show that the algorithm has a high accuracy (i.e., a low number of errors) and a good localization performance.

The results on the larvae segmentation and tracking dataset are given in Table 1 and the output of the DNF-based algorithm for a specific video is given in the supplementary materials. The second, fourth, and fifth column show the evaluation results of the method proposed by [14], [16], and [15], respectively. The third column reports the results when tracking and segmentation are performed by the methods proposed in [14] and [15], respectively. These results are taken from [15]. The sixth column shows the evaluation results of our proposed DNF method. It tracked all objects with the highest MOTA and MOTP compared to the other methods. We can see this improvement both for individual videos and for the average results.Table 2 reports the total number of mismatch errors in all videos when the correspondence between trajectory and object identity switches.

Table 1 Evaluation results for the moving zebrafish larvae segmentation and tracking dataset. According to [15], NaN indicates that no valid data was generated due to a running error while testing. A MOTA score of 1 indicates perfect accuracy and a MOPT score of 0 indicates optimal precision.

	MOTP (pixels) ↓					MOTA ↑				
No.	[14]	[14] and [15]	[16]	[15]	ours	[14]	[14] and [15]	[16]	[15]	ours
1	11.388	NaN	11.662	6.346	**5.394**	1	NaN	0.990	0.988	**1**
2	21.434	12.176	18.395	15.024	**10.981**	0.541	0.922	0.921	0.853	**1**
3	20.648	20.879	18.854	8.113	**6.294**	**1**	0.975	0.985	0.998	0.997
4	16.728	17.665	23.208	10.669	**9.752**	0.0987	0.840	0.987	0.993	**0.998**
5	21.545	21.727	21.890	15.525	**5.658**	-0.01	0.895	0.139	0.994	**1**
6	13.151	12.562	15.020	12.786	**5.671**	-0.27	-0.086	0.914	0.936	**1**
7	25.230	43.746	80.630	30.082	**12.655**	0.005	0.725	0.504	0.954	**0.981**
8	53.096	59.666	98.936	36.901	**8.868**	-0.39	0.739	0.209	0.956	**0.999**
9	29.921	48.739	142.834	15.960	**10.801**	0.673	-0.107	0.327	0.989	**1**
10	219.329	27.532	189.975	25.121	**8.468**	-0.18	0.133	0.906	0.920	**0.999**
avg	43.25	29.41	62.14	17.65	**8.45**	0.33	0.56	0.69	0.96	**0.99**

The evaluation results on the ant tracking dataset are given in Table 3. The output of the DNF-based algorithm for a specific video is given in the supplementary materials. The second and third columns show the results of

Table 2 Total number of obtained mismatches for the zebrafish larvae dataset. NaN indicates that no valid data was generated due to the running error while testing.

No.	[14]	[14] and [15]	[16]	[15]	ours
1	6	NaN	2	**0**	**0**
2	2	3	4	3	**0**
3	**0**	**0**	**0**	**0**	1
4	4	**0**	**0**	**0**	**0**
5	3	2	6	**0**	**0**
6	1	1	1	2	**0**
7	2	1	3	1	**0**
8	11	41	24	14	**1**
9	4	17	3	**0**	**0**
10	4	8	**0**	8	1
avg	3.7	8.1	4.3	2.8	**0.5**

[12] and our proposed DNF algorithm, respectively. In accordance with the approach used by Cao and colleagues (2020), the MOTP definition (Equation 19) is modified by calculating $\frac{D_{i,t} \cap GT_{i,t}}{D_{i,t} \cup GT_{i,t}}$ instead of applying the Euclidian distance between the estimated location and the ground truth. $D_{i,t}$ indicates here the bounding rectangle of object D as the i-th object in frame t and $GT_{i,t}$ is its ground-truth's bounding rectangle. In fact, similar to the processing of the larvae tracking dataset, we segment all of the object's pixels before estimating their positions. However, to allow for direct comparison with the results in [12], we define a bounding rectangle for each object. A MOTP score close to 100% thus indicates a high tracking precision.

