Optimal estimation of broiler movement for commercial tracking

Henry Brunet, Pauline Creach, D Concordet

Smart Agricultural Technology 3 (2023) 100113

Contents lists available at ScienceDirect

Smart Agricultural Technology

journal homepage: www.journals.elsevier.com/smart-agricultural-technology

Optimal estimation of broiler movement for commercial tracking

Henry Brunet [a], Pauline Creach [b], Didier Concordet [a,*]

[a] INTHERES, Université de Toulouse, INRA, ENVT, 23 Chemin des Capelles, 31300, France
[b] Institut Technique de l'aviculture - ITAVI, Ploufragan 22440, France

ARTICLE INFO

Keywords:
Point tracking
Broilers welfare assessment
Displacement model

ABSTRACT

Nowadays, video tracking has taken a considerable part in monitoring systems. It allows identifying and follow every object in the camera field over time. While most of these algorithms are rather well suited to regular movements (following cars, pedestrians), they are often limited in more complex situations (high variations in speed, low detection rate, frequent shape variation). This paper proposes three methods adapted to broilers tracking in commercial environment. Past movements analysis of known broilers enable to estimate their motions and therefore to predict their new position. New unidentified broilers positions are then compared to these predicted positions. Distances between these two sets of positions are then used in the Hungarian Algorithm to assign an ID to new detected broilers regarding their past positions. Our methods differentiate by the way they predict the future positions. Contrary to most methods, they do not seek perfect regularity of movements and can deal with low rate detection. The proposed methods showed better performances than existing one. Tests have been made at 21, 26, and 37 days of age. At 21 days, our best method produces up to 35% fewer errors than a method with no estimation of movement. At 26 days of age, displacement distances can be set to only 68% of the maximum recorded displacement while improving an average of 21% of tracking errors across all methods.

1. Introduction

One of the most widely used techniques in the field of crowd surveillance remains video analysis. It is a non-invasive and inexpensive method for monitoring objects. Video tracking makes it possible to detect and follow each object in video to document their activities. There are today many tracking methods. These depend on the nature of the tracked objects, the time constraints, and the resources available for the calculations. We are interested in the tracking of several hundred of broilers. The natural framework is the so-called Multiple Object Tracking (MOT) which can deal with many objects compared to single object tracking. Video tracking consists in putting the same ID on the same object on all subsequent images. In his review, Luo et al. [1] distinguished the tracking methods according to the way the initialization, the processing steps and the pairing step are performed. There are mainly two ways to perform the initialization step: either detection of all objects is performed at each image (tracking by detection) or at each image one keeps the objects already present in the previous images. This last way to proceed characterizes the detection-free tracking methods. As it does not detect new objects entering the camera field, it is not suited to broilers tracking that are not enclosed in the camera field. In the tracking by detection approach [2], each object is limited by a bounding box which returns the position and size of the detected object. Detection requires searching objects in the whole image and it is generally applied to well-defined classes of objects. These methods guarantee a pretty good localization of detected objects. The effectiveness of MOT models largely depends on the detection model that precedes them. Nowadays, convolutional neural network detection methods achieve excellent detection rates, and some of them can be used for real-time analyses [3,4]. They are preferred to simple image processing methods like those summarized in Balaji and Karthikeyan [5], Kothiya and Mistree [6]. These methods are highly dependent on environmental conditions like illumination, background contrast, noise in the Image. Two processing modes can be considered; namely, the online [7,8] and the offline modes [9]. When using the online mode, the images are analyzed sequentially, one by one according to their recording time. To identify objects on an image, the tracking algorithm can only refer to past detected objects. The second mode is offline tracking. Even though it is still possible to identify objects sequentially, in offline mode, detection is realized over the whole image beforehand. It means that for identification, tracking algorithm has access to past identified objects and future detected objects. Where online mode is well

* Corresponding author.
E-mail addresses: henry.brunet@univ-tlse3.fr (H. Brunet), didier.concordet@envt.fr (D. Concordet).

https://doi.org/10.1016/j.atech.2022.100113
Received 16 July 2022; Received in revised form 29 August 2022; Accepted 31 August 2022
Available online 6 September 2022

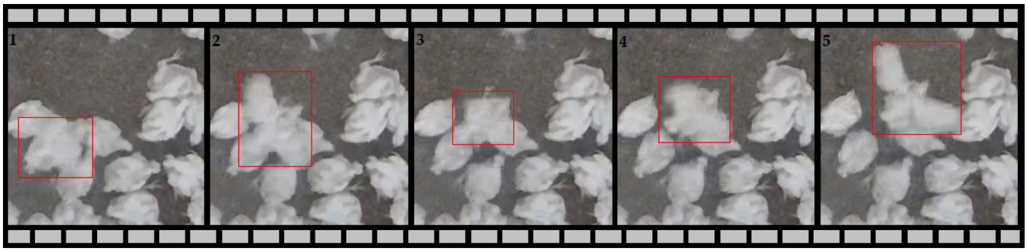

Fig. 1. Silhouette broiler evolution, 20 frames/second.

suited to real time process, offline mode has the advantage of having more detection data at its disposal, enabling thus to make better allocation decisions. In practice, offline mode does not deal with all data but only with a batch of frames. Despite its potential better performances, this mode leads to too long calculations, running out of memory, and a delay in the results output [1,10].

