



CENTER FOR
MACHINE PERCEPTION



CZECH TECHNICAL
UNIVERSITY IN PRAGUE

BACHELOR THESIS

ISSN 1213-2365

Detection, Description and Tracking of Ants in Video Sequences

Filip Naiser

filip@naiser.cz

May 23, 2014

Thesis Advisor: prof. Ing. Jiří Matas, Ph.D.

Published by

Center for Machine Perception, Department of Cybernetics
Faculty of Electrical Engineering, Czech Technical University
Technická 2, 166 27 Prague 6, Czech Republic
fax +420 2 2435 7385, phone +420 2 2435 7637, www: <http://cmp.felk.cvut.cz>

Detection, Description and Tracking of Ants in Video Sequences

Filip Naiser

May 23, 2014

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: Filip N a i s e r

Studijní program: Otevřená informatika (bakalářský)

Obor: Informatika a počítačové vědy

Název tématu: Detekce, popis a sledování mravenců ve videosekvencích

Pokyny pro vypracování:

1. Seznamte se s problematikou segmentace a sledování více objektů a problematikou párování.
2. Seznamte se s publikovanými pracemi zabývajícími se sledováním hmyzu.
3. Navrhněte metodu pro sledování mravenců v laboratorním prostředí. Metodu implementujte.
4. Řešte případy, kdy dochází k zákrytu jedinců a jejich segmentace je obtížná.
5. Detekujte stav jedince v jednotlivých snímcích videosekvence.
6. Vyhodnoťte metodu na datech z laboratoře. Porovnejte kvalitu řešení s již existujícími metodami.

Seznam odborné literatury:

- [1] D. A. Forsyth, J. Ponce: Computer Vision: A Modern Approach. Prentice Hall, 2011.
- [2] R. Szeliski: Computer Vision: Algorithms and Applications. Springer, 2010.
- [3] L. V. Ugelvig & S. Cremer: Social prophylaxis: group interaction promotes collective immunity in ant colonies. Current Biology, 17(22), 1967-1971, 2007.

Vedoucí bakalářské práce: prof. Ing. Jiří Matas, Ph.D.

Platnost zadání: do konce letního semestru 2014/2015

L.S.

doc. Dr. Ing. Jan Kybic
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 10. 1. 2014

BACHELOR PROJECT ASSIGNMENT

Student: Filip N a i s e r

Study programme: Open Informatics

Specialisation: Computer and Information Science

Title of Bachelor Project: Detection, Description and Tracking of Ants in Video Sequences

Guidelines:

1. Get familiar with the segmentation, multi object-tracking and assignment problems.
2. Review the state-of-the-art in the problem of insect tracking.
3. Propose a method for ant tracking in lab conditions. Implement the method.
4. Address cases with mutual occlusion where ant segmentation is difficult.
5. Detect the ant state in each frame of video sequences.
6. Evaluate the method on realistic images. Compare its performance with existing solutions.

Bibliography/Sources:

- [1] D. A. Forsyth, J. Ponce: Computer Vision: A Modern Approach. Prentice Hall, 2011.
- [2] R. Szeliski: Computer Vision: Algorithms and Applications. Springer, 2010.
- [3] L. V. Ugelvig & S. Cremer: Social prophylaxis: group interaction promotes collective immunity in ant colonies. Current Biology, 17(22), 1967-1971, 2007.

Bachelor Project Supervisor: prof. Ing. Jiří Matas, Ph.D.

Valid until: the end of the summer semester of academic year 2014/2015

L.S.

doc. Dr. Ing. Jan Kybic
Head of Department

prof. Ing. Pavel Ripka, CSc.
Dean

Prague, January 10, 2014

Prohlášení autora práce

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne Podpis autora práce

I would like to express the deepest appreciation to my supervisor Jiří Matas, who has shown the attitude and the substance of a genius: he continually and persuasively conveyed a spirit of adventure in regard to research and scholarship, and an excitement in regard to teaching. Without his supervision and constant help this thesis would not have been possible. Special thanks goes to my family for supporting me during my studies.

Abstrakt

V poslední době vzniká řada studií, pozorujících chování a interakce uvnitř společenstev sociálního hmyzu. Zvyšuje se tak poptávka po metodách z oblasti počítačového vidění určených pro jeho sledování. K tomu jsou zapotřebí nástroje, plně či částečně automatizované, které usnadňují sběr dat. Představujeme metodu pro sledování hmyzu, která je založená na technikách počítačového vidění. Tato metoda byla úspěšně implementována, testována a porovnána se třemi aktuálně nejpoužívanějšími aplikacemi. Porovnávání a vyhodnocování bylo provedeno na datech z reálných experimentů. I přesto, že je původně určena pro sledování mravenců, ukazuje se, že funguje i na jiných zvířatech.

Abstract

Recently, there has been high demand for computer vision methods in the field of insect tracking and observation. Many species of social insects have been lately studied at the level of individual behaviors and their interactions inside the colony. There is a need for automatic or semiautomatic tracking tools which would simplify collection of observational data. A multi-insect tracking method, based on computer vision techniques, has been proposed, implemented and tested, along with an user interface for insect tracking. In a video recording of the experiment, identities of tracked objects are maintained, and a state description is provided for each object in a frame. Efficiency of this method has been evaluated on multiple datasets from real-world experiments, and compared to three state-of-the-art alternatives. Despite being targeted on ants, it is shown that other species can be tracked with good results as well.

Contents

1	Introduction	2
1.1	Motivation	3
1.2	Related work	3
1.3	Problem Formulation	4
2	Method	5
2.1	Interesting regions extraction	7
2.2	Assignment problem	7
2.2.1	Score functions	7
	Position score function	8
	Orientation score function	10
	Ant-likeness score function	10
2.2.2	Assignment score function formulation	10
2.3	Fitting	15
2.3.1	Optimization formulation	16
2.3.2	Adding heuristics	16
3	Datasets	21
4	Experiments	22
4.1	Tracking performance evaluation	22
4.2	Influence of background subtraction	22
4.3	Comparison of Ferda with other methods on IST datasets	23
4.4	Comparsion with idTracker	24
4.5	Measurement of certainty of algorithm decision	24
4.6	Generalization to other species	24
5	Implementation	27
	PyQt	27
	NetworkX	27
6	Future work	30
	Bibliography	31
7	Appendix	33

1 Introduction

Recently, algorithms for automatic and semi-automatic multiple insect tracking have been in high demand. An increase in the use has been documented by citations in biological papers. In this thesis, a new method in the field of insect tracking is introduced. This method is called **Ferda**¹.

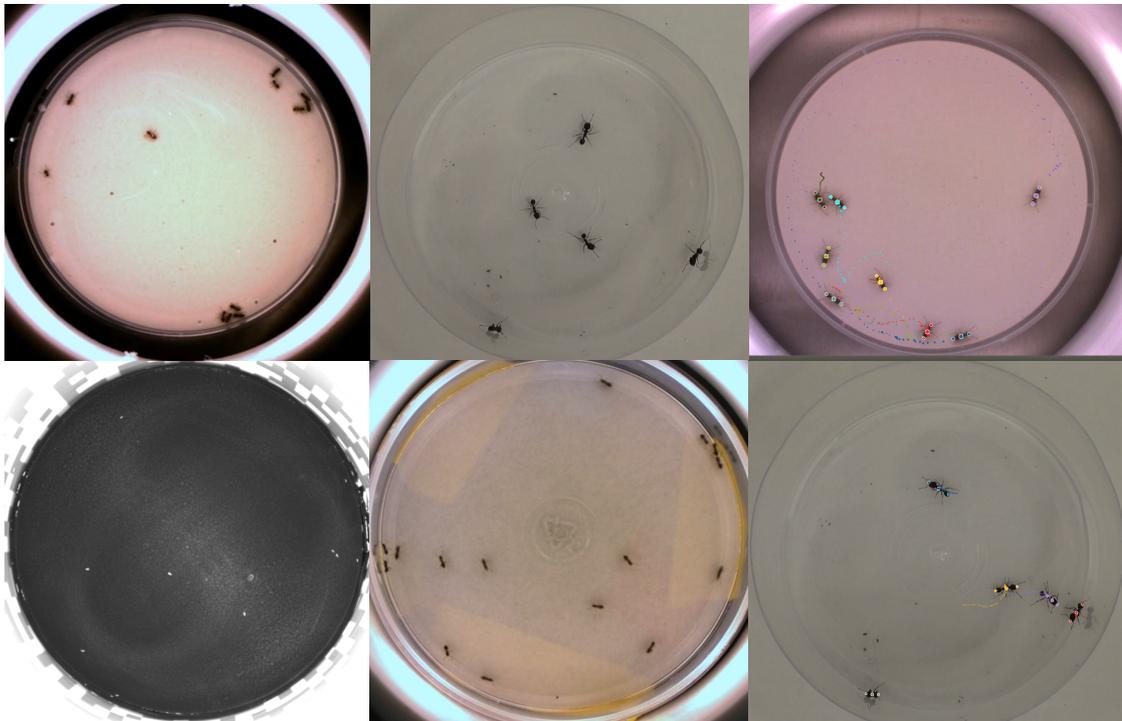


Figure 1.1 Illustration of insect tracking tasks

¹Ferda is literary character of sympathetic ant who appears in the books of Ondřej Sekora. Ferda is always optimistic guy with a red dotted ribbon around his neck. He is fearless, clever, resource full and always knows what to do. Unlike his colleague Beetle Baggins, Ferda is practical. That is a reason why "Ferda" was chosen as a name of this project. Its purpose is to be practical and helpful.

1.1 Motivation

There are interesting opportunities for application of computer science in fields like natural science and humanities. Recently, many species of social insects has been studied at the level of behavior of individuals and their interactions inside the colony. Our interest in the problem has been triggered by biologists from IST - Institute of Science and Technology Austria. The group of Sylvia Cremer at IST has been studying ant interactions. They have been using ctrax [4] method but they were dissatisfied because application output required significant time-consuming manual corrections. Here is a description of the part of the field of activity of group of Sylvia Cremer [16]:

”Colonies of social insects, like other societies, face the problem of a high risk of disease transmission among the group members. This is primarily due to close interactions and high within colony densities. Despite this risk, epidemics occur extremely rarely in the colonies of social insects (e.g. bees, ants, termites), as they have evolved collective anti-pathogen defenses that complement the individual immune systems of group members. This ”social immunity” comprises a) hygiene behaviors, such as mutual allogrooming, b) joint physiological defenses, as the production and application of antimicrobial substances and c) the modulation of interaction types and frequencies upon exposure of group members to pathogens.”

Our objective is to support biologists in facilitating the examination of various phenomena inside the world of insect using computer vision methods.

1.2 Related work

There are several approaches to insect tracking. They are categorized by various criteria. One is whether tagging of individuals is possible and whether it does not affect the behavior of observed animals. The method of marking is addressed by many researchers. Tagging using RFID is described by Henry et al. [18] or by Schneider et al. [19]. Methods using radio receivers and radars were delineated by O’Neal et al. [9] and by Keni et al. [7]). Another approach has been introduced by Mersch et al. [5] when QR codes have been used for ant tracking.

Now the computer vision based methods will be introduced. Ctrax published in 2009 in Nature Methods by Branson et al.[4] is a freely available [22] widely used tool. The method called Flydra published in 2011 by Straw et al. [10]. It describes the three-dimensional real-time tracking of multiple flying animals based on multiple-camera system. In 2013 Kimura et al. published paper [6] on multiple honey bees tracking. It is available in the application called K-Track [23]. The latest Method will be published in Nature Methods in June 2014. It is called idTracker [27]. It is also available as an application [24]. Compared to previous methods it is based on detection rather than tracking.

1 Introduction

There is also application called AnTracks [25] developed by M. Stumpe. It is commercially available and it has not been published in scientific paper. It will not appear in comparison.

1.3 Problem Formulation

There are experiments when global information (like average speed of group or frequently visited areas) is not sufficient (e.g. interactions between infected and healthy ants are observed), thus it is important to maintain identity of each tracked individual during given video sequence even if animal bodies overlap greatly and frequently.

At the highest level, the problem formulation is given:

Input: Video sequence and number of objects to be tracked

Output: Object state matrix: $N \times A \times D$
Confidence matrix $N \times A$

Where N is the number of frames, A is the number of ants and D is dimension of object description space.

Object state $s \in \{\mathbb{R}^D \cup \emptyset\}$, where \emptyset stands for undefined state.

The **quality** of the method has been measured by comparing the results with ground truth using CLEAR Multi Object Tracking metrics described in the paper published by B. Keni et al. [8] The quality is described by 5 values. The most important is **the number of mismatches**, which means that the ground truth id assigned to an object was changed, the **MOTA** score, which can be roughly understood as the ratio of correctly tracked objects to the number of all objects. and the **MOTP** score which is the mean distance to the ground truth. Detailed description is situated in Section 4.1.

Note: in the following text the word object will be interchanged with words like ants, animals or insect.

2 Method

The following sections will describe three main parts of this method. How to extract interesting regions from an image (in Section 2.1), how to connect a previous frame with the current one - which consists of assigning ants to regions (in Section 2.2) and how to deal with situations when more ants are in one region - the region fitting problem (in Section 2.3). The basic skeleton of the program is described in Algorithm 1 and shown in Figure 2.1.