A comparison with the performance of the tracker tested by Cao and colleagues reveals that the DNF-based approach is better able to keep consistent track of all objects over time. However, the precision of the DNF model is much worse also compared to the performance in the larvae tracking task. This relative lack in precision can be attributed to the removal of small objects as noise which takes off the ant's thin legs as part of the object boundary. As a consequence, the reported ants' bounding rectangles are smaller than their ground truth. Cao and colleagues achieve higher tracking precision by offline learning of a high-dimensional feature vector characterizing an ant's appearance which becomes associated with a specific trajectory.

Table 3 Evaluation results for the ant tracking dataset. According to the definition in [12], MOTP sores close to 100% represent high precision.

	FP ↓		FN ↓		IDS ↓		MOTA ↑		MOTP (%) ↑	
No.	[12]	ours	[12]	ours	[12]	ours	[12]	ours	[12]	ours
Indoor1	**0**	**0**	**0**	**0**	**0**	**0**	99.4	**100**	**92.1**	48.3
Indoor3	6	**0**	6	**0**	**0**	**0**	99.1	**100**	**89.8**	35.2
Indoor4	8	**0**	8	**0**	4	**0**	98.9	**100**	**91.8**	30.5
Indoor5	1	**0**	**2**	4	**0**	2	**99.8**	98.2	**94**	42.1
avg	3.75	**0**	4	**1**	1	**0.5**	99.3	**99.5**	**91.9**	39.0

The computational cost of the algorithm depends on the number of to-be-tracked objects. For evaluation datasets containing one, three, five and ten

objects, the runtime was about 0.19s, 0.37s, 0.69s and 1.36s per frame on average.

Figure 4 shows an example of an occlusion scenario from video 7 of the fish larvae dataset. The first, second and third columns show the fields u, v and p, respectively, taken at different times of the video sequence. The last column presents the output of the DNF method. Objects in the original videos are all black. We assign a unique color to each for the purpose of illustrating motion trajectories of individual fish larvae. In the first row, the red and yellow larvae are about to occlude. From the prediction field p, we can conclude that all objects are almost stationary except the yellow one. It's neural representation is characterized by stronger surround inhibition (darker blue) and weaker peak activity (lighter red) compared to the stationary objects for which the prediction is not updated. In the second row, occlusion starts and the prediction field maintains the location and orientation information from the preceding frame. When the blobs start to separate, the prediction mechanism guarantees again a consistent trajectory extrapolation for all larvae. The output generated by the DNF model for the entire video sequence is available in the Supplementary material.