The two pairing modes are: Deterministic tracking and Stochastic tracking [11,12]. Deterministic tracking seeks for the best paring between new detected objects and previous identified objects. To doing so, a metric is chosen in order to favor specific displacements like constant motion. Hungarian algorithm [13] is still one of the main cited and used method for such an assignment problem. Greedy algorithms [12,14,15] are the mains substitutes to the Hungarian algorithm. Stochastic tracking assumes noises in data coming from the detection model. Therefore, it introduces uncertainty in the measures. These methods are often based on Kalman filter algorithm [16,17]. The main stochastic tracking algorithms are Network flow [18], Shortest path [19] and conditional random field [20], they are all suited for offline mode tracking. The two widely used methods for multi objects tracking are Joint Probability Data Association Filtering [21] and Multiple Hypothesis Tracking [22]. The Multi Target Tracking (MTT) try all possible associations between targets and tracked objects over time. This method leads to heavy computation. The Joint Probability Data Association Filtering (JPDAF) is less computationally expensive as it uses a filtering process based on the distance between target and tracked objects to reduce the number of potential association. The main drawback is that JPADF works only for a fix number of tracked objects and struggles to dissociate too closed objects. Main stochastic methods are not suited to unfixed and high number of objects to track.

While most of tracking algorithms are rather well suited to regular movements (following cars, pedestrians), they are often limited in more complex situations with high variations in speed, low detection rate and frequent shape variation. This article aims at developing a tracking method for broilers. This is why, we opt for the use of a detection based method that provides a solution to incoming and out-coming broilers in the camera field. To reduce the complexity of calculation due to the high amount of broilers in the camera field and therefore the high number of possible data association, we opt for an sequential processing (that has the advantage to work for both online and offline modes) and a deterministic pairing mode. This tracking method seeks to be independent to animals activity, to their spacing, to their weak movement regularity, to their occlusions etc. More precisely, we present three different ways to predict broilers positions.

The next section summarizes the works published on multi object tracking. The main algorithms used as benchmark to compare to our methods and the description of tree methods we suggest for broilers tracking are presented. Finally, the performances of the tracking methods suggested in this article are compared to the benchmark methods.

2. Related work

As reported by Yilmaz et al. [23], tracking algorithms can separated into three groups according to the information used to identify object in

Fig. 2. Example of input image.

each image. The main idea behind all these algorithms is to select specific object information that remains similar into successive frames for each object, or at least that is easily predictable among successive frames. kernel tracking algorithms use information like distribution of colors or object's texture included in the bounding box from previous detection. They are usually represented as histograms. The idea is to find object having the "closest" histogram [24] in successive frames. An example of distance measurement between two histograms is given by Pele and Werman [25]. For the texture analysis, Tuceryan [26] summarizes the different kinds of features we can find in the literature. At last, template tracking methods are also part of kernel tracking family. It's a brute method looking for correlations between small images delimited by bounding boxes [27]. These kinds of methods are really sensitive to illumination.

Silhouette tracking: While kernel tracking deals only with the information included in a fix shape (bounding box), silhouette tracking treats with object's shapes and contours. Shapes tracking seeks to match objects having closed silhouettes in two consecutive frames based on edge map image. If tracked objects are nonrigid, the silhouette has to be updated on each frame [28,29]. The contour can be seen as a surface that needs to be matched to previous surface features. In this way, Huttenlocher et al. [28] uses the Hausdorff distance in order to evaluate the distance between two surfaces. On the other hand, the contours can be represented as a closed outline. If so, one of the most suitable methods to deal with such feature is the Freeman chain code [30].

Point Tracking: These methods are usually used for small objects or objects with few details. There are deterministic methods [12] and probabilistic methods [31]. Deterministic methods define a cost according to the distance between two points (in two different frames). It could be a spatial distance, a difference between two speeds, or anything else that can be computed thanks to recorded positions over time. Then a combinatorial optimization algorithm is used to combine pairs of points in order to minimize the cost sum. The Hungarian algorithm [13] is the mainly used algorithm to solve such problem. However, greedy algorithms like those from [11] and [15] are also widely used.

In our application, we have few choices on the features to use. Our situation is to track several hundred chickens in a farm. Chickens are filmed from above by a camera set upright. All chickens in the video are the same age and have similar colors (Fig. 2). However, their shape that are quite similar when they are resting, can change very quickly as soon as they are moving (Fig. 1). We can then reasonably exclude the features based only on the visual part of the object, as they are weakly discriminative between chickens or are difficult to predict. As chickens have quite the same appearance, we choose the object's position as the main feature representing the chicken.

Tracking of objects summarized by their positions is really sensitive to miss detections. It can end up locally with no points (miss detection) or with one point for many objects (occlusion). One way to correct these problems is to assess object movement and therefore to predict its future positions as accurately as possible. Most of the methods deal with good detection, no entrance objects and straight movements. While an assumption like a straight movement [15] suit quite well for objects like cars, it becomes less usable for chickens due to their jerky movement.