Algorithm 1 Main

```
1: procedure MAIN
2:    $ants_{t_0} \leftarrow \text{INITIALIZATION}()$ 
3:    $i \leftarrow 0$ 
4:   while  $image \leftarrow \text{NEXTFRAME}()$  do
5:      $image \leftarrow \text{PREPAREIMAGE}(image)$ 
6:      $regions \leftarrow \text{MSER}(image, ants_i)$ 
7:      $ants_{i+1}, merged \leftarrow \text{ASSIGNANTSSTOREGIONS}(ants_i, regions)$ 
8:      $newRegions \leftarrow \text{FITTING}(merged)$ 
9:      $ants_{i+1} \leftarrow \text{ASSIGNRESTANTSOSPLITREGIONS}(ants_i, newRegions)$ 
10:     $i \leftarrow i + 1$ 
11:  end while
12:
13:   $\text{SAVERESULTS}(ants)$ 
14: end procedure
```

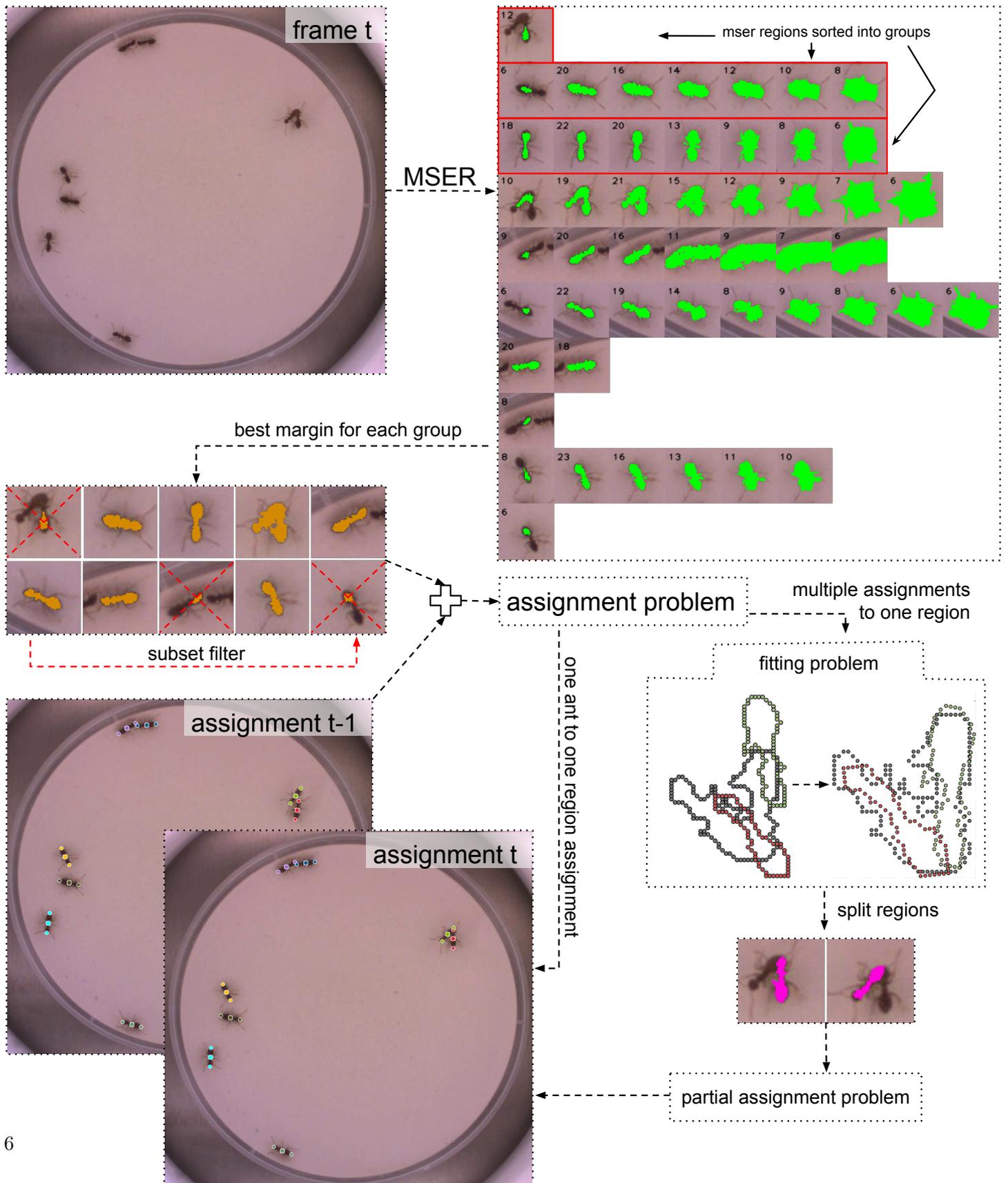


Figure 2.1 Processing a single image: main steps.

2.1 Interesting regions extraction

For a given image I a set of regions R is obtained as a result of the MSER [1] algorithm. Regions from the set R are sorted into groups G_i based on the knowledge of the connected component tree created in the MSER algorithm. Each group G_i contains only regions from a tree branch before the branch has been appended to another one (Figure 2.2).

Let us define a set S . From each group G_i the region with the highest margin is added to the set S . The margin is the number of thresholds for which the region is stable. Then the result set T is defined as: $T = \bigcup_{s \in S} s$. This filters out all subsets. The component tree is shown in Figure 2.2

Each region from the set T is described by following values computed using image moments [3]. These values are:

- **Center of mass.** In the following parts will be referred as a position or center.
- **Orientation** in degrees
- **Major and minor axis length** in pixels
- **Area** in px^2 .

2.2 Assignment problem

This part covers linking previous frame to the current one. Here is the definition of the problem.

- Let us have the sets $A = \{\text{ant states at frame } t - 1\}$ and $R = \{\text{regions returned by interesting region extraction (Section 2.1)}\} \cup \{n \text{ virtual regions}\}$, where $n = \text{number of ants}$.
- Define P as set of all possible mappings $f : A \rightarrow R$
- The goal is to find such a mapping $p \in P$ which maximizes

$$\sum_{a \in \mathbf{A}} \text{score}(a, p(a)). \quad (2.1)$$

When an ant is assigned to a region it can be described by the same parameters (position, orientation, main/minor axis and area) as introduced in Section 2.1. The assignment is known in the first frame so it is possible to use this information in the score computing function in Section 2.2.1.

2.2.1 Score functions

The general score function is defined as:

$$f(x, \sigma) = e^{\frac{-x}{2\sigma^2}} \quad (2.2)$$

where x is measured magnitude and σ defines the rate of descent of the exponential function. This definition assures that the score values will be in the interval $(0; 1)$.

2 Method

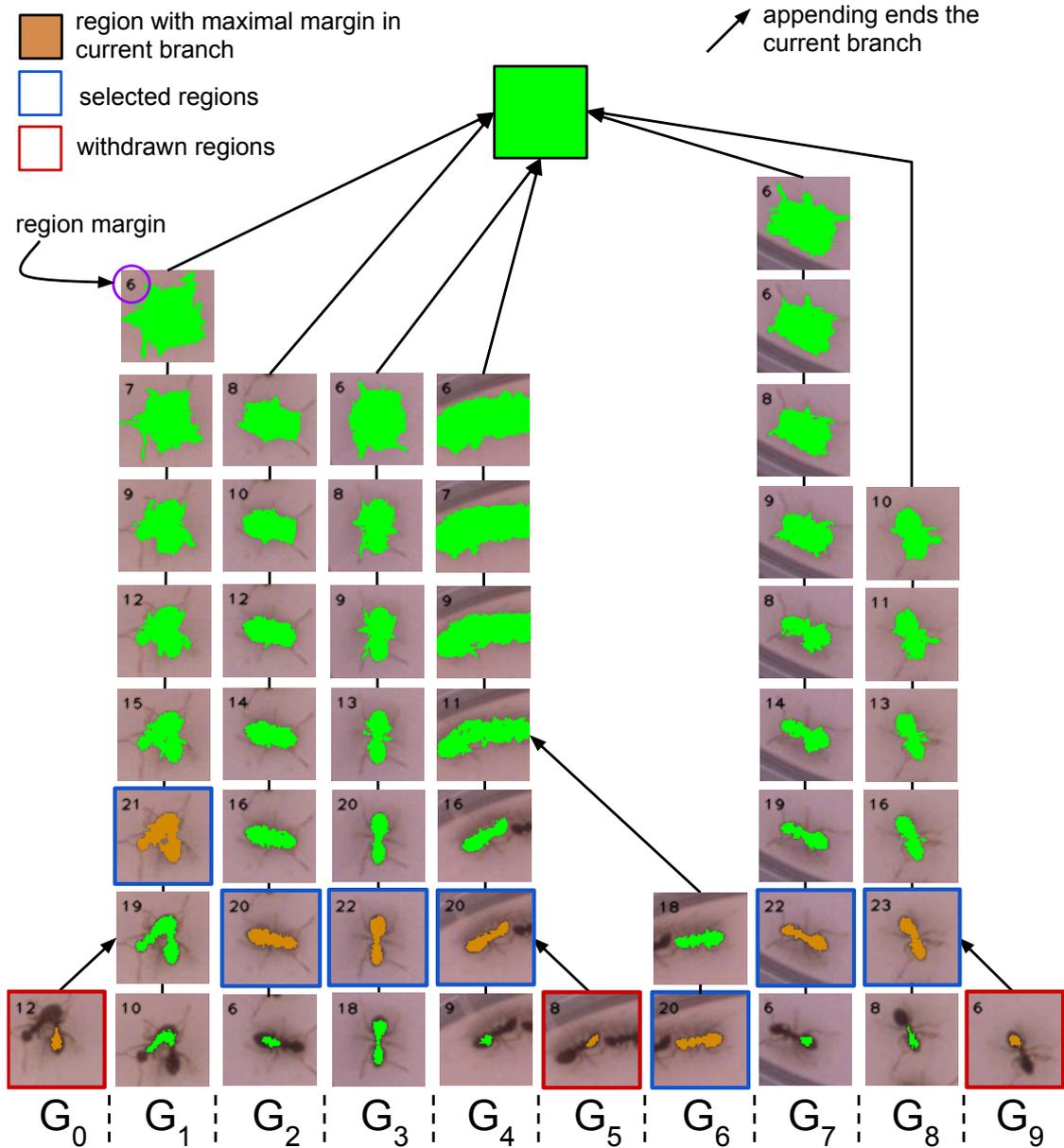


Figure 2.2 MSER connected component tree

Position score function

The first idea is that the best score has a pair (a, p) whose euclidean distance between centers is smallest. In Figure 2.3 it could be seen that this approach does not work in cases when the individuals are passing each other in a close distance. That is the reason

why the movement prediction is needed.

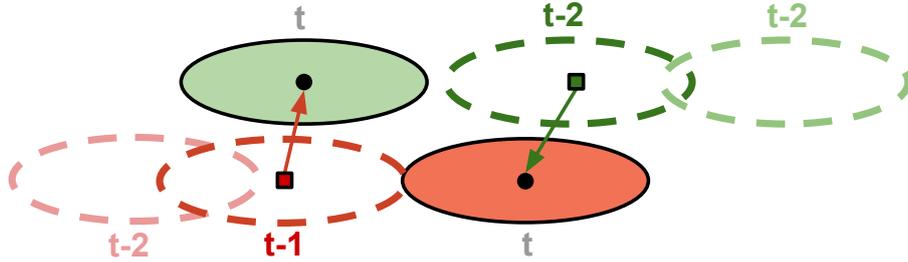
Let us define a simple function for linear movement prediction:

$$\text{prediction}(c_{t-1}^{\vec{c}}, c_{t-2}^{\vec{c}}) = c_{t-1}^{\vec{c}} - c_{t-2}^{\vec{c}}, \quad (2.3)$$

where \vec{c}_t is a position of an ant in the frame t . The position score has been defined as:



a)



b)

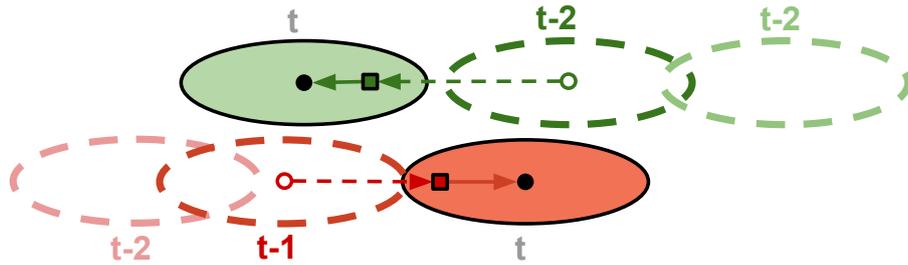


Figure 2.3 Situation of close ant passing. The color of regions shows the right solution of this situation. **a)** Ants in $t - 1$ are wrongly assigned to the nearest region in time t . **b)** Ants in $t - 1$ are assigned to the nearest region from their movement prediction based on movement between frames $t - 2$ and $t - 1$.

$$\begin{aligned} \text{position}(a, p(a)) &= f(x, \sigma) \\ x &= \|\text{prediction}(\vec{ac}_{t-1}, ac_{t-2}) - rc_t\|_2, \end{aligned} \quad (2.4)$$

where ac_t is a position of an ant in frame t , rc_t is a position of a region and σ is a hyperparameter which was set on the basis of observations to the average of the main axis of regions tracked as animals.

Orientation score function

Sometimes the linear movement prediction is not good enough. To improve the robustness of assignment, the information about the object orientation is being used. The orientation score function is formulated as:

$$\begin{aligned} \text{orientation}(a, p(a)) &= f(\theta, \sigma) \\ \text{theta} &= |a_\theta - r_\theta| \end{aligned} \tag{2.5}$$

where a_θ is the ant orientation at frame $t - 1$, r_θ is the region orientation at frame t , f is the general score function (2.2) and σ is a hyperparameter set to 0.8095 based on observations.

Ant-likeness score function

In order to promote the regions which are similar to the regions labeled as ant-like in the first frame, the ant-likeness function is introduced.

$$\text{antlikeness}(p(a)) \rightarrow \langle 0; 1 \rangle \tag{2.6}$$

For measuring the ant-likeness the area size and main axis length of a measured region are used. More precisely they are normalized which guarantees the scale invariance.