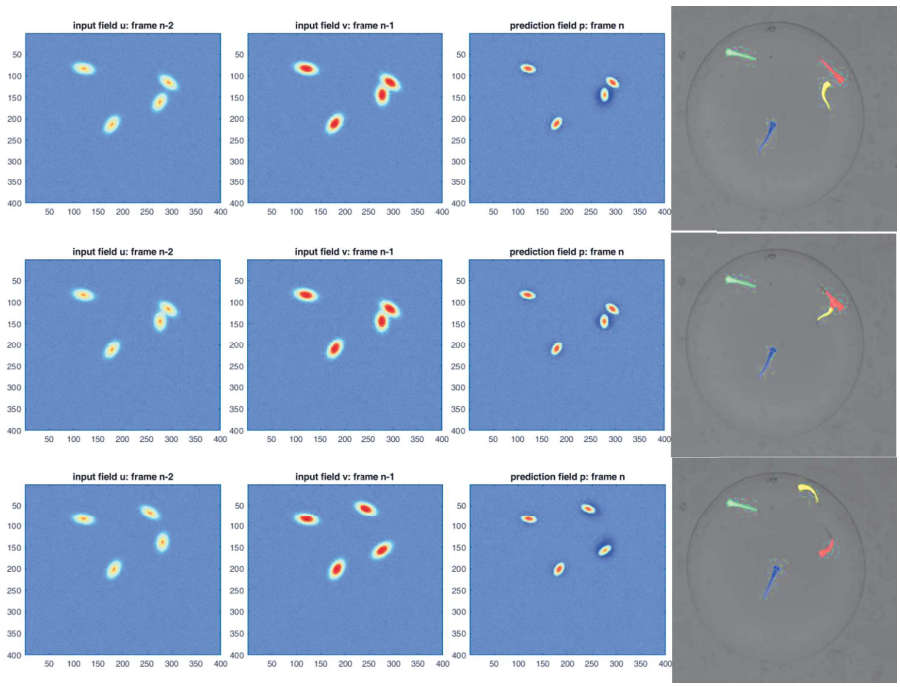

Fig. 4 Sample output of the DNF-based algorithm during an occlusion event. Each row shows the activation of the input fields u (first column), v (second column), the prediction field p (third column) and the labeled output of the algorithm.

6 Discussion

The proposed neuro-dynamics approach to MTT has advantages over existing ones. First, using a prediction field supports simultaneous prediction of future locations for all objects in the scene, instead of calculating a separate prediction for each object [46]. The predictive mechanism relies on a distance measure between corresponding objects in succeeding video frames. As shown in the labeled fish larvae videos, the proposed method works well even when the objects move spiky with pronounced acceleration or show little or no movement.

Second, using the orientation feature in addition to the object's location and assigning adaptable weights to each feature in the labeling process, increase the flexibility of the data association. In particular, when occlusion occurs and two or more objects overlap, location information is difficult to extract, but the predicted orientation of each object can be used to label objects when they get separated.

Third, using morphological operations to discriminate the objects from background, helps to handle the specific challenges of tracking non-rigid objects. Most of the applied deformations can be kept and utilized in the further processing of the relevant object information for a consistent tracking behavior.

Forth, the proposed method doesn't need to be trained by large amounts of data. The spatial ranges of the excitatory and inhibitory inputs to the prediction fields controlling the predictive shift of the peak position are hand-coded in the present application. They could be found automatically in the future using optimization techniques. The specific model parameter values used in the evaluation are summarized in Table 4. Importantly, our model simulations show that variations of these values in a reasonable range do not critically affect the tracking behavior in the challenging applications.

Table 4 Parameters used for DNFs. For the parameters of Σ, the initial covariance matrices are given. These matrices vary according to the instantaneous orientation of the objects throughout the video sequences.

Neural fields		Gaussian inputs	
τ	20	a_u	10
h	-5	a_v	10
ε	1	a_z	15
Lateral interaction kernel		a_r	20
c_w	0.1	Σ'_u	$\begin{bmatrix} 100 & 0 \\ 0 & 10 \end{bmatrix}$
c_v	20	Σ'_v	$\begin{bmatrix} 100 & 0 \\ 0 & 10 \end{bmatrix}$
c_z	1	Σ'_w	$\begin{bmatrix} 40 & 0 \\ 0 & 4 \end{bmatrix}$
$(\sigma_{w,x}, \sigma_{w,y})$	(4,4)	$\Sigma_z = \Sigma_r$	$\begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix}$
$(\sigma_{v,x}, \sigma_{v,y})$	(16,16)	Σ_b	$\begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$
$(\sigma_{z,x}, \sigma_{z,y})$	(4,4)	α	0.3

Although the numerical implementation of a DNF can be seen as a neural network, there are major differences to popular neural network-based machine learning algorithms [46]. Their power resides in their ability to learn rich representations and to extract complex and abstract features from their input. However, they typically require a large batch of training samples that are all available during the training phase. Changing the number of objects in a MTT task would require a retraining of the network parameters. Moreover, online algorithms are often too slow for real-time MTT since they are computationally expensive [46]. Our model-based approach uses pre-defined features that are task relevant and implements processing principles that can be analyzed and understood from a dynamical systems point of view. The neural population activity evolves continuously in time under the influence of various inputs shaping the activation patterns. Since no training process is involved, the DNF model can be applied to video sequences with different numbers of moving objects as long as the number remains fixed during the task. The numerical implementation of the 2D fields could be optimized so that the proposed DNF method shows real-time tracking performance on a normal computer. Strategies such as using a GPU implementation or applying fast Fourier transformation instead of a quadrature formula for the spatial convolution will greatly reduce the computational cost and will further increase the efficiency of the algorithm [47].