2.1. Definitions

Let us first suppose that the system is closed, that is, all the objects remain in the field of the camera and no object comes out. In addition, we assume a perfect detection. In other words, all objects present are detected. If n_t denotes the number of detected broilers at time t, these assumptions ensure that $n_0 = \ldots = n_{t_{\max}} = N$. It should be noted that at time $t = 0$, there are only detections and arbitrarily ID affectations. The first ID association arises at time $t = 1$. Let's denote by Z_i^t the i object's position in $\mathscr{R}^2$ at time t. In our simplified situation (closed system), performing tracking amounts to build a sequence of $t_{\max}$ permutations $(\sigma^t)_{t \le t_{\max}}$ where for all $t \in 0, \ldots, t_{\max}$, $\sigma^t \in S_N$, so that for all $i \le n$ two successive positions of the sequence are "close". The permutation σ^t is the permutation to be applied to the objects of the image $t-1$ to obtain the number of the object on the image t. In the following, we note σ^{-t} the reciprocal permutation of σ^t such that $\sigma^{-t} \circ \sigma^t = \sigma^t \circ \sigma^{-t} = Id$, and $\sigma^{t,1}(i)$ is the contracted form of $\sigma^t \circ \ldots \circ \sigma^1(i)$

$$Z_i^0 \rightarrow Z^1_{\sigma^1(i)} \rightarrow \ldots \rightarrow Z^t_{\sigma^{t,1}(i)}$$

$$Z^0_{\sigma^{-1,-t}(j)} \leftarrow \ldots \leftarrow Z^{t-1}_{\sigma^{-t}(j)} \leftarrow Z_j^t$$

In the above equation, $i = \sigma^{-1,-t}(j)$ and $j = \sigma^{t,1}(i)$.

The velocity vector $\mathbf{V}^t_{i,k} = Z^t_k - Z^{t-1}_i$ allows to go from the point Z^{t-1}_i to Z^t_k is noted. When these points belong to the path of the same broiler like $Z^{t-1}_{\sigma^{-t}(i)}$ and Z^t_i this notation is simplified to $\mathbf{V}^t_i$ instead of $\mathbf{V}^t_{\sigma^{-t}(i),i}$. Its magnitude corresponds to the speed of the object. The angle between two successive velocity vectors $\mathbf{V}^{t-1}_{\sigma^{-t}(i)}$ and $\mathbf{V}^t_i$ is denoted α^t_i. By convention, objects at time t are the last tracked objects to which an identifier has been assigned. Those at time $t+1$ correspond to the newly detected objects, which must be assigned to the previously identified objects.

Point tracking may either operate data association of new detected points directly on previous recorded points or on estimated points. Point tracking operating on previous recorded is based on the notion of proximity (small position variation) and regular motion (small speed and direction variation). It defines a metric to measure the similarity between the velocity vectors $\mathbf{V}^t$ and the vectors $\mathbf{V}^{t+1}$. With a reasonable capture frequency, the object varies only very slightly in speed and orientation [11]. It may be well suited for tracking cars as they do not vary drastically of motion due to their inertia.

Point tracking operating on estimated points seek to predict the position of the points Z^{t+1} denotes $\widehat{Z}^{t+1}$, and the associated velocity vectors $\widehat{\mathbf{V}}^{t+1}$. In this case, the distance is measured between the vectors $\mathbf{V}^{t+1}$ and $\widehat{\mathbf{V}}^{t+1}$.

After measuring a potential match score between two objects, the tracking uses a combinatorial optimization algorithm to establish the best possible average correspondence between all the objects tracked at time t and all the objects detected at time $t+1$, based on the match scores.

2.2. Distance for data association

The reference method in this category is carried by Sethi and Jain [11]. They define standardized distances which favor the uniformity of directions ($d1^t_{i,j}$) and velocities ($d2^t_{i,j}$), between the vectors $\mathbf{V}^t_i$ and $\mathbf{V}^{t+1}_k$.

$$d1^t_{i,j} = 1 - \frac{\left\langle \mathbf{V}^t_i, \mathbf{V}^{t+1}_j \right\rangle}{\| \mathbf{V}^t_i \| \, \| \mathbf{V}^{t+1}_j \|} = 1 - cos\left(\mathbf{V}^t_i, \mathbf{V}^{t+1}_j\right) \tag{1}$$

The distance $d1_{i,j}$ (Eq. (1))is called directional coherence, it can be seen as the cosine of the angle between motion vectors $\mathbf{V}^t_i$ and $\mathbf{V}^{t+1}_j$. The distance is minimal when the two vectors are collinear and in the same direction. As the distance depends on the cosine of the angle, it does not discriminate easily two vectors with a low angle.

$$d2^t_{i,j} = 1 - \frac{2*\sqrt{\| \mathbf{V}^t_i \| \, \| \mathbf{V}^{t+1}_j \|}}{\| \mathbf{V}^t_i \| + \| \mathbf{V}^{t+1}_j \|} = \frac{\left(\sqrt{\| \mathbf{V}^t_i \|} - \sqrt{\| \mathbf{V}^{t+1}_j \|}\right)^2}{\| \mathbf{V}^t_i \| + \| \mathbf{V}^{t+1}_j \|} \tag{2}$$

The distance $d2_{i,j}$ (Eq. (2)) is called speed coherence. This term measures the variation of magnitude between the two vectors $\mathbf{V}^t_i$ and $\mathbf{V}^{t+1}_j$. This distance is null when the two vectors magnitudes are equal and it increases as the magnitude differ from each other (Fig. 4). The maximum value is 1. It happens either if $\| \mathbf{V}^{t+1}_j \|$ or $\| \mathbf{V}^t_i \|$ is null, or if $\| \mathbf{V}^{t+1}_j \|$ or $\| \mathbf{V}^t_i \|$ is infinite.