In the figures 2.4 and 2.6 the distribution of ant-like regions inside the space of area sizes and main axis lengths divided by the average of these variables is shown. As it can be seen in Figure 2.4, the ant-like regions are concentrated in one cluster. The lookup table has been created. The ant-like region has been manually labeled and inserted into the 2D space. After that, the space has been divided into boxes and the histogram has been computed. Then each histogram value has been increased by one and logarithmized. At the end the result is blurred using Gaussian blur and normalized. The lookup table is shown in Figure 2.5

2.2.2 Assignment score function formulation

In order to grasp the whole problem of selecting the best mapping, it is converted onto a graph theory problem - the matching [2]. Due to the nature of matching problem, it is impossible to assign multiple ants to one region. To allow this behavior, the new virtual regions must be added to the set R defined in Section 2.1 and another score function will be added. The rule for adding new regions is as follows.

For each region $r \in R$ the value m is defined as area of region r divided by average area of tracked ants. Then for each such region r the n derived regions rv_i are added

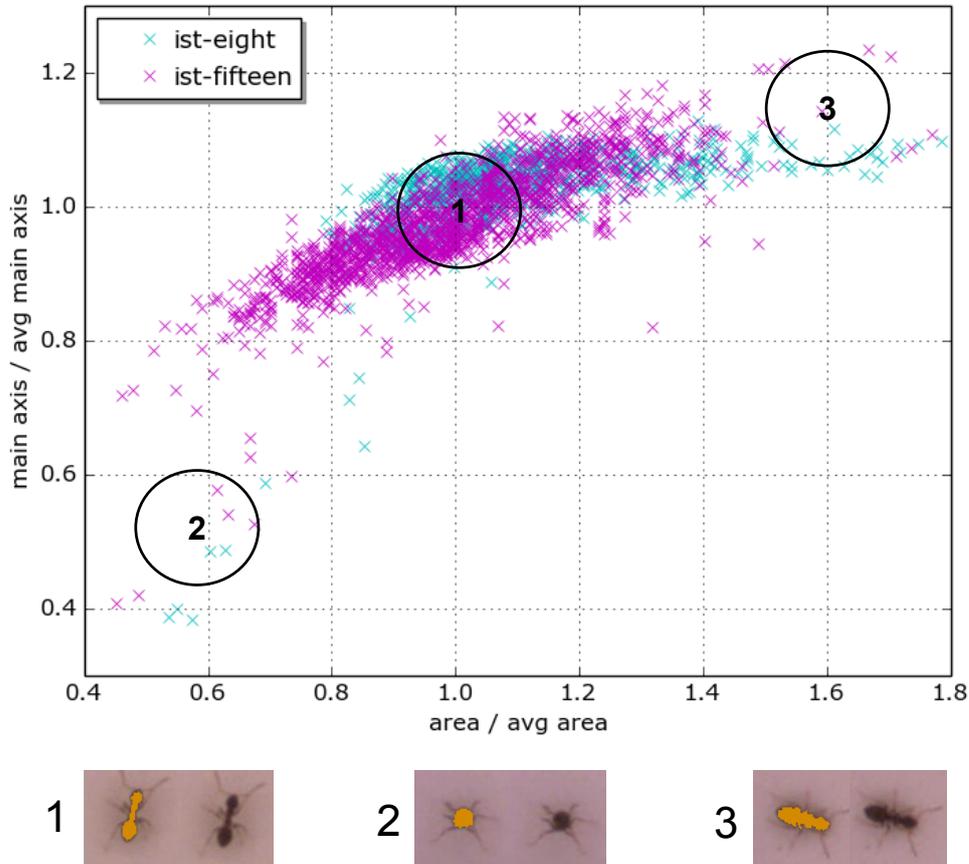


Figure 2.4 Examples of ant-like regions with different values of the normalised main axis and area. Ant-like regions have been manually labeled in the **ist-eight** and **ist-fifteen** sequences.

(they are indexed from 1), where n equals the m rounded up and subtracted by one. The original region's id is stored for each new derived region. Each region even the original one carries an area score. Original region score = $\max(1, m)$. Derived region r_i has score = $\max(1, i + 1 - m)$ (e.g. region with $m = 1.9$ obtains score 1 and new derived region with area score 0.9 is be added).

2 Method

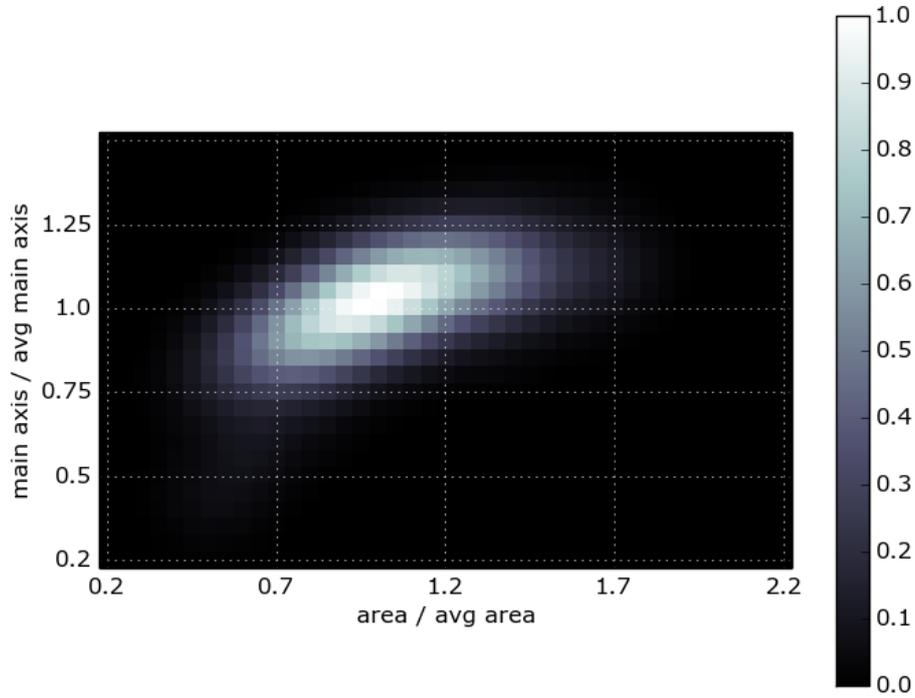


Figure 2.5 Ant-likeness histogram

Complete score function is defined as:

$$score(a, p(a)) = \begin{cases} position(a, p(a)) \cdot area(p(a)) + C & : \textit{original region} \\ position(a, p(a)) \cdot area(p(a)) & : \textit{derived region} \\ \textit{undefined state value} & : \textit{virtual region} \end{cases} \quad (2.7)$$

$$C = position(a, p(a)) \cdot angle(a, p(a)) \cdot antlikeness(p(a))$$

Where component C ensures that if there is a good unoccupied ant-like region in a reasonable distance, it will be preferred to merged region for which the C value will be decreased by ant-likeness and by orientation. The undefined state value is cost for the ant state "I don't know". This parameter is set to 0.01. The example of assign problem is visualized in Figure 2.7 and more detailed description can be found in Algorithm 2.

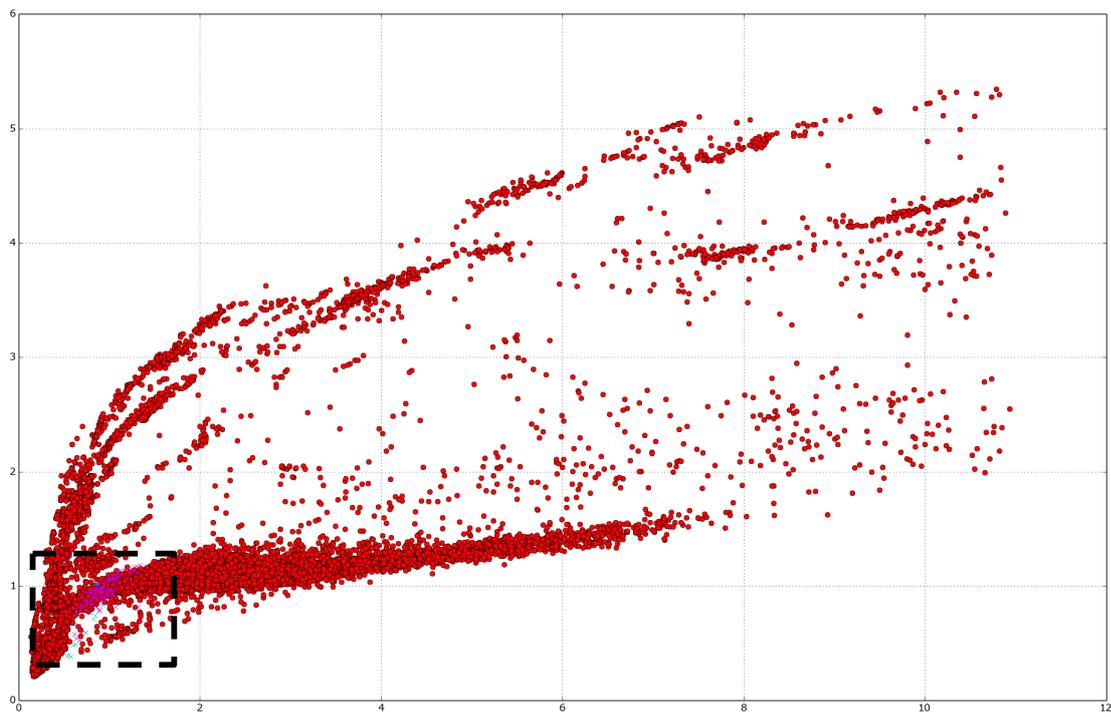


Figure 2.6 Distribution of all regions from the MSER algorithm in the space of relative area sizes and main axis lengths. Selection shows the region where the ant-like regions are placed.

Algorithm 2 Assignment to Regions

```

1: procedure ASSIGNANTSSTOREGIONS(regions, ants)
2:    $g \leftarrow \text{PREPAREGRAPH}(\textit{regions}, \textit{ants})$ 
3:    $\textit{results} \leftarrow \textit{graph}.\text{MAXIMUMMWEIGHTEDMATCHING}()$ 
4:    $\textit{antToRegionPairs} \leftarrow \text{GETUNAMBIGUOUSASSIGNMENT}(\textit{results})$ 
5:    $\textit{toBeSplit} \leftarrow \text{GETASSIGNMENTTOBESPLIT}(\textit{results})$ 
6:
7:   return  $\textit{antToRegionPairs}, \textit{toBeSplit}$ 
8: end procedure
9:
10: procedure PREPAREGRAPH(regions, ants)
11:    $\textit{graph} \leftarrow \text{GRAPH}()$ 
12:    $\textit{uc} \leftarrow \textit{parameters}.\text{UndefinedStateCost}$   $\triangleright$  parameters - global variable
13:   for all  $\textit{ant} \in \textit{ants}$  do
14:      $\textit{graph}.\text{ADDNODE}(\text{"a"} + \textit{ant}.\text{id})$   $\triangleright$  i.e. a0, a1, ...
15:      $\textit{graph}.\text{ADDNODE}(\text{"u"} + \textit{ant}.\text{id})$   $\triangleright$  "u" means undefined, i.e. 'u1'
16:      $\textit{graph}.\text{ADDEDGE}(\text{"a"} + \textit{ant}.\text{id}, \text{"u"} + \textit{ant}.\text{id}, \textit{uc})$   $\triangleright$  From, To, EdgeCost
17:   end for
18:   for all  $r \in \textit{regions}$  do
19:      $a \leftarrow r.\text{area} / \textit{parameters}.\text{averageAntArea}$ 
20:      $\textit{ac} \leftarrow \text{CEIL}(a)$ 
21:     for  $i \leftarrow 1; i \leq \textit{ac}; i \leftarrow i + 1$  do
22:        $\textit{graph}.\text{ADDNODE}(\text{"r"} + r.\text{id} + "-" + i)$   $\triangleright$  i.e. r1-0, r1-1...
23:       for all  $\textit{ant} \in \textit{ants}$  do
24:          $\textit{distanceScore} \leftarrow \text{GETDISTANCESCORE}(\textit{ant}, r)$ 
25:          $\textit{areaScore} \leftarrow \text{MIN}(1, 1 + a - \textit{ac})$ 
26:
27:         if  $i == 1$  then
28:            $\textit{angleScore} \leftarrow \text{GETANGLESCORE}(\textit{ant}, r)$ 
29:            $\textit{antlikeScore} \leftarrow \text{GETANTLIKESCORE}(r)$ 
30:            $\textit{score} \leftarrow \textit{distanceScore} \cdot (\textit{areaScore} + \textit{angleScore} \cdot \textit{antlikeScore})$ 
31:         end if
32:          $\textit{score} \leftarrow \textit{distanceScore} \cdot \textit{areaScore}$ 
33:         if  $\textit{score} > \textit{uc}$  then
34:            $\textit{graph}.\text{ADDEDGE}(\text{"a"} + \textit{ant}.\text{id}, \text{"r"} + r.\text{id} + "-" + i, \textit{score})$ 
35:         end if
36:       end for
37:     end for
38:   end for
39:
40:   return  $\textit{graph}$ 
41: end procedure

```

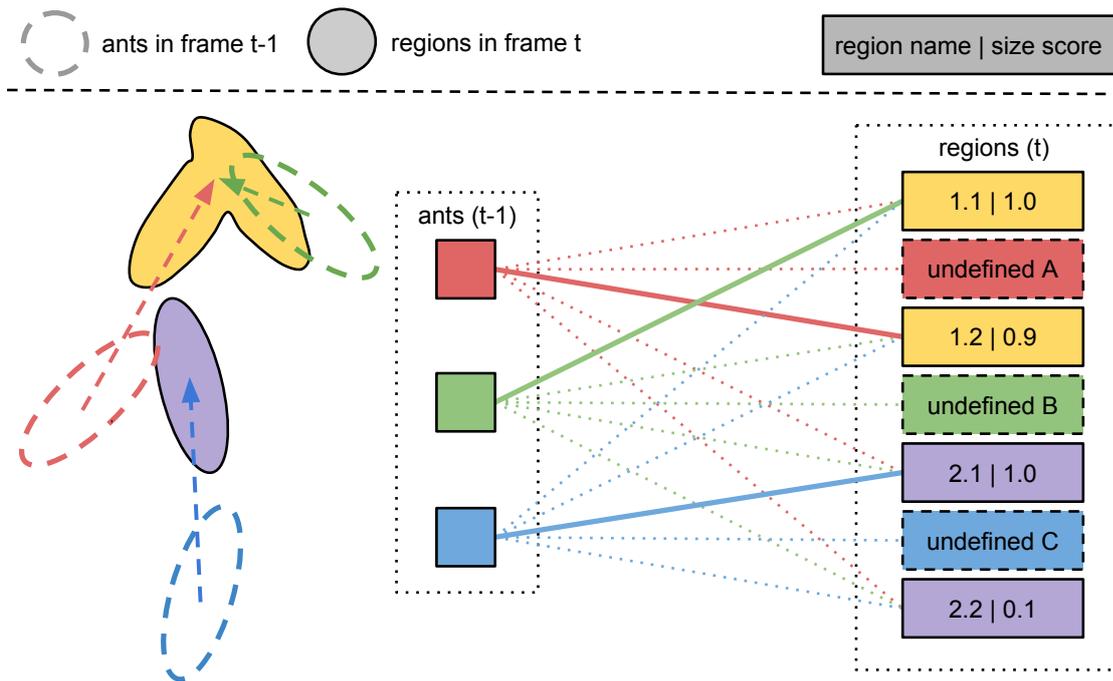


Figure 2.7 Ants to region assignment problem using maximum-weighted matching. On the left side situation is described, dashed arrows showing final assignment. The right side shows constructed graph. The solid lines shows maximum weighted matching results.