The DNF architecture proposed in this paper is different from the model suggested by Spencer and colleagues (2012) in three main aspects. First, in Spencer's model, no prediction of future positions and orientations occurs. A bump follows in a reactive manner the position input when it falls in the range of the local excitatory interaction from the working memory field. Otherwise, the target is missed and the tracking fails. Various sources of experimental evidence have been interpreted as evidence for the existence of neural mechanisms for extrapolating motion trajectories [28, 48] and for prediction during MTT [19, 49]. Predicting the future locations of moving objects from past trajectory information as implemented in our DNF model leads to a nearly perfect tracking behavior in the challenging example video sequences. Second, in Spencer's model, non-targets (distractors) are also considered and tracked. However, in technical real-world applications one is mostly interested in tracking targets and try to use already in the image processing stage discriminant features to distinguish targets and distractors. Third, Spencer's model is applied to behavioral data collected from human subjects while doing a synthetic tracking task introduced by [41]. In this task, the movements of target-distractor pairs follow a specific pattern, in contrast to the highly irregular movements of the zebrafish larvae and ants. A purely reactive tracking based on distance-dependent attractive forces between the input and the neural position representation will have problems to cope with sudden movement adjustments.

Yet our proposed algorithm faces some limitations which we will address in future work. First, we assumed that the environmental conditions such as background and lighting remain unchanged during MTT. Otherwise, an online

background estimation method should be applied. Second, we assumed that the number of objects in the scene is known when the tracking starts and remains constant throughout the video sequence. This allowed us to detect occlusions when the number of individual object blobs changed. Other methods of occlusion detection should be explored to allow the appearances and disappearance of objects in the scene. In addition, using morphological operations, objects are correctly detected and segmented in spite of their deformable shape. However, when two or more objects occlude, a consistent tracking of all objects cannot be always guaranteed. The DNF-based model could benefit from an object detection module to associate an appearance description of an object with a specific trajectory [12].
Finally, this study shows that using brain-inspired methods and in particular DNFs can improve MTT algorithms. Neural fields implement basic processing mechanisms that are consistent with the brain's neural and cognitive functions. Arguably, taking inspiration from the brain may pave the way to new solutions of technical problems since the brain has evolved over a long time under evolutionary pressure. The specific computer vision application is just another example that complements the large variety of applications of the DNF framework in perception, cognition and action.

7 Conclusion

In this study, we proposed a brain-inspired multiple-object tracking method. This algorithm uses both computer vision techniques and dynamic neural fields as a biologically plausible theory. Background subtraction and blob extraction are applied to each frame of the input video. Two prediction fields anticipate the object's future location and orientation according to its loci and orientation in the previous two frames. Data association is done based on proximity and orientation changes. This algorithm outperformed state-of-the-art algorithms on tacking zebrafish larvae and ants. The results showed that the proposed predictive mechanism works well in the data association process even when occlusions occurs.

Acknowledgment

This work has been supported by the Center for International Scientific Studies and Collaboration (CISSC), Ministry of Science Research and Technology, Iran, Grant No. 1483.

Compliance with Ethical Standards:

Funding: This work has been supported by the Center for International Scientific Studies and Collaboration (CISSC), Ministry of Science Research and Technology, Iran, Grant No. 1483. Ethical approval: This article does not contain any studies with human participants performed by any of the authors.

References

[1] Braso G, Leal-Taixe L.: Learning a Neural Solver for Multiple Object Tracking. Available from: https://bit.ly/motsolv.