This speed coherence does not imply that the difference between the two speeds has to be minimal. If a broiler progress at low speed, this speed coherence might prefer to associate this broiler to far away chicken in the next frame than with itself. This situation appears in particular when broilers stops or get start. Sethi and Jain then define $\Delta = w1*d1^t_{i,j} + w2*d2^t_{i,j}$ which is a combination of two distances. The weights w1 and w2 are chosen between 0 and 1, such that $w1 + w2 = 1$.

A second much more classical method is carried by Rangarajan [15]. Its distance is made up of two terms. The first one measures the difference between the velocity vectors formed at times t and $t+1$. Like the Sethi and Jain method, it assumes that speed and direction vary slightly between two consecutive frames. The second one corresponds to speed intensities at time $t+1$. It favors small displacements (low speeds) between two consecutive frames.

$$d3^t_{i,j} = \frac{\| \mathbf{V}^t_i - \mathbf{V}^{t+1}_{i,j} \|}{\sum_{k=1}^{N^t} \sum_{h=1}^{N^{t+1}} \| \mathbf{V}^t_k - \mathbf{V}^{t+1}_{k,h} \|} + \frac{\| \mathbf{V}^{t+1}_{i,j} \|}{\sum_{k=1}^{N^t} \sum_{h=1}^{N^{t+1}} \| \mathbf{V}^{t+1}_{k,h} \|} \tag{3}$$

A critical point of these methods comes from the fact that they only depend on the last object's movement. The two main drawbacks are: some movements may require more than one registered movement due to a non straight motion, and a single detection error quickly makes this method unusable. Actually, dealing with more registered points helps to smooth the movement and so to keep direction movement information whereas few past points have been lost.

Rather than measuring the distance between two positions in two successive frames, it could be more advisable to measure this distance with predicted positions computed with more past frames. By using our knowledge on past positions, we can estimate the next objects positions and then we can deal with shortest distances. Karunasekera [32] defines a normalized displacement estimation. The displacement is equal to 0 if the difference between the predicted position and the estimated position is null. The displacement is equal to 1 if this difference is greater or equal to the object's size. The object's size is written n_{dst}. It is deduced from the dimensions of the region circumscribed to the object, resulting from the detection. It implies that between two images, the object's displacement is less than its size. The definition of the distance between the position of the object at Z_i^{t+1} and its estimated position $\widehat{Z}_i^{t+1}$ is written:

$$d4_i^t = \min\left(1, \frac{\| Z_i^{t+1} - \widehat{Z}_i^{t+1} \|}{n_{dst}}\right) \tag{4}$$

The estimation of $\widehat{Z}_i^{t+1}$ is based on the following model:

$$\mathbf{V}_{\sigma^{t+1}(i)}^{t+1} = \gamma_0 \mathbf{V}_i^t + \sum_{s=1}^{P} \gamma_s \mathbf{V}_{\sigma^{-(t-s+1)},\ldots\sigma^{-t}(i)}^{t-s} + \epsilon_i^t \tag{5}$$

where $\epsilon_i \sim \mathcal{N}(0, \Sigma)$

Unfortunately, the way the author estimated the α_S coefficients in Eq. (5) is not developed in Karunasekera et al. [32]. He uses integer values with a higher weight for most recent velocity vectors. The new position estimation is written as follow:

$$\widehat{Z}_i^{t+1} = Z_i^t + \widehat{\mathbf{V}}_i^{t+1} \tag{6}$$

The new velocity vector $\widehat{\mathbf{V}}_i^{t+1}$ modeling is defined as:

$$\widehat{\mathbf{V}}_i^{t+1} = \frac{\sum_{a=1}^{P} a \mathbf{V}_{\sigma^{-(t-a)},\ldots\sigma^{-t}(i)}^{t-a+1}}{M(M+1)/2} \tag{7}$$

P is set to limit the number of past velocity vectors to use. It assumes that only recent velocity vectors are relevant to the new position estimation. According to the author, the optimal value of M is 5, as it is: "*large enough to get good average for predictions and small enough not to drift because of old information*" (ref: [32]). We could regret a lack of information about the way to set the different parameters: the parameter M as it may depend on many factors (number of frame per second, etc. ...), and the coefficients α_i as it limits the model of the Eq. (5) to a particular case. We will see later a more global approach.

Fletcher [33] defines a method whose parameters are set by the method of Recursive Least Squares. The model can be expressed as follow:

$$Z_i^t = \Psi^T \mathbf{x}_i(t) + \epsilon_i^t \tag{8}$$

where

- $\epsilon \sim \mathcal{N}(0, \Sigma)$
- $\mathbf{x}(t) = (Z_{\sigma^{-t}(i)}^{t-1}, \ldots, Z_{\sigma^{-t+p-1}\circ\ldots\circ\sigma^{-t}(i)}^{t-p})$ is the set of past broiler positions.
- Ψ^T the $\mathscr{R}^p$ parameters vector to be estimated

Z_i^t is then a linear combination of past broilers positions. The real value of Ψ is unknown. The method seeks to estimate $\widehat{\Psi}$ which is a estimator of Ψ. The vector of parameters $\widehat{\Psi}$ is recursively updated based on previous estimation errors.