2.3 Fitting

When it is decided which ant belongs to which region it is time to take regions with multiple ants as signed and fit the ants inside.

Several approaches have been tested. Region of the body of ant can be easily approximated by an ellipse so there were several approaches using some kind of ellipse fitting. The k-means [11] modification failed due to problems with points sharing thus the cases with high overlaps of bodies had poor results. The EM [12] based algorithm had similar problem with points sharing as k-means and moreover there was a problem how to restrict the ellipse size.

Branch and bound [13] approach failed due combination of two factors. High performance cost of loss function and high dimension of searched space with several local minima so the number of samples must be high. (For n ants it is $(x \cdot y \cdot \theta)^n$, where x/y is the number of samples on the x/y axis and theta is number of samples on orientation).

2.3.1 Optimization formulation

The problem of fitting n ants (from frame $t-1$) into a single region (in frame t) is simplified in the following way. Ants and the merged region are represented by their contour. Let us define the sets $C_r = \{\text{region contour points}\}$, $C_{ai} = \{\text{ant}_i \text{ contour points}\}$ and set $O = \{\forall a \in C_{ai} : a \text{ is outside the region}\}$.

The goal is to minimize the distance of each $r \in C_r$ to the nearest point from sets C_{ai} . Additionally, distance of points $a \in O$ to the nearest point $r \in C_r$ is also minimized. The loss function is defined as:

$$f(C_{ai}, C_r, O) = \sum_{r \in C_r} \min_{A \in C_a} \min_{a \in A} \|r - a\| + \sum_{a \in O} \min_{r \in R} \|a - r\| \quad (2.8)$$

where C_{ai} is understood as a set of all ant's contours.

While searching for nearest points (from each $r \in C_r$ to $a \in C_{ai}$ and from each $a \in O$ to region contour points) the correspondences between points are established.

These correspondences are used for finding the ants transformation in iterative process of optimizing. The transformation - rotation and translation for each ant is being computed using process described in the paper published by O. Sorkine [14]. The algorithm of this part is described in Algorithm 3.

2.3.2 Adding heuristics

In most cases, ants are not moving fast during interactions and the process described in Section 2.3.1 is working. Unfortunately, sometimes they are moving fast and then this method fails as it is shown in Figure 2.9 in parts a, b and c. The appearance of this situation in video sequence is shown in Figure 2.8. The fitness heuristics is formulated as follows:

At the beginning of the fitting process, the fitness score is computed for each ant. Each ant whose score is below some threshold (using value 0.75) is being classified as unsettled and then is optimized based on the point references similarly to Section 2.3.1. The fitness score is formulated as:

$$fitness(C_{ai}, C_r) = \frac{1}{l} \sum_{a \in C_{ai}} g(\text{nearestPointFrom}(C_r, a), a) \quad (2.9)$$

$$g(r, a) = \begin{cases} 1 & \|a - r\| < \epsilon \\ 0 & a \in O \\ -1 & \text{else} \end{cases}$$

where l is the length of ant contour and sets C_{ai} , C_r and O have the same meaning as in the previous section 2.3.1.

The change consists in the fact that the points from ants labeled as settled cannot be assigned to the region points except the case when the distance is below some threshold ϵ .

As it can be seen, it allows the uninterpreted parts of a region contour to be interpreted by ants classified as unsettled. The function h describes whether the point a can be assigned to a region point r .

$$h(r, a) = \begin{cases} False & a \in S \text{ and } \|a - r\| > \epsilon \\ False & \|a - r\| > \tau \\ True & \text{else} \end{cases} \quad (2.10)$$

$$S = \{\forall a \in Ants : \text{ant is classified as settled}\}$$

The algorithm of this part is described in Algorithm 4 and the impact of this heuristic can be seen in Figure 2.9

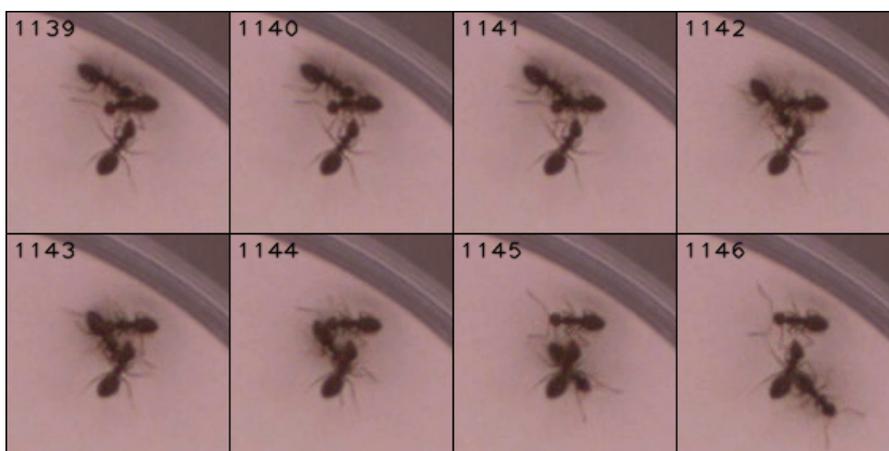


Figure 2.8 Frames 1139-1146 from the **ist-eight** dataset demonstrating difficult situations for tracking.

2 Method

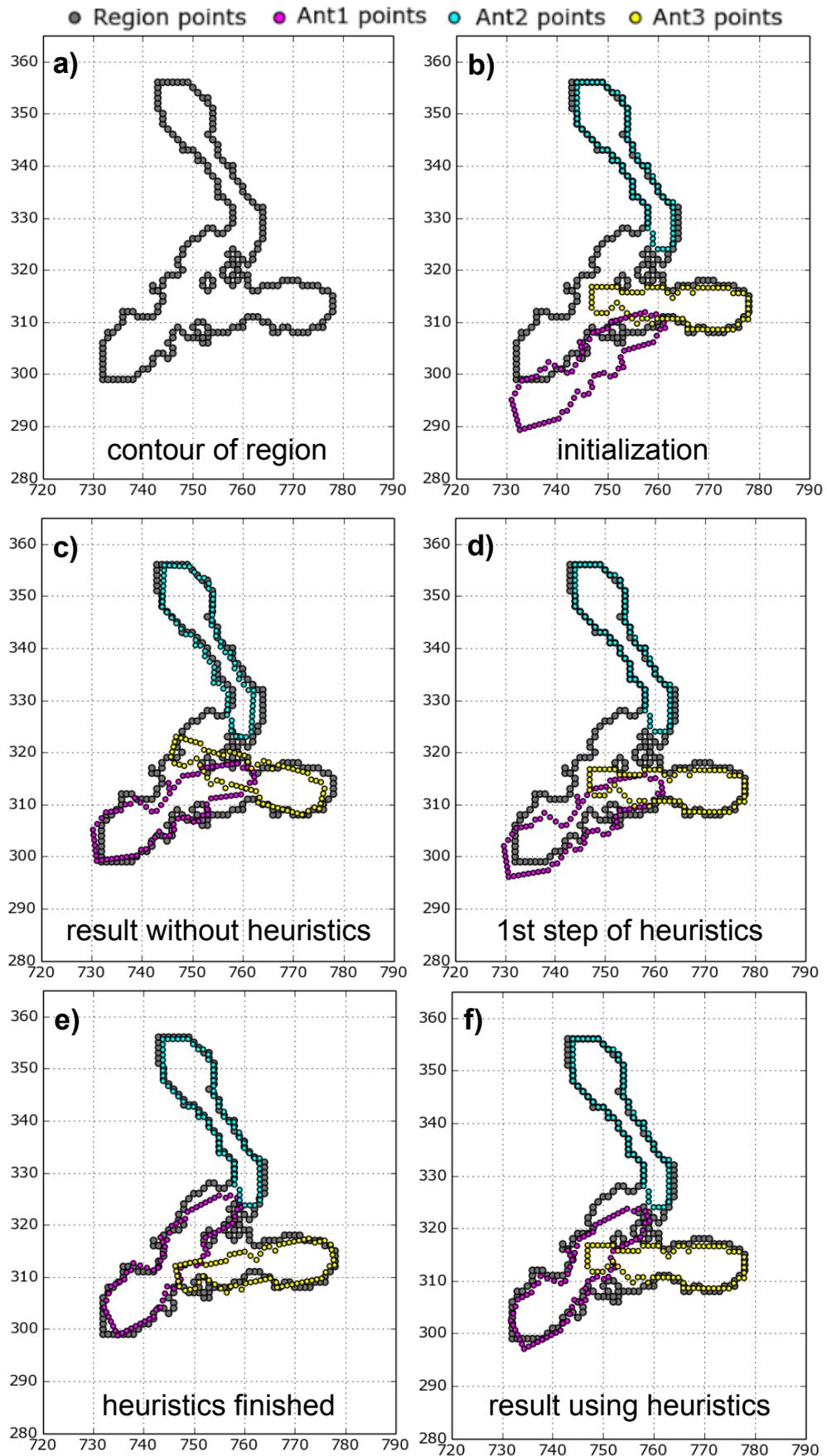


Figure 2.9 Fitting results, frame 1142, ist-eight seq. Cyan and yellow ants classified as settled.

Algorithm 3 Fitting

```

1: procedure FITTING(region, ants)
2:   ants[:].cont, region.cont  $\leftarrow$  GETCONTOURS(ants, region)
3:   ants[:].dmap, region.dmap  $\leftarrow$  CREATEDISTANCEMAPS(ants, region)
4:
5:   ants, region  $\leftarrow$  MOVEUNSETTLED(ants, region) ▷ heuristics
6:   while True do
7:     for all a  $\in$  ants do
8:       protrudingPairs[:,ps][:]  $\leftarrow$  MATCHPROTRUDINGANTSSTOREGION(a, region)
9:     end for
10:    ids  $\leftarrow$  SORTANTSBY(penalty, ants) ▷ descending, worst first
11:    penalty  $\leftarrow$  penalty + sum(ps)
12:
13:    for all id  $\in$  ids do
14:      regionPairs[:,p]  $\leftarrow$  MATCHREGIONTOANTS(ants, region)
15:      penalty  $\leftarrow$  penalty + p
16:      t  $\leftarrow$  FINDTRANSFORMATION(ants[id], protrudingPairs, regionPairs)
17:      ants[id]  $\leftarrow$  APPLYTRANSFORMATION(ants[id], t)
18:    end for
19:
20:    if TESTCONVERGENCE(ants, penalty) then
21:      break
22:    end if
23:  end while
24:
25:  return ants
26: end procedure
27:
28: procedure MATCHPROTRUDINGANTSSTOREGION(region, ant)
29:   pairs  $\leftarrow$  [], penalty  $\leftarrow$  0
30:   for all apt  $\in$  ant.cont do
31:     if ISOUTSIDEREGION(apt) then
32:       pt, d  $\leftarrow$  GETNEARESTPOINT(region, apt)
33:       pairs.APPEND([apt, pt])
34:       penalty  $\leftarrow$  penalty + d
35:     end if
36:   end for
37:   return Pairs, Penalty
38: end procedure
39:
40: procedure MATCHREGIONTOANTS(region, ants)
41:   pairs  $\leftarrow$  [], penalty  $\leftarrow$  0
42:   for all rpt  $\in$  region.cont do
43:     pt, d  $\leftarrow$  GETNEARESTPOINT(ants, rpt)
44:     pairs.APPEND([rpt, pt])
45:     penalty  $\leftarrow$  penalty + d
46:   end for
47:   return Pairs, Penalty
48: end procedure

```

2 Method

Algorithm 4 MoveUnsettled

```

1: procedure MOVEUNSETTLED(region, ants)
2:   settled  $\leftarrow$  []
3:   for all a  $\in$  ants do
4:     counter  $\leftarrow$  0
5:     for all apt  $\in$  a.cont do
6:       if DISTTONEARESTPOINT(region, apt)  $<$  params.settledDist then
7:         counter  $\leftarrow$  counter + 1
8:       else if ISOUSIDEREION(region, apt) then
9:         counter  $\leftarrow$  counter - 1
10:      end if
11:    end for  $\triangleright$  settledThreshold is as default 0.75
12:    if counter/LENGTH(a.cont)  $<$  params.settledThreshold then
13:      settled.APPEND(a)
14:    end if
15:  end for
16:  if LEN(settled) == LEN(ants) then
17:    return  $\triangleright$  nothing to do
18:  end if
19:
20:  while True do
21:    for all a  $\in$  settled do
22:      protrudingPairs  $\leftarrow$  MATCHPROTRUDINGANTSSTOREGION(a, region)
23:      regionPairs  $\leftarrow$  MATCHREGIONTOANTSHEURISTICS(ants, region)
24:      t  $\leftarrow$  FINDTRANSFORMATION(a, protrudingPairs, regionPairs)
25:      a  $\leftarrow$  APPLYTRANSFORMATION(a, t)
26:    end for
27:    if TESTCONVERGENCE(ants) then
28:      break
29:    end if
30:  end while
31: end procedure
32:
33: procedure MATCHREGIONTOANTSHEURISTICS(region, ants)
34:   pairs  $\leftarrow$  []
35:   for all rpt  $\in$  region do bestPt  $\leftarrow$  Null, bestDist  $\leftarrow$   $\infty$ 
36:     for all a  $\in$  ants do pt, d = GETNEARESTPOINT(a, rpt)
37:       if d  $<$  bestDist  $<$  params.settledThreshold then
38:         bestPt  $\leftarrow$  pt, bestD  $\leftarrow$  d
39:       else if d  $<$  bestDist  $<$  params.unsettledMaxDist then
40:         bestPt  $\leftarrow$  pt, bestD  $\leftarrow$  d
41:       end if
42:     end for
43:     if bestPt is not None then
44:       pairs.append(bestPt)
45:     end if
46:   end for return pairs
47: end procedure

```

3 Datasets

Name	Resolution	Frames	FPS	Objects
ist-eight	1280 x 1024	1502	20	8
ist-fifteen	800 x 800	2262	25	15
idTracker-zebrafish	1920 x 1080	14880	32	5
idTracker-mesor- structor	1920 x 1080	51050	25	5
ctrax-drosophyla	1024 x 1024	100	20	10

Table 3.1 Datasets used in evaluation of Ferda.