[2] Karunasekera H, Wang H, Zhang H. Multiple Object Tracking with Attention to Appearance, Structure, Motion and Size. IEEE Access. 2019;7:104423–104434. https://doi.org/10.1109/ACCESS.2019.2932301.

[3] Kumar N, Sukavanam N. A cascaded CNN model for multiple human tracking and re-localization in complex video sequences with large displacement. Multimedia Tools and Applications. 2020 mar;79(9-10):6109–6134. https://doi.org/10.1007/s11042-019-08501-4.

[4] Ma C, Yang F, Li Y, Jia H, Xie X, Gao W. Deep Human-Interaction and Association by Graph-Based Learning for Multiple Object Tracking in the Wild. International Journal of Computer Vision 2021 129:6. 2021 apr;129(6):1993–2010. https://doi.org/10.1007/S11263-021-01460-0.

[5] Peng J, Wang T, Lin W, Wang J, See J, Wen S, et al. TPM: Multiple object tracking with tracklet-plane matching. Pattern Recognition. 2020 nov;107:107480. https://doi.org/10.1016/J.PATCOG.2020.107480.

[6] Sun S, Akhtar N, Song H, Mian A, Shah M. Deep Affinity Network for Multiple Object Tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2021 jan;43(1):104–119. https://doi.org/10.1109/TPAMI.2019.2929520. https://arxiv.org/abs/1810.11780.

[7] Xu R, Guan Y, Huang Y. Multiple human detection and tracking based on head detection for real-time video surveillance. Multimedia Tools and Applications. 2015 feb;74(3):729–742. https://doi.org/10.1007/s11042-014-2177-x.

[8] Kim HJ. Multiple vehicle tracking and classification system with a convolutional neural network. Journal of Ambient Intelligence and Humanized Computing. 2019 aug;1:3. https://doi.org/10.1007/s12652-019-01429-5.

[9] Soroush E, Mirzaei A, Kamkar S, Nasir K. Near Real-Time Vehicle Detection and Tracking in Highways; 2016. Available from: https://www.researchgate.net/publication/303352347.

[10] Sudha D, Priyadarshini J. An intelligent multiple vehicle detection and tracking using modified vibe algorithm and deep learning algorithm. Soft Computing. 2020;24. https://doi.org/10.1007/s00500-020-05042-z.

[11] Barreiros MdO, Dantas DdO, Silva LCdO, Ribeiro S, Barros AK. Zebrafish tracking using YOLOv2 and Kalman filter. Scientific

Reports 2021 11:1. 2021 feb;11(1):1–14. https://doi.org/10.1038/s41598-021-81997-9.

[12] Cao X, Shihui G, Juncong L, Wenshu Z, Minghong L. Online tracking of ants based on deep association metrics: method, dataset and evaluation. Pattern Recognition. 2020;103. https://doi.org/https://doi.org/10.1016/j.patcog.2020.107233.

[13] Gallois B, Candelier R. FastTrack: An open-source software for tracking varying numbers of deformable objects. PLOS Computational Biology. 2021 feb;17(2):e1008697. https://doi.org/10.1371/JOURNAL.PCBI.1008697.

[14] Pérez-Escudero A, Vicente-Page J, Hinz RC, Arganda S, De Polavieja GG. IdTracker: Tracking individuals in a group by automatic identification of unmarked animals. Nature Methods. 2014 jun;11(7):743–748. https://doi.org/10.1038/nmeth.2994.

[15] Wang X, Cheng E, Burnett IS, Huang Y, Wlodkowic D. Automatic multiple zebrafish larvae tracking in unconstrained microscopic video conditions. Scientific Reports. 2017 dec;7(1):1–8. https://doi.org/10.1038/s41598-017-17894-x.

[16] Závorka L, Koeck B, Cucherousset J, Brijs J, Näslund J, Aldvén D, et al. Co-existence with non-native brook trout breaks down the integration of phenotypic traits in brown trout parr. Functional Ecology. 2017 aug;31(8):1582–1591. https://doi.org/10.1111/1365-2435.12862.