The error of estimation is noted:

$$e(t) = Z_i^t - \widehat{\Psi}(t)^T \mathbf{x}(t) \tag{9}$$

where $\widehat{\Psi}(t)$ is estimated by:

$$\widehat{\Psi}(t) = \widehat{\Psi}(t-1) + \mathbf{K}(t) e(t) \tag{10}$$

The vector $\mathbf{K}(t)$ is a set of coefficients that manages the speed of convergence of $\widehat{\Psi}(t)$ to Ψ. Having small coefficients implies a slow response to update parameters $\widehat{\Psi}(t)$ but it is less sensitive to noise. We can find in Godfrey and Jones [34], Bozic [35] different ways of fixing $\mathbf{K}(t)$ based on the error covariance matrix.

This method has the advantage to set automatically its parameters unlike the Karanusekara one. On the other hand, this method measures positions and not movements, which makes the estimation position-dependent. The further these positions are from the origin of the image, the greater the variation of the estimated point. To get around the problem, it is judicious to replace positions by velocity vectors. Velocity vectors do not vary as they get away from the origin of the image.

3. Prediction models

The method we propose relies on estimating several hundred chickens' movements to predict their future positions. The peculiarity here comes from the fact that, unlike a car or a pedestrian, a chicken does not move at a constant speed with a well-defined direction. We therefore seek to determine an estimator $\widehat{\mathbf{V}}_{\sigma^{t+1}(i)}^{t+1}$ of $\mathbf{V}_{\sigma^{t+1}(i)}^{t+1}$ such that $\epsilon_i = \| \widehat{\mathbf{V}}_{\sigma^{t+1}(i)}^{t+1} - \mathbf{V}_{\sigma^{t+1}(i)}^{T+1} \|$ has the smallest variance. Since hundreds of chickens may create many occlusions, the movement predictor has to deal with low detection rate. Finally, as the model of detection may not be perfect, the method has to deal with motion due to detection imprecision. As the chickens position may vary slightly around their actual positions, the movement estimation method has to take it into account to determine the natural movement of chickens.

Given what has already been done, the new method must meet specific requirements.

- It must be adapted to the type of object followed to be as precise as possible.
- This method must be independent of the object's activity. The parameters must therefore be robust whatever the activity is.
- The method must be independent of the displacement orientation. Rotating the camera should not influence the results.
- The coefficients used should ideally be known from the start of the tracking to manage the incoming objects more easily and limit the calculation times during the tracking.

3.1. Methodology

In order to estimate the parameters required by the different methods, we dispose of several tracking data sets of reference whose points position have been manually identified. The results have been checked experimentally. When an object seems to disappear from the video due to occlusions or poor detection, its velocity vector was set to null vector as we did not have any information about the new displacement.

3.1.1. The models

First model The first model consists in considering the movement of the animal as being null.

$$\mathbf{V}_{\sigma^{t+1}(i)}^{t+1} = \epsilon_i^t \tag{11}$$

With: $\epsilon_i^t \sim \mathcal{N}(0, \Sigma)$. This model can be considered as a reference one. The matrix ϵ is diagonal, and its terms are inversely proportional to the squares of the pixel area.

Second model The second model is based on the notion of constant motion from Sethi and Jain (Section 2.2). It seeks for two same movements in two consecutive frames.

$$\mathbf{V}_{\sigma^{t+1}(i)}^{t+1} = \mathbf{V}_i^t + \epsilon_i^t \tag{12}$$

With: $\epsilon_i^t \sim \mathcal{N}(0, \Sigma)$.

This model is well suited when the frame rate is high and motion varies slowly between two consecutive frames.

Third model This model is the Hasith Karunasekera [32] one.

$$\mathbf{V}_{\sigma^{t+1}(i)}^{t+1} = \gamma_0 \mathbf{V}_i^t + \sum_{s=1}^{P} \gamma_s \mathbf{V}_{\sigma^{-(t-s+1)},\ldots\sigma^{-t}(i)}^{t-s} + \epsilon_i^t \tag{13}$$

where $\epsilon_i^t \sim \mathcal{N}(0, \Sigma)$ and $\gamma_s \in \mathbb{R}$, with $0 \leq \gamma_s < 1$

3.1.2. Motion prediction

Based on these three models, it is possible to define different motion predictions.

Nearest neighbor This method predicts movement based on the model (12). There are no parameters to estimate.

$$\widehat{\mathbf{V}}^{t+1}_{\sigma^{t+1}(i)} = 0 \tag{14}$$

The estimated position $\widehat{Z}^{t+1}_{\sigma^{t+1}(i)}$ is therefore in a region centered in Z^t_i and whose radius depends on the variance of $\mathbf{V}^{t+1}_{\sigma^{t+1}(i)}$.