The sequences are provided by Institute for Science and Technology (**ist-eight**, **ist-fifteen**) and by the Cajal Institute (Consejo Superior de Investigaciones Científicas) in Madrid, Spain by the team working on idTracker [27] (the sequences idTracker-zebrafish and idTracker-mesor-structor). The ctrax-drosophyla video sequence is available at Ctrax application web pages [4].

4 Experiments

4.1 Tracking performance evaluation

In video sequences where ground truth data are available the performance of tracking is being evaluated using M. Smid implementation [15] of CLEAR Multi Object Tracking based on paper published by B. Keni et al. [8]. Parameters describing tracking performance are:

- **MOTA** can be roughly understood as a ratio of correctly tracked objects. Bigger / closer to 1 is better.
- **MOTP** is mean distance to ground truth / mean error. Lower is better.
- **FN** occurs when no ground truth in defined distance(precision) is found.
- **FP** occurs when there is no object in defined distance(precision) for ground truth.
- **MC** - mismatches count. One mismatch occurs when measurement id assigned to a gt id changes. E.g. identity switch in one frame equals to 2 identity mismatches.

In all experiments, unless stated otherwise, precision 7 is used due to inaccuracy of ground truth data.

4.2 Influence of background subtraction



Figure 4.1 MSER leakage example, in frame 45 of ist-fifteen sequence.

Sometimes segmentation using MSERs was inaccurate or even wrong due to the dark spots on the arena border and there was leakage of object into borders (see Figure 4.4). Background subtraction was suggested as a treatment of this leakage. The efficiency of this approach is shown in Table 4.1. If there are big changes on the background during

4.3 Comparison of Ferda with other methods on IST datasets

experiment, the subtraction fails. This case is not solved in this thesis but it has not proven to be a problem because in laboratory conditions it does not occurs often.

	modification	MOTA	MOTP	FN	FP	MC
ist-eight	MSERs	0.8882	0.8383	670	670	7
	MSERs + bg subtraction	0.9972	0.8545	15	15	4
ist-fifteen	MSERs	0.7947	0.8765	3475	3475	18
	MSERs + bg subtraction	0.9975	0.8590	38	38	9

Table 4.1 Effectivity of Ferda algorithm in different configurations.

4.3 Comparison of Ferda with other methods on IST datasets

Ferda algorithm has been compared with others on two sequences with ground truth introduced in Section 3. The evaluation has been done using CLEAR MOT metrics [8] as it has been described in s Section 4.1. There were two inconveniences. The first one - K-Track crashed at 813 frame in **ist-eight** sequence which has 1502 frames and the second one is that idTracker crashed on sequence **ist-fifteen**. It should be also noted that idTracker is fresh new method and as authors claimed, the application available at idTracker [27] websites is not the final version. Results are shown in Table 4.2.

	tracker	MOTA	MOTP	FN	FP	MC
ist-eight	Ctrax	0.9456	0.0485 ¹	223	423	9
	idTracker	0.7747	0.8829	2704	5	6
	K-Track	0.9608	2.2282	147	104	4 ²
	Ferda	0.9972	0.8545	15	15	4
ist-fifteen	Ctrax	0.9858	0.1140 ³	159	292	31
	idTracker	-	-	-	-	-
	K-Track	0.8946	2.9892	2536	1010	11
	Ferda	0.9975	0.8590	38	38	9

Table 4.2 Comparison of Ferda with other methods. Ferda shows best results except for FP value on **ist-eight** sequence. Green the best, red the worst, gray has issues.

¹Ground truth has been created correcting Ctrax data. That is the reason for such a small MOTP value.

²The mismatches count is 4 but the K-Track crashed at frame 813/1502

³The same as in 1

4 Experiments

In figures 7.6, 7.7 the distribution of false positives, false negatives and identity mismatches is displayed.

4.4 Comparison with idTracker

Ferda has been evaluated also on sequence idTracker-messor-structor provided by idTracker developers [27] with ground truth for first 13 512 frames. Due the higher resolution of this sequence, the precision for CLEAR MOT metrics has been set to 20. In Table 4.3 the results are shown. As it can be seen, there are no mismatches for Ferda. The high number of FP and FN is due to incompleteness of ground truth data. It is based on the data evaluated by idTracker (which also explains the 0.0 for MOTP for idTracker) and whenever the idTracker returns an undefined state it is undefined also in GT which adds up one FN for idTracker but one FN and FP for Ferda. The most interesting information from this experiment is that Ferda maintained ants identities without any mismatch from frame 1 to frame 13 512 (the length of ground truth).

tracker	MOTA	MOTP	FN	FP	MC
idTracker	0.9562	0.0	2951	0	5
Ferda	0.9114	8.3791	2992	2992	0

Table 4.3 The Ferda method compared to idTracker on **idTracker-messor-structor** sequence.

4.5 Measurement of certainty of algorithm decision

In each frame for each ant the certainty of assignment is determined as the score defined in equation 2.8 which is being used for finding the assignment in maximum weighted matching. The important thing is that whenever the error has been evaluated based on ground truth, the certainty was low. That opens the door for future work where more complex detection of frames to be checked will be done. The correlation between uncertainty and errors can be seen in Figure 7.8.

4.6 Generalization to other species

To test whether the Ferda algorithm is applicable to tracking other species two experiments have been performed. Ferda has been tested on video **ctrax-drosophyla** where the tracked objects are really small. The sequence is not very long but as a test of scale invariance it served well. On sequence with zebra fishes it has been working perfectly in situations without crossings. When body overlaps occur the fitting problem works only if the body shape was not changing.

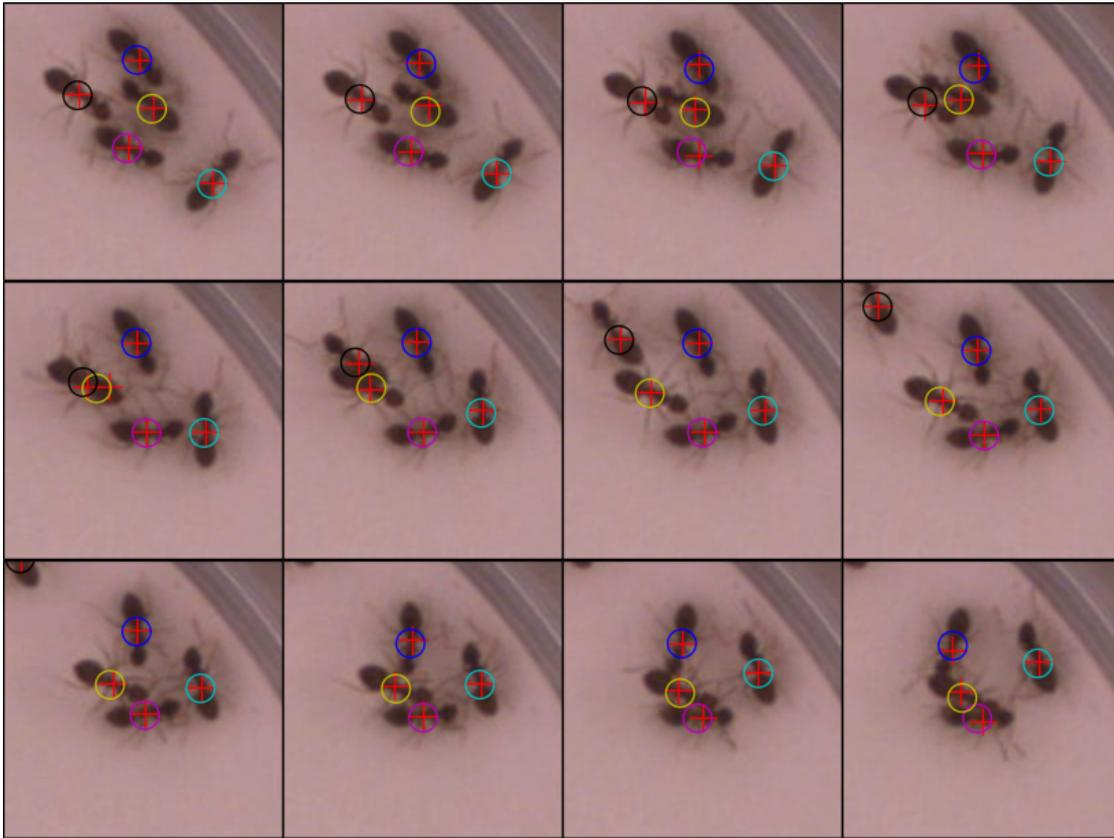


Figure 4.2 Results of Ferda on **ist-eight** sequence, frames 668 - 680. Red crosses stands for ground truth and circles for Ferda results. In this subsequence there is a identity mismatch error between black and yellow ant.

However to test this hypothesis, more video sequences are needed. Figures from **ctrax-drosophyla** and **id-tracker-zebrafishes** tracking are situated in appendix (figures 7.4, 7.5).

4 Experiments

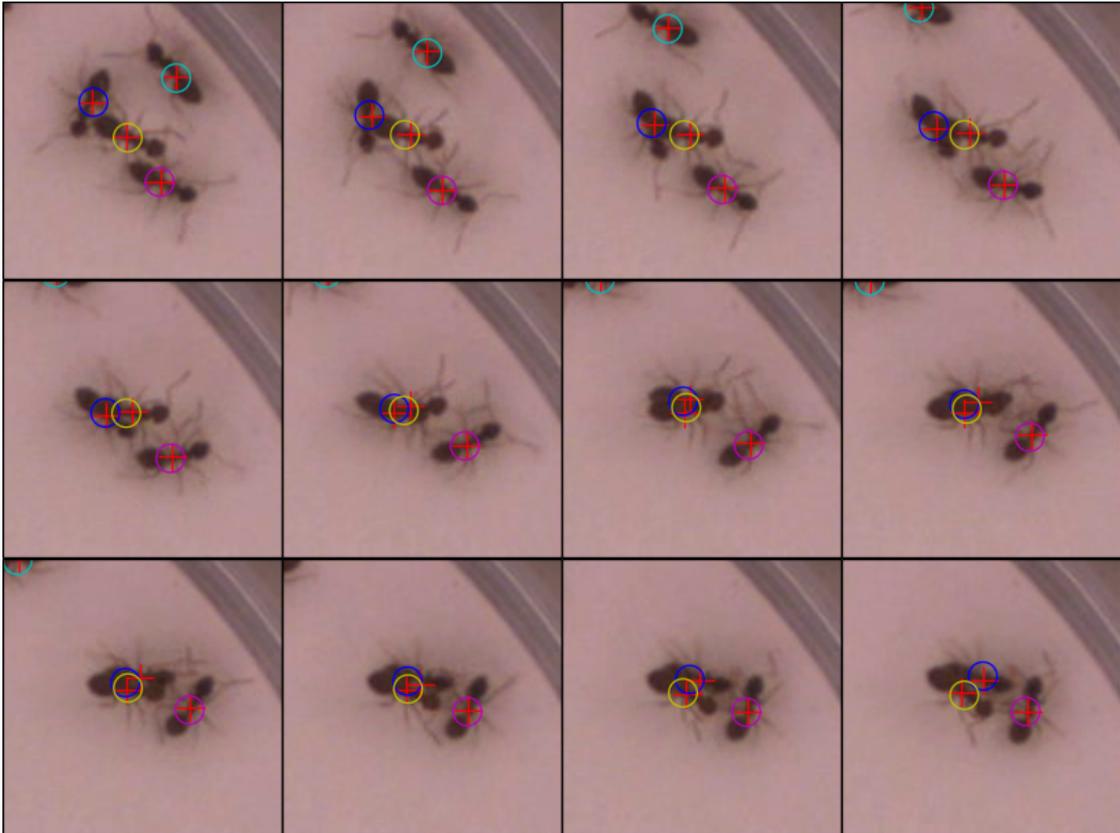


Figure 4.3 Results of Ferda on **ist-eight** sequence, frames 681 - 693. Red crosses stands for ground truth and circles for Ferda results. In this subsequence there is a identity mismatch error between blue and yellow ant.