[17] Kamkar S, Ghezloo F, Moghaddam HA, Borji A, Lashgari R. Multiple-target tracking in human and machine vision. PLOS Computational Biology. 2020 apr;16(4):e1007698. https://doi.org/10.1371/journal.pcbi.1007698.

[18] Luo W, Xing J, Milan A, Zhang X, Liu W, Kim TK. Multiple object tracking: A literature review. Artificial Intelligence. 2020 dec;293:103448. https://doi.org/10.1016/j.artint.2020.103448. https://arxiv.org/abs/1409.7618.

[19] Meyerhoff HS, Papenmeier F, Huff M.: Studying visual attention using the multiple object tracking paradigm: A tutorial review. Springer New York LLC.

[20] Pylyshyn Z, Storm R. Tracking multiple independent targets: evidence for a parallel tracking mechanism. Spatial vision. 1988;3(3):179–197. https://doi.org/10.1163/156856888X00122.

[21] Cheng C, Kaldy Z, Blaser E. Two-year-olds succeed at MIT: Multiple identity tracking in 20- and 25-month-old infants. Journal of Experimental Child Psychology. 2019 nov;187:104649. https://doi.org/10.1016/j.jecp.2019.06.002.

[22] Schöner G, Spencer J, Research Group D. Dynamic Thinking. Oxford University Press; 2016.

[23] Xu Y, Zhou X, Chen S, Li F. Deep learning for multiple object tracking: A survey. IET Computer Vision. 2019 jun;13(4):411–419. https://doi.org/10.1049/iet-cvi.2018.5598.

[24] Wang X, Li T, Sun S, Corchado JM.: A survey of recent advances in particle filters and remaining challenges for multitarget tracking. MDPI AG. Available from: www.mdpi.com/journal/sensors.

[25] Kamkar S, Moghaddam HA, Lashgari R, Oksama L, Li J, Hyönä J. Effectiveness of "rescue saccades" on the accuracy of tracking multiple moving targets: An eye-tracking study on the effects of target occlusions. Journal of Vision. 2020 nov;20(12):1–15. https://doi.org/10.1167/jov.20.12.5.

[26] Vater C, Kredel R, Hossner EJ. Disentangling vision and attention in multiple-object tracking: How crowding and collisions affect gaze anchoring and dualtask performance. Journal of Vision. 2017 may;17(5):21–21. https://doi.org/10.1167/17.5.21.

[27] Erlhagen W. Internal models for visual perception. Biological Cybernetics 2003 88:5. 2003 may;88(5):409–417. https://doi.org/10.1007/S00422-002-0387-1.

[28] Erlhagen W, Jancke D. The role of action plans and other cognitive factors in motion extrapolation: A modelling study. Visual Cognition. 2004;11(2-3):315–340.

[29] Fix J, Rougier N, Alexandre F. A Dynamic Neural Field Approach to the Covert and Overt Deployment of Spatial Attention. Cognitive Computation 2010 3:1. 2010 nov;3(1):279–293. https://doi.org/10.1007/S12559-010-9083-Y.

[30] Jenkins GW, Samuelson LK, Penny W, Spencer JP. Learning words in space and time: Contrasting models of the suspicious coincidence effect. Cognition. 2021 may;210:104576. https://doi.org/10.1016/j.cognition.2020.104576.

[31] Lomp O, Faubel C, Schöner G. A neural-dynamic architecture for concurrent estimation of object pose and identity. Frontiers in Neurorobotics. 2017 apr;11(APR):23. https://doi.org/10.3389/fnbot.2017.00023.

[32] Erlhagen W, Bicho E. The dynamic neural field approach to cognitive robotics*. Journal of Neural Engineering. 2006 jun;3(3):R36. https://doi.org/10.1088/1741-2560/3/3/R02.

[33] Sousa E, Erlhagen W, Ferreira F, Bicho E. Off-line simulation inspires insight: A neurodynamics approach to efficient robot task learning. Neural Networks. 2015 dec;72:123–139. https://doi.org/10.1016/J.NEUNET.2015.09.002.