Regular motion The movement prediction is based on the model (15). It does not need any parameters to set either.

$$\widehat{Z}^{t+1}_{\sigma^{t+1}(i)} = Z^t_i + \widehat{\mathbf{V}}^{t+1}_{\sigma^{t+1}(i)} \tag{15}$$

In such situation the predicted position is estimated as:

$$\widehat{\mathbf{V}}^{t+1}_{\sigma^{t+1}(i)} = \mathbf{V}^t_i \tag{16}$$

Weighted average This one is based on the model (13). The parameters are set as in the same they are presented in the article [32] where the model comes from. P is fixed to 5 and $\gamma_s \in 1,2,3,4,5$.

$$\widehat{\mathbf{V}}^{t+1}_{\sigma^{t+1}(i)} = \frac{1}{3}\mathbf{V}^t_i + \frac{1}{15}\sum_{s=1}^{4}(5-s)\mathbf{V}^{t-s}_{\sigma^{-(t-s+1),-t}(i)} \tag{17}$$

Cartesian least squares It is the same model as the Karunasekera one (13), but differs by the way of estimating the model parameters. We estimate the parameters M and γ_s, based on reference data sets. This time the estimated position is written as:

$$\widehat{Z}^{t+1}_{\sigma^{t+1}(i)} = Z^t_i + \sum_{s=0}^{t-1}\widehat{\gamma}_s\mathbf{V}^{t-s}_{\sigma^{-(t-s+1),-t}(i)} \tag{18}$$

where $\widehat{\gamma}_s$ is estimated by the least squares method. Proceeding this way, the method becomes adapted to the type of object being tracked, and the coefficients are determined upstream to work with the most accurate coefficients possible from the start of tracking. Since several past points are used, this makes it possible to smooth the last displacements and be less sensitive to detection imprecision and miss detections. V^t_i has a Vx^t_i and a Vy^t_i components. To ensure a common γ_s to both of the components, V^t_i is then express in the complex domain. $V^t_i = Vx^t_i + iVy^t_i$. The estimation problem is reduced to:

$$\mathbf{V}^t_i = \sum_{s=1}^{t-1}\gamma_s\left(\mathbf{V}_{X^{t-s}_{\sigma^{-(t-s+1),-t}(i)}} + i\mathbf{V}_{Y^{t-s}_{\sigma^{-(t-s+1),-t}(i)}}\right) + \epsilon^t_i \tag{19}$$

where $\epsilon \sim \mathcal{N}(0,\Sigma)$ The least squares solution for complex values is given

$$A = B + iC = \begin{pmatrix} \mathbf{V}_{X^{t-1}_{\sigma^{-t}(1)}} & \cdots & \mathbf{V}_{X^{t-n}_{\sigma^{-(t-n+1),-t}(1)}} \\ \vdots & & \vdots \\ \mathbf{V}_{X^{t-1}_{\sigma^{-t}(m)}} & \cdots & \mathbf{V}_{X^{t-n}_{\sigma^{-(t-n+1),-t}(m)}} \end{pmatrix} + i\begin{pmatrix} \mathbf{V}_{Y^{t-1}_{\sigma^{-t}(1)}} & \cdots & \mathbf{V}_{Y^{t-n}_{\sigma^{-(t-n+1),-t}(1)}} \\ \vdots & & \vdots \\ \mathbf{V}_{Y^{t-1}_{\sigma^{-t}(m)}} & \cdots & \mathbf{V}_{Y^{t-n}_{\sigma^{-(t-n+1),-t}(m)}} \end{pmatrix}$$

by Turetsky [36]. We note $A = B + iC \in \mathcal{R}^{m,n}$ the input matrix representing past movement values.

$b = u + iv \in \mathcal{R}^m$ is the output representing current movement values.

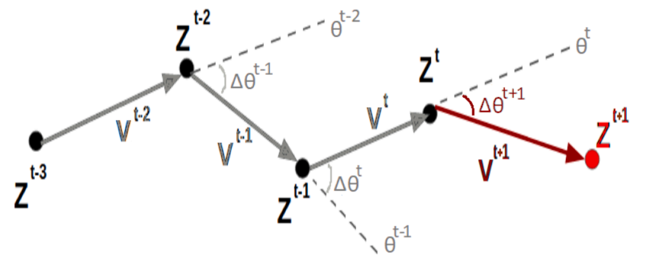

Fig. 3. nomenclature.

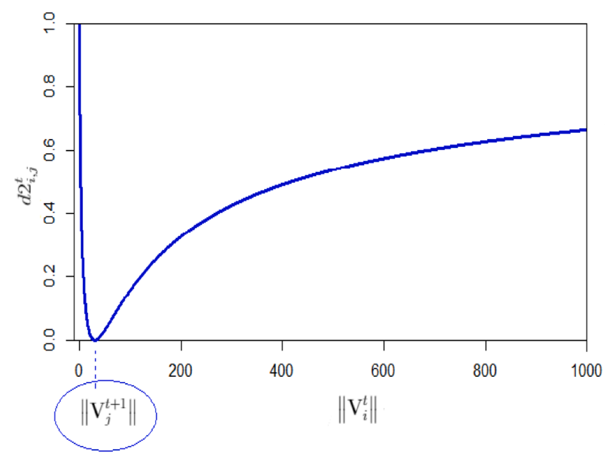

Fig. 4. Curve corresponding to the speed coherence equation for a fixed value of $\|\mathbf{V}^{t+1}_j\|$. Here $\|\mathbf{V}^{t+1}_j\| = 30$.

$$b = u + iv = \begin{pmatrix} \mathbf{V}_{X^t_1} \\ \vdots \\ \mathbf{V}_{X^t_m} \end{pmatrix} + i\begin{pmatrix} \mathbf{V}_{Y^t_1} \\ \vdots \\ \mathbf{V}_{Y^t_m} \end{pmatrix}$$