5 Implementation

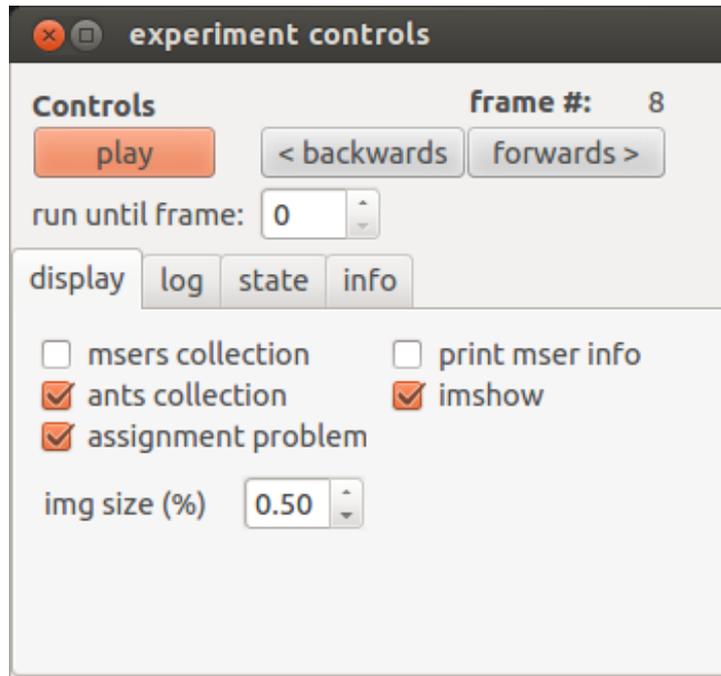


Figure 5.1 GUI for experiment controls in Ferda implementation.

The main part of the application is written in Python because it is a language that allows very rapid development whose biggest downside lies in slow execution times of some commands. That is the reason why for the crucial parts (image processing etc.) the c++ libraries are used (e.g. OpenCV[17], MSER[1] algorithm).

PyQt

PyQt [20] is a Python binding of the cross-platform GUI toolkit Qt. It is one of Python's options for GUI programming. The arrangement of GUI has been designed in QtDesigner and then wrapped by pyuic4¹ script for usage in python.

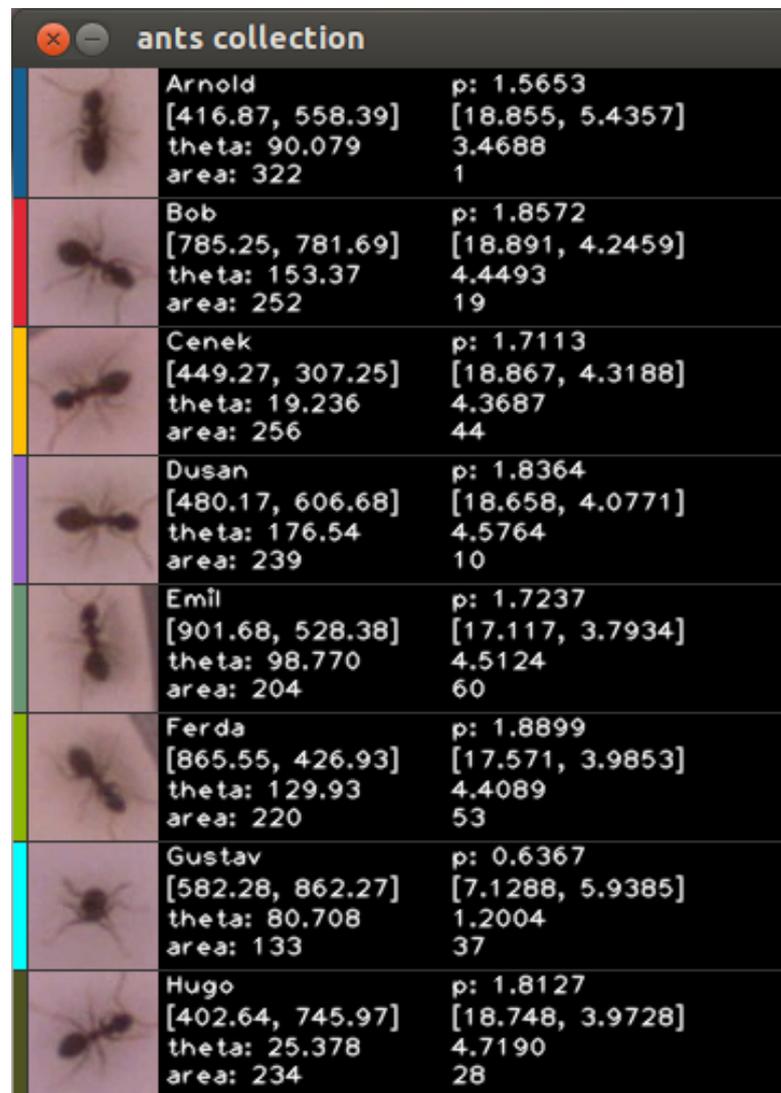
NetworkX

NetworkX [21] is a Python language software package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks. This library

¹<http://pyqt.sourceforge.net/Docs/PyQt4/designer.html#pyuic4>

5 Implementation

is used for solving the maximum weighted matching² problem.



Ant Name	Image	p	Coordinates	theta	area	Count
Arnold		1.5653	[416.87, 558.39]	90.079	322	1
Bob		1.8572	[785.25, 781.69]	153.37	252	19
Cenek		1.7113	[449.27, 307.25]	19.236	256	44
Dusan		1.8364	[480.17, 606.68]	176.54	239	10
Emil		1.7237	[901.68, 528.38]	98.770	204	60
Ferda		1.8899	[865.55, 426.93]	129.93	220	53
Gustav		0.6367	[582.28, 862.27]	80.708	133	37
Hugo		1.8127	[402.64, 745.97]	25.378	234	28

Figure 5.2 Visualization of each ant state in Ferda implementation.

²http://networkx.github.io/documentation/latest/reference/generated/networkx.algorithms.matching.max_weight_matching.html

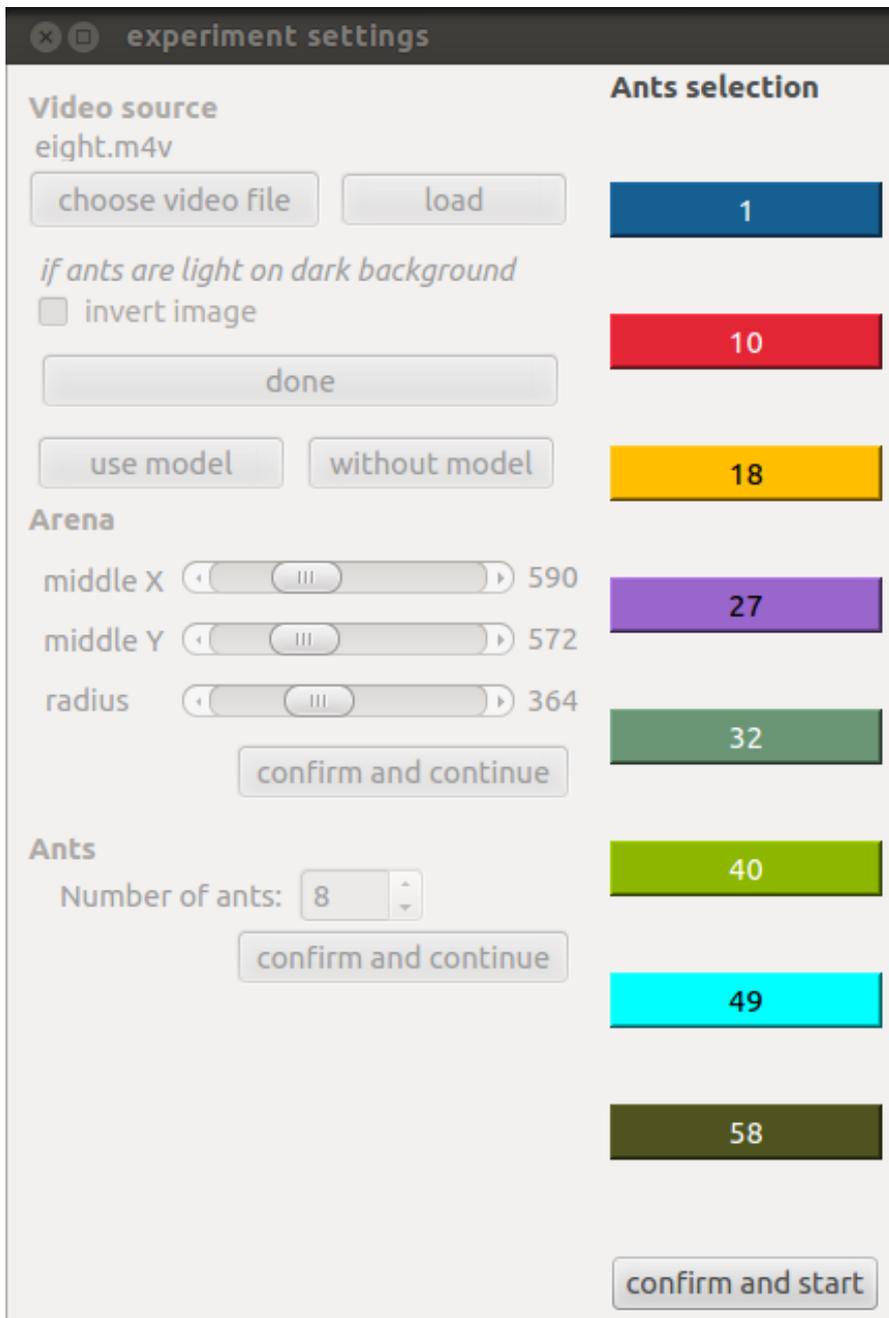


Figure 5.3 GUI for initialization in Ferda implementation.

6 Future work

A new method for multiple insect tracking has been developed. It is based on the MSER[1] region extraction. The assignment problem has been converted to the maximum weighted matching problem. The score/weight function for ant to region assignment has been introduced as well as the way how to build graph which is being searched to find the maximum weighted matching. As the third main part of the algorithm, a technique for fitting problem has been shown.

There are still few things to be improved and several issues to be solved:

- The fitting problem is operating with contour of an animal from the frame before the interaction starts. That is working good for animals like ants or flies where there are no big changes in body shapes but for animals like snakes, fishes etc. it fails whenever the shape is changed distinctly.
- For now it is using only information from previous frame to predict the movement. More complex movement model based on longer history could increase the assignment problem robustness, mainly in situations where animals are overlapping.
- The application should be extended by tools for trajectories repairing based on auto-analyzing the frames where certainty is low.
- The application should be optimized to be faster and the code should be refactored to make it more readable for other developers.

Bibliography

- [1] Matas, J., Chum, O., Urban, M., & Pajdla, T. (2004). *Robust wide-baseline stereo from maximally stable extremal regions*. Image and vision computing, 22(10), 761-767. 7, 27, 30
- [2] Galil, Z. (1986). *Efficient algorithms for finding maximum matching in graphs*. ACM Computing Surveys (CSUR), 18(1), 23-38. 10
- [3] Hu, M. K. (1962). *Visual pattern recognition by moment invariants*. Information Theory, IRE Transactions on, 8(2), 179-187. 7
- [4] Branson, K., Robie, A. A., Bender, J., Perona, P., & Dickinson, M. H. (2009). *High-throughput ethomics in large groups of Drosophila*. Nature methods, 6(6), 451-457. 3, 21
- [5] Mersch, D. P., Crespi, A., & Keller, L. (2013). *Tracking individuals shows spatial fidelity is a key regulator of ant social organization*. Science, 340(6136), 1090-1093. 3
- [6] Kimura, T., Ohashi, M., Crailsheim, K., Schmickl, T., Okada, R., Radspieler, G., & Ikeno, H. (2014). *Development of a New Method to Track Multiple Honey Bees with Complex Behaviors on a Flat Laboratory Arena*. PloS one, 9(1), e84656. 3
- [7] Psychoudakis, D., Moulder, W., Chen, C. C., Zhu, H., & Volakis, J. L. (2008). *A portable low-power harmonic radar system and conformal tag for insect tracking*. Antennas and Wireless Propagation Letters, IEEE, 7, 444-447. 3
- [8] Keni, B., & Rainer, S. (2008). *Evaluating multiple object tracking performance: the CLEAR MOT metrics*. EURASIP Journal on Image and Video Processing, 2008, 4, 22, 23
- [9] O'Neal, M. E., Landis, D. A., Rothwell, E., Kempel, L., & Reinhard, D. (2004). *Tracking insects with harmonic radar: a case study*. American Entomologist, 50(4), 212-218. 3
- [10] Straw, A. D., Branson, K., Neumann, T. R., & Dickinson, M. H. (2011). *Multi-camera real-time three-dimensional tracking of multiple flying animals*. Journal of The Royal Society Interface, 8(56), 395-409. 3
- [11] Hartigan, J. A., & Wong, M. A. (1979). *Algorithm AS 136: A k-means clustering algorithm*. Applied statistics, 100-108. 15

Bibliography

- [12] Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). *Maximum likelihood from incomplete data via the EM algorithm*. Journal of the Royal statistical Society, 39(1), 1-38. 15
- [13] Hansen, E., & Walster, G. W. (Eds.). (2003). *Global optimization using interval analysis: revised and expanded* (Vol. 264). CRC Press. 15
- [14] Sorkine, O. (2009). *Least-squares rigid motion using svd*. Technical notes, 120, 3. 16
- [15] M. Smid, <https://github.com/smids/clearmetrics> 22
- [16] <http://ist.ac.at/research/research-groups/cremer-group/> 3
- [17] Bradski, G., & Kaehler, A. (2008). *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media, Inc.. 27
- [18] Henry, M., Beguin, M., Requier, F., Rollin, O., Odoux, J. F., Aupinel, P., ... & Decourtye, A. (2012). *A common pesticide decreases foraging success and survival in honey bees*. Science, 336(6079), 348-350. 3
- [19] Schneider, C. W., Tautz, J., Grünewald, B., & Fuchs, S. (2012). *RFID tracking of sublethal effects of two neonicotinoid insecticides on the foraging behavior of *Apis mellifera**. PLoS One, 7(1), e30023. 3
- [20] <https://wiki.python.org/moin/PyQt> 27
- [21] <http://networkx.github.io/> 27
- [22] Ctrax application, <http://ctrax.sourceforge.net/> 3
- [23] K-Track application, <http://software.incf.org/software/the-tracking-software-k-track> 3
- [24] idTracker application, <http://www.idtracker.es/download> 3
- [25] M. Stumpe, AnTracks, <http://www.antracks.org/> 4
- [26] <http://www.york.ac.uk/news-and-events/features/ant-behaviour/>
- [27] Alfonso Pérez-Escudero, Julián Vicente-Page, Robert Hinz, Sara Arganda, Gonzalo G. de Polavieja. '*idTracker: Tracking individuals in a group by automatic identification of unmarked animals*' Nature Methods (in press, doi:10.1038/nmeth.2994) 3, 21, 23, 24

7 Appendix

Figure 7.1 Illustration from tracking on **ist-eight** sequence.