[34] Wojtak W, Ferreira F, Vicente P, Louro L, Bicho E, Erlhagen W. A neural integrator model for planning and value-based decision making of a robotics assistant. Neural Computing and Applications. 2021 apr;33(8):3737–3756. https://doi.org/10.1007/s00521-020-05224-8.

[35] Martel JNP, Sandamirskaya Y. A neuromorphic approach for tracking using dynamic neural fields on a programmable vision-chip. In: ACM International Conference Proceeding Series. vol. 12-15-Sept. Association for Computing Machinery; 2016. p. 148–154.

[36] De Vangel BC, Torres-Huitzil C, Girau B. Randomly spiking dynamic neural fields. ACM Journal on Emerging Technologies in Computing Systems. 2015 apr;11(4). https://doi.org/10.1145/2629517.

[37] De Vangel BC, Torres-Huitzil C, Girau B. Event based visual attention with dynamic neural field on FPGA. In: ACM International Conference Proceeding Series. vol. 12-15-Sept. Association for Computing Machinery; 2016. p. 142–147.

[38] Spencer JP, Barich K, Goldberg J, Perone S. Behavioral dynamics and neural grounding of a dynamic field theory of multi-object tracking. Journal of Integrative Neuroscience. 2012 sep;11(3):339–362. https://doi.org/10.1142/S0219635212500227.

[39] Spencer J, Perone S. A dynamic neural field model of multi-object tracking. Journal of Vision. 2010 mar;8(6):508–508. https://doi.org/10.1167/8.6.508.

[40] Zibner SKU, Faubel C, Iossifidis I, Schöner G, Spencer JP. Scenes and tracking with dynamic neural fields: How to update a robotic scene representation. In: 2010 IEEE 9th International Conference on Development and Learning, ICDL-2010 - Conference Program; 2010. p. 244–250.

[41] Franconeri SL, Jonathan SV, Scimeca JM. Tracking multiple objects is limited only by object spacing, not by speed, time, or capacity. Psychological Science. 2010 jul;21(7):920–925. https://doi.org/10.1177/0956797610373935.

[42] ichi Amari S. Dynamics of pattern formation in lateral-inhibition type neural fields. Biological Cybernetics. 1977 jun;27(2):77–87. https://doi.org/10.1007/BF00337259.

[43] Wu S, Hamaguchi K, Amari SI. Dynamics and computation of continuous attractors. Neural Computation. 2008 apr;20(4):994–1025. https://doi.org/10.1162/neco.2008.10-06-378.

[44] Gonzalez RC, Woods RE. Digital Image Processing. 4th ed. Pearson Education Limited; 2018.

[45] Franconeri SL, Pylyshyn ZW, Scholl BJ. A simple proximity heuristic allows tracking of multiple objects through occlusion. Attention, Perception, and Psychophysics. 2012 may;74(4):691–702. https://doi.org/10.3758/s13414-011-0265-9.

[46] Ciaparrone G, Luque Sánchez F, Tabik S, Troiano L, Tagliaferri R, Herrera F. Deep learning in video multi-object tracking: A survey. Neurocomputing. 2020 mar;381:61–88. https://doi.org/10.1016/J.NEUCOM.2019.11.023.

[47] Nichols EJ, Hutt A. Neural field simulator: two-dimensional spatio-temporal dynamics involving finite transmission speed. Frontiers in Neuroinformatics. 2015;0(OCTOBER):25. https://doi.org/10.3389/FNINF.2015.00025.

[48] Jancke D, Erlhagen W. Bridging the gap: a model of common neural mechanisms underlying the Frbhlich effect, the flash-lag effect, and the representational momentum effect. In: Space and time in perception and action; 2010. .

[49] Luu T, Howe PDL. Extrapolation occurs in multiple object tracking when eye movements are controlled. Attention, Perception, and Psychophysics. 2015 aug;77(6):1919–1929. https://doi.org/10.3758/s13414-015-0891-8.