$z = x + iy \in \mathcal{R}^n$ is the vector of parameters to estimate.

$$z = x + iy = Re\begin{pmatrix} \gamma_1 \\ \vdots \\ \gamma_n \end{pmatrix} + iIm\begin{pmatrix} \gamma_1 \\ \vdots \\ \gamma_n \end{pmatrix}$$

The solution of γ is written under the form:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} BB^T + CC^T & -C^TB + B^TC \\ C^TB - B^TC & BB^T + CC^T \end{pmatrix}^{-1}\begin{pmatrix} B^T & C^T \\ C^T & B^T \end{pmatrix}\begin{pmatrix} u \\ v \end{pmatrix} \tag{20}$$

Polar least squares This method is based on the model (13). It is the equivalent of the method 3.1.2.4 but in polar coordinates, with $\mathbf{V}^t = (R^t, \theta^t)$. An angle may represent either a direction or a rotation. While it makes sense to sum rotation angles (two rotations of $\pi/4$ is equivalent to one rotation of $\pi/2$), the summation of $\pi/2$ direction with a $-\pi/2$ direction has no physical interpretation. This is why it is better to deal with variation of orientation (rotation) referred as $\Delta\Theta$ in Fig. 3 and not directly with velocity vector directions Θ. Dealing with polar

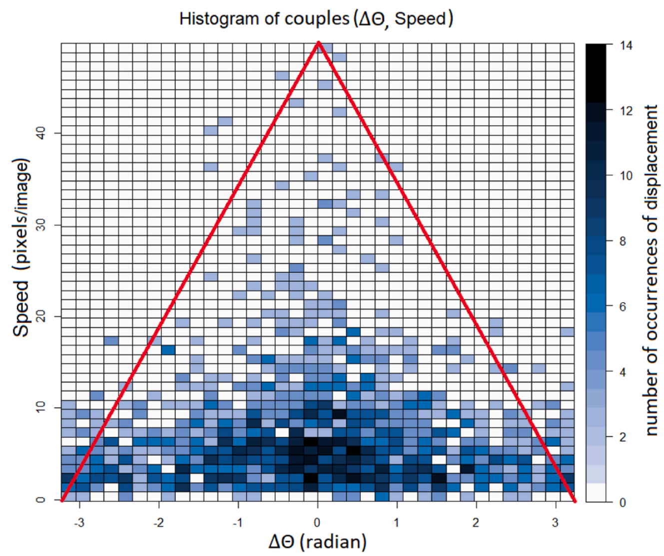

Fig. 5. Relation between the speed and the variation of orientation.

coordinates faces to constraints on speeds. In this sense, speeds values R are forced to be positives, as variations of orientation are includes in $[-\pi, \pi)$.

To ensure a positive speed value, the analysis uses the logarithm of the speed value.

$$\begin{pmatrix} \log(R_1^t) \\ \vdots \\ \log(R_m^t) \end{pmatrix} = \begin{pmatrix} \log(R_1^{t-1}) & \cdots & \log(R_1^{t-n}) \\ \vdots & & \vdots \\ \log(R_1^{t-1}) & \cdots & \log(R_1^{t-n}) \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{pmatrix}$$

The motion is then estimated as:

$$R_i^t = \prod_{s=1}^{t-1} \left(R^{t-s}_{\sigma^{-(t-s+1),-t}(i)} \right)^{\alpha_s} \tag{21}$$

$$\Delta\Theta_i^t = \sum_{s=1}^{t-1} \beta_s \Delta\Theta^{t-s}_{\sigma^{-(t-s+1),-t}(i)} \tag{22}$$

$$\Delta\Theta_i^{t-1} = \Theta_i^t - \Theta_i^{t-1} \tag{23}$$

$$\widehat{\mathbf{V}}^{t+1}_{\sigma^{t+1}(i)} = \widehat{R}^{t+1}_{\sigma^{t+1}(i)} \begin{pmatrix} cos\left(\Theta_i^t + \Delta\widehat{\Theta}^{t+1}_{\sigma^{t+1}(i)}\right) \\ sin\left(\Theta_i^t + \Delta\widehat{\Theta}^{t+1}_{\sigma^{t+1}(i)}\right) \end{pmatrix} \tag{24}$$

Parameters in (21) and (22) are also estimated with least squares method on the reference database.

As said previously, a non detected object would be considered to be at the same position as its last measure, since its motion is set as null. It means, a null speed but also an undetermined orientation of motion. Because *undetermined* is a number difficult to handle with, we arbitrary set it to zero.

Least squares under constraints In the previous motion estimation there is no dependence between speed values and variations of orientation. Thanks to the tracking database created manually (all objects are detected, there are no false positive and no tracking error), it is possible to check the relationship between the speed and the variation of orientation. Variations of orientation are centered around zero. For each

Table 1
Specifications of each video. The first row is the camera field area. The second row is the mean number of broilers over each image of the video. The third row is the broilers activity intensity. The fourth row is the maximal recorded speed in the video. The fifth row is the minimal distance recorded between the closest broilers. The sixth row is the broilers largest size median. The seventh row is percentage of detected boilers out of the CNN. The last row is the percentage of false positive out the CNN.