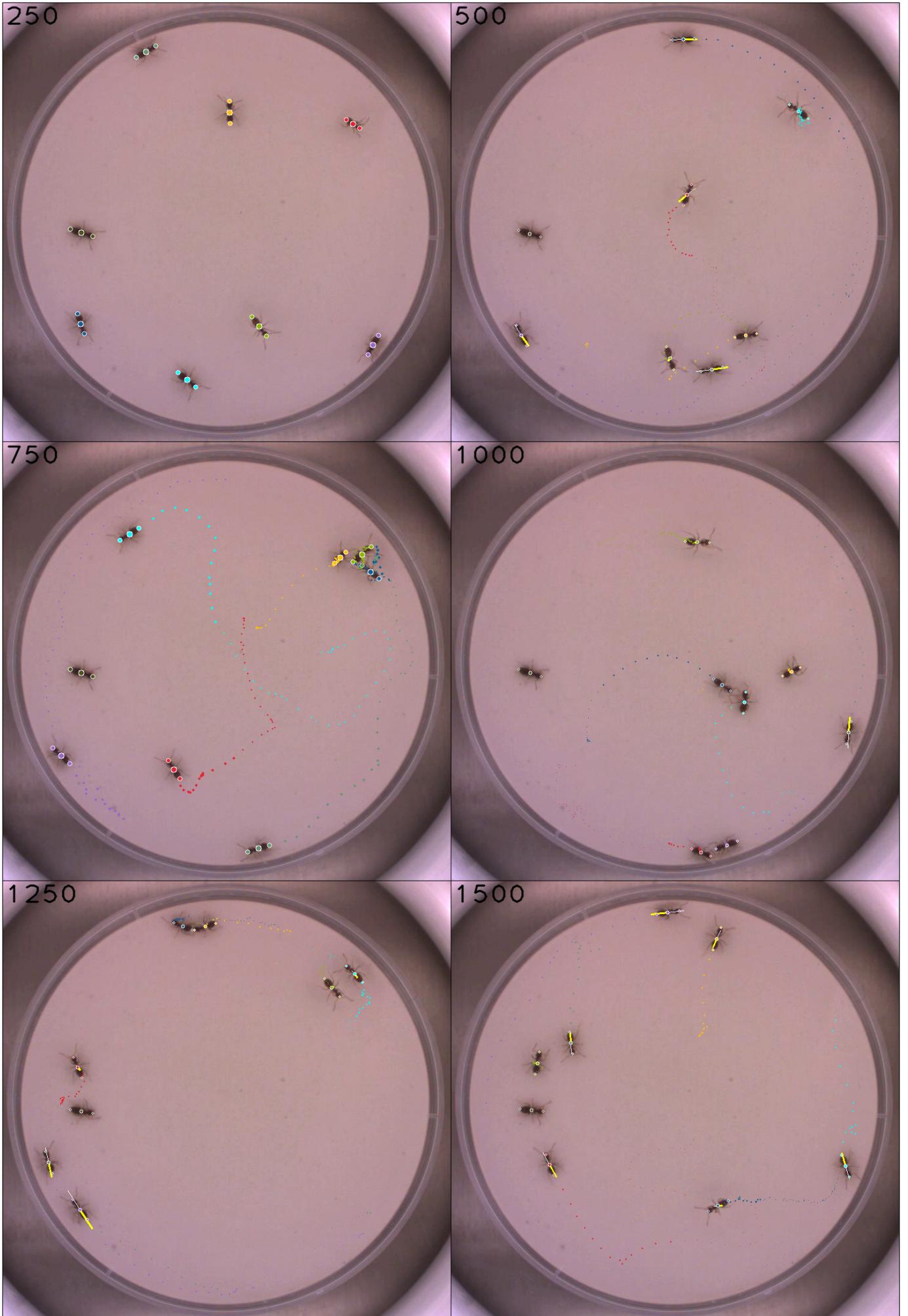


Figure 7.2 Illustration from tracking on **ist-fifteen** sequence.

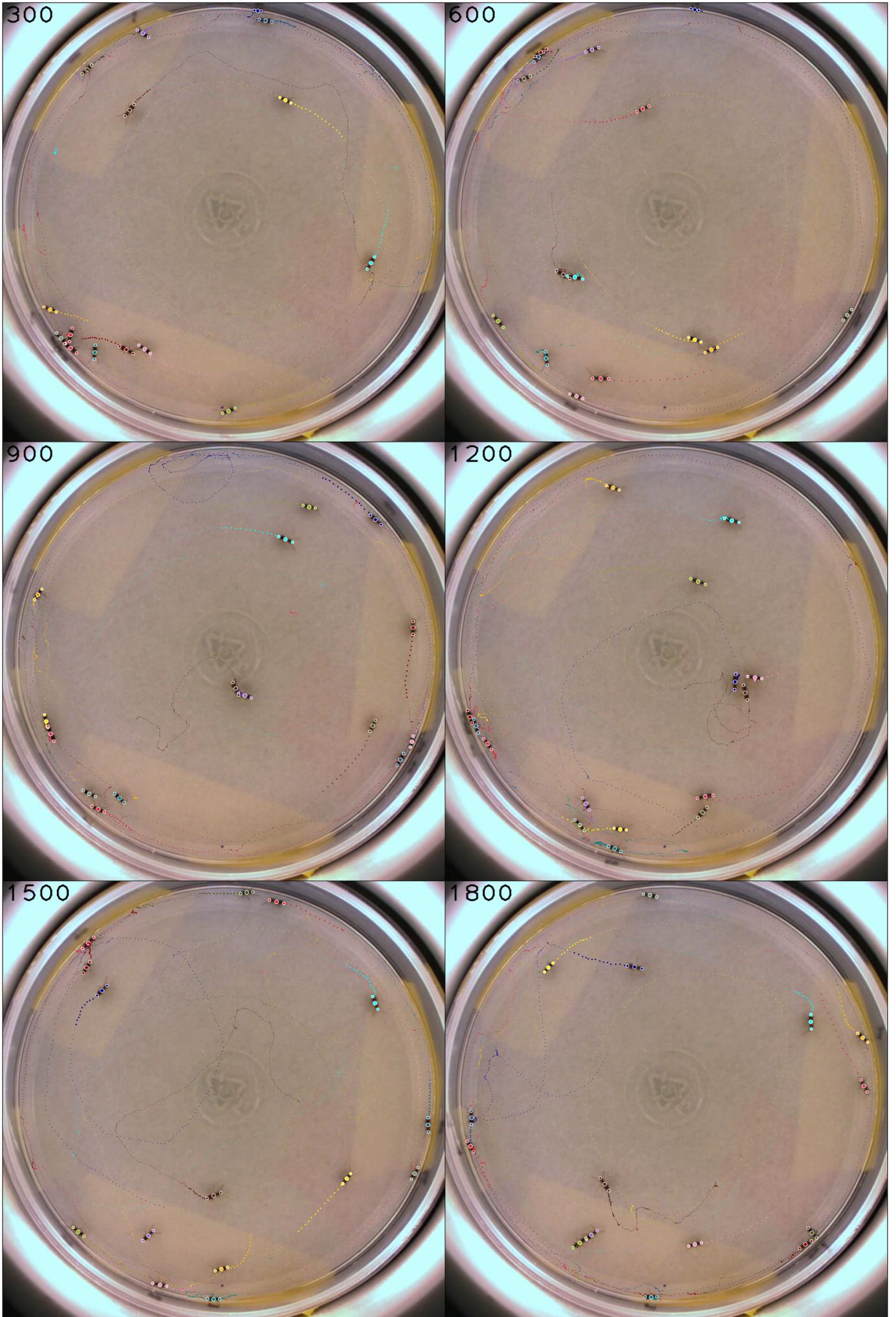


Figure 7.3 Illustration from tracking on **idTracker-Mesor-Structor** sequence.

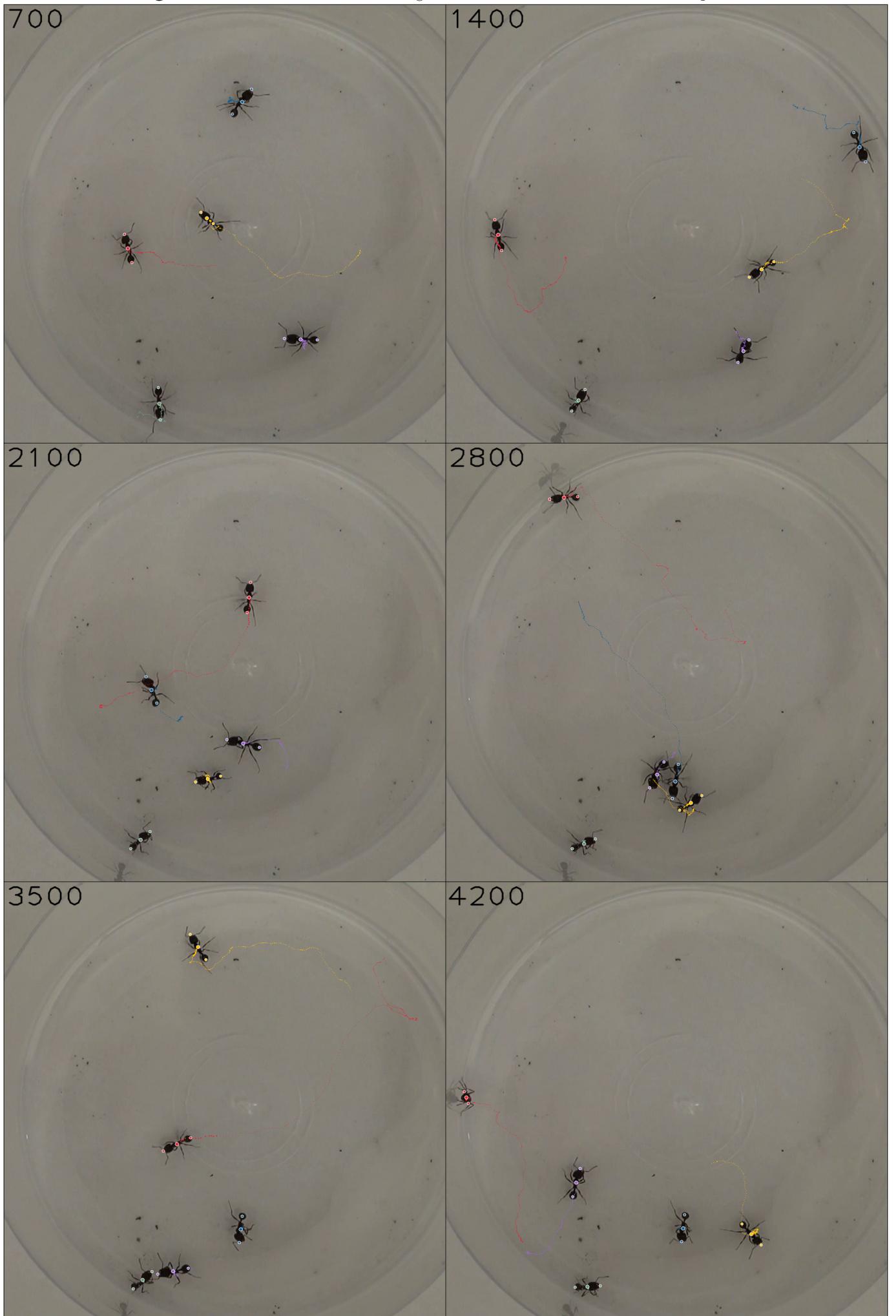


Figure 7.4 Illustration from tracking on idTracker-Zebrafish sequence.

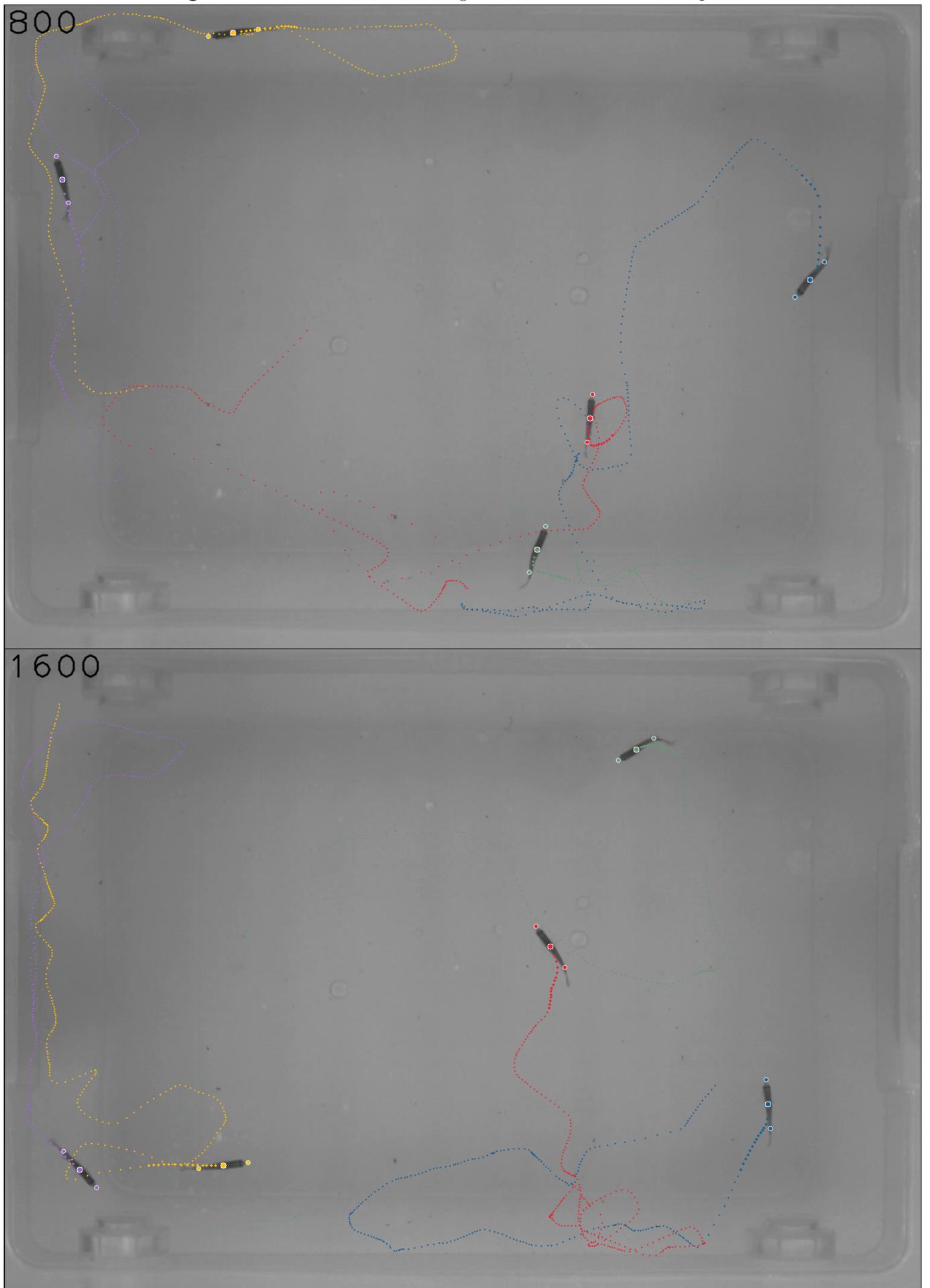


Figure 7.5 Illustration from tracking on *ctrax-Drosophyla* sequence.

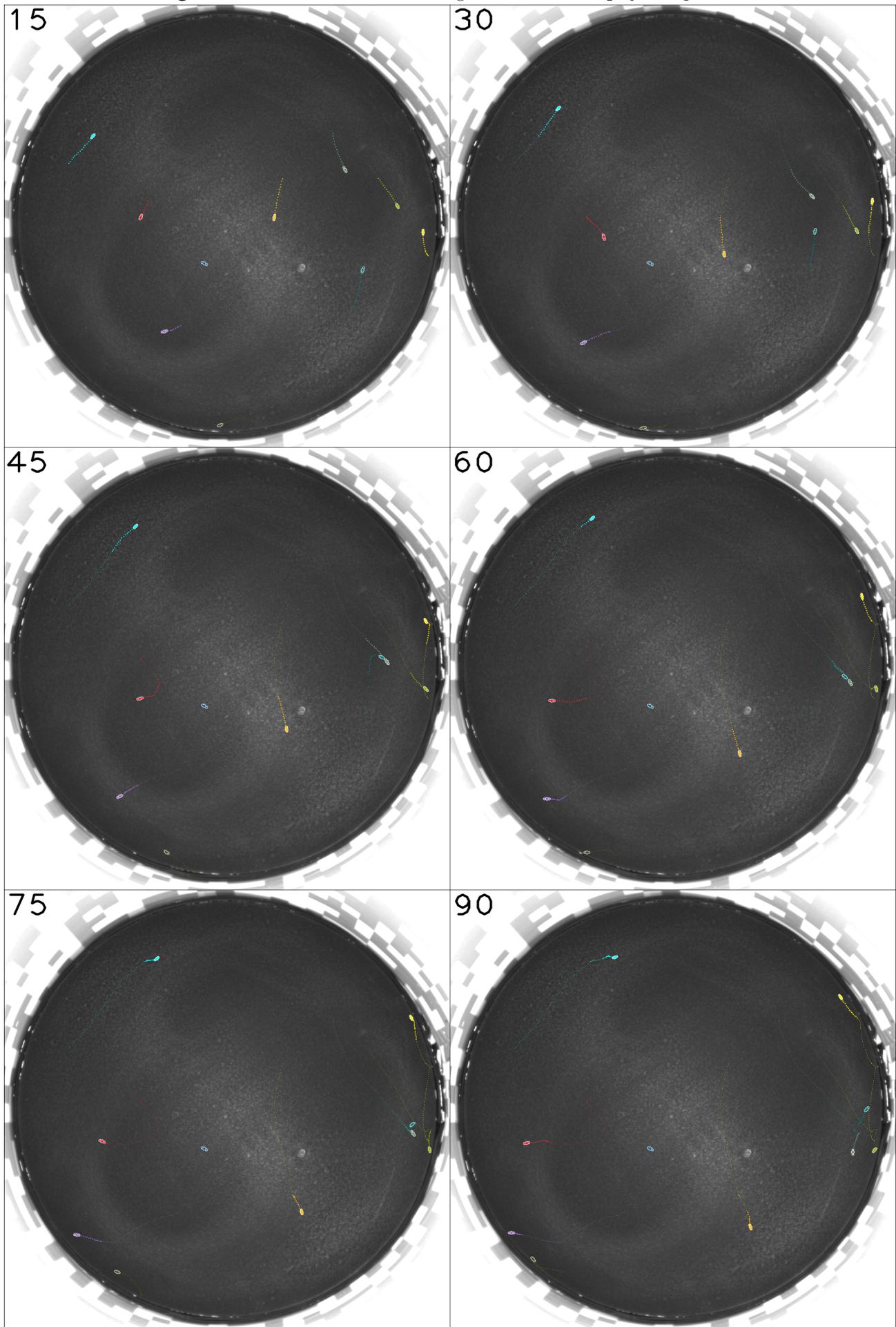


Figure 7.6 Mismatches, false positive and false negative errors placement in **ist-eight**. O = mismatch, x = FP, + = FN. Red dashed line shows the frame when K-Track crashed.

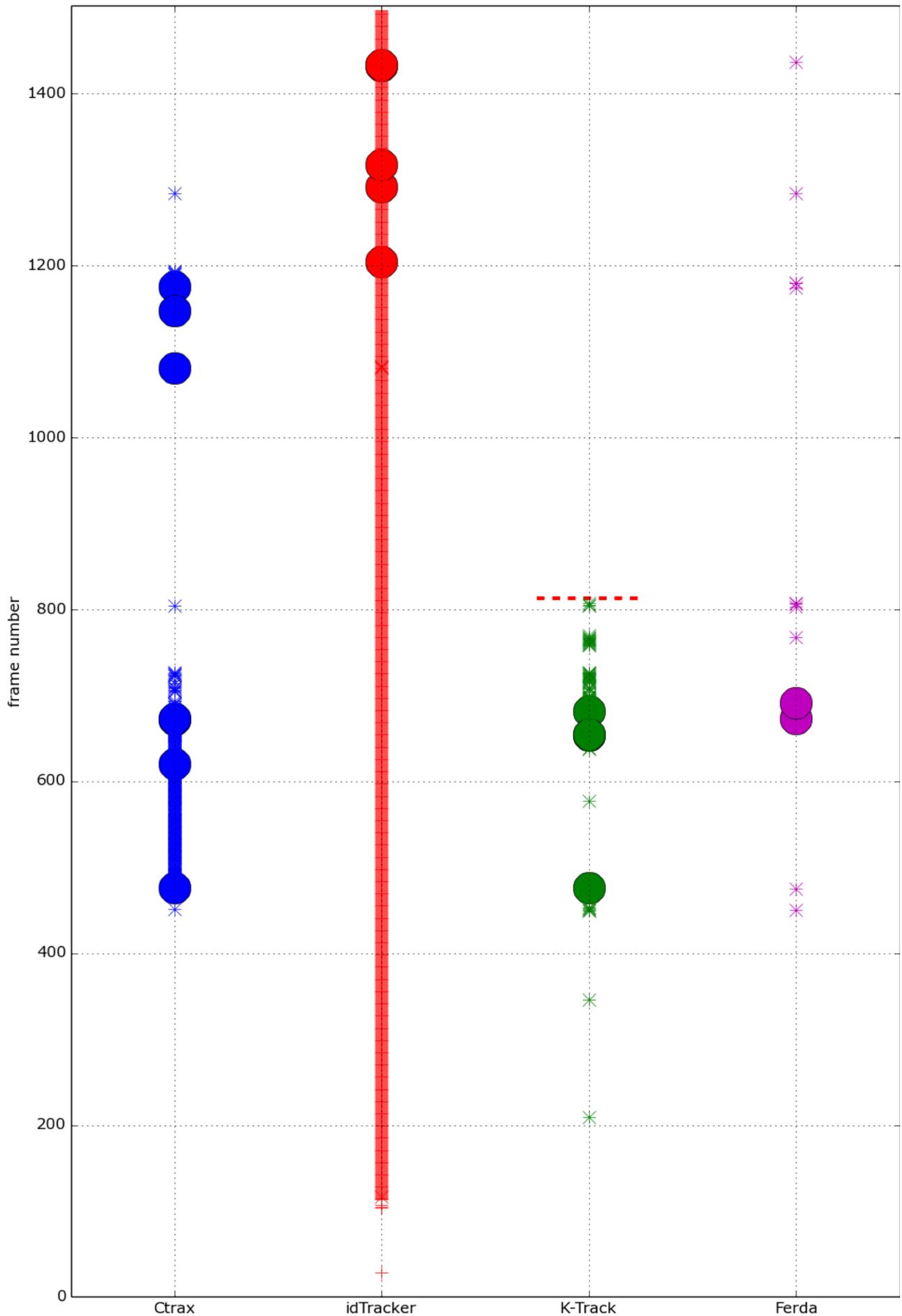


Figure 7.7 Mismatches, false positive and false negative errors placement in **ist-fifteen**. O = mismatch, x = FP, + = FN.

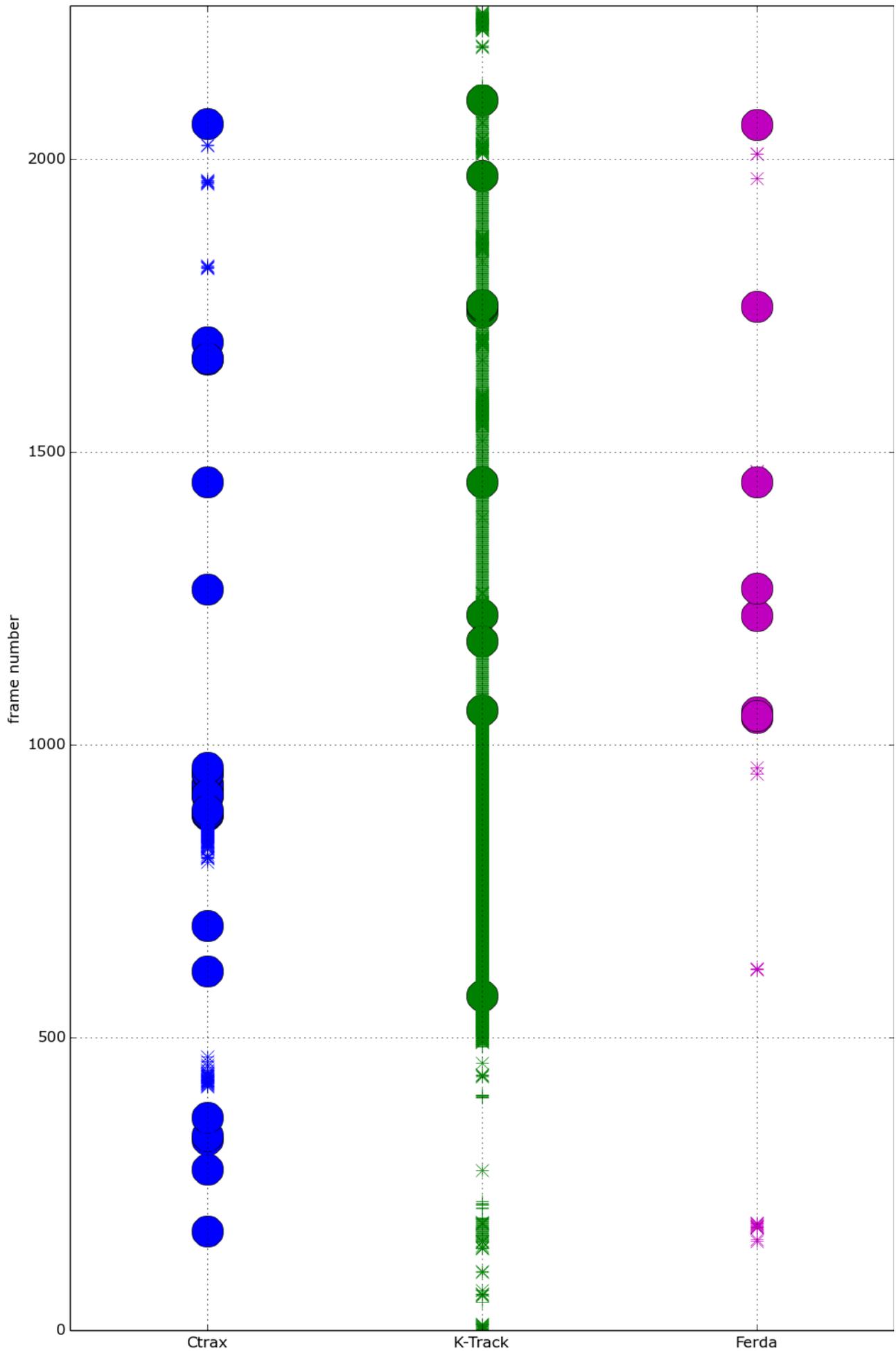


Figure 7.8 Evolution of certainty on *ist-eight* sequence. Dashed line shows mismatches positions