	21 days of age	26 days of age	37 days of age
Surface (m^2)	3.4	3.8	3.8
Mean number of broilers	34	64	38
Activity	high	low	low
Speed (pixel s^{-1})	826	630	544
Distance intra broilers (pixel)	36.9	32.5	34.4
Broilers size	106	102	141
Sensitivity	97	99	99
Precision	0.2	2.28	1.6

Fig. 6. 21 days of age.

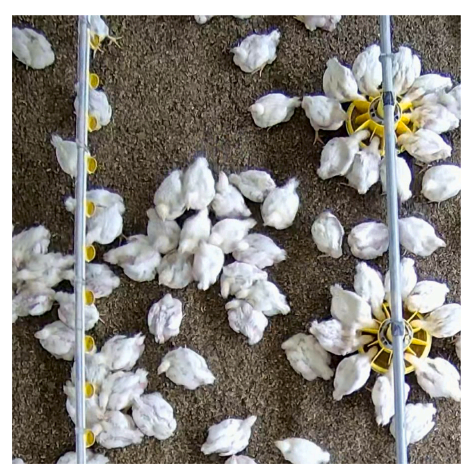

Fig. 7. 26 days of age.

Fig. 8. 37 days of age.

cell of the 2 dimensional histogram in the Fig. 5, the intensity represents the occurrence of couples speed and orientation variation. It can be noticed that for small speed the range of orientation variation is large. This range decreases as speed increases. It means that a displacement of 40 pixels/image paired with a π orientation variation is considered as an outlier. Based on the maximum recorded speed, two red lines are drawn as a first order approximation of the limit of the distribution. The idea is not to model exactly the relationship between speeds and variations of orientation, but to limit outliers.

High speed values are therefore accepted when the expected orientation variation is small. On the other hand, if the speed estimator is high, constraints are applied on variation orientation in order to limit outliers. This approximation of the distribution limit put constraints to the speed and to the orientation variation estimation of the model (12).

Once $\widehat{R}_i^t$ and $\Delta\widehat{\Theta}_i^t$ are estimated they are used to constrain new estimators $\widehat{R'}_i^t$ and $\Delta\widehat{\Theta'}_i^t$

$$\widehat{R}'^t_i = \widehat{R}_i^t\left(1 - abs\left(\frac{\Delta\widehat{\Theta}_i^t}{\pi}\right)\right) \tag{25}$$

$$\Delta\widehat{\Theta'}_i^t = \min\left(\Delta\widehat{\Theta}_i^t, \pi\left(1 - \frac{\widehat{R}_i^t}{\max(R)}\right)\right) \tag{26}$$

The value $\max(R)$ is measured from the data base of reference.

The estimation is then written as:

$$\widehat{Z}^{t+1}_{\sigma^{t+1}(i)} = Z_i^t + \widehat{\mathbf{V}}^{t+1}_{\sigma^{t+1}(i)} \tag{27}$$

$$\widehat{\mathbf{V}}^{t+1}_{\sigma^{t+1}(i)} = \widehat{R}'^{t+1}_{\sigma^{t+1}(i)}\begin{pmatrix} cos\left(\Theta_i^t + \Delta\widehat{\Theta'}^{t+1}_{\sigma^{t+1}(i)}\right) \\ sin\left(\Theta_i^t + \Delta\widehat{\Theta'}^{t+1}_{\sigma^{t+1}(i)}\right)\end{pmatrix} \tag{28}$$

3.2. Videos and detections

Three videos where used to estimate the models parameters.

When broilers grow up, the rate of detected broilers over total number of broilers increases. Here, the activity is referred as the global motion of broilers. The more broilers are moving with high speeds the higher is the activity. High speed motion is generally accompanied by flapping wings. A false positive is detection recorded where no broiler is present. The minimal recorded distance between broilers is a threshold below which two broilers positions can not be recorded.

Finally, for each broiler is given its bounding box size. The last row of the Table 1 gives the median of the larger bounding box size of all broilers. The characteristics of the three videos (see Figs. 6–8) used to estimate the model parameters are summarized in Table 1. They differ mainly in the age of broilers, in their activity and density (see Table 1).

The first video corresponds to a twenty one days of age broilers flock. This is the video with the higher activity. It is the video with the lower rate of detection, only 97% of present broilers are detected.

Broilers of the next video are 26 days of age. This second video corresponds to the most tightly spaced broilers. Broilers trying to clear a path in such crowded area may hide other broilers and cause tracking errors. The last video corresponds to 37 days of age broilers. Broilers are much bigger compare to those in the two previous videos. Even though the density and the activity are quite low, occlusions might occur due to their biggest size. This being said, this video records the less interactions between broilers compare to the two previous videos.

Table 3
Standard deviation of the absolute difference between measured and estimated positions according to the motion estimation method .

Model	Day 21	Day 26	Day 37
Method 1	13.34	6.71	10.53
Method 2	11.5	7.13	11.21
Method 3	10.26	5.75	8.8
Method 4	9.47	5.77	8.81
Method 5	6.82	4.18	5.87
Method 6	5.47	3.47	4.48

3.3. Parameters estimations

Two data sets are associated to each video. The first data set is the reference one. A surrounding bounding box has been drawn manually around all broilers in order to have a good localization, without any miss detection nor false positive detection. The second data set are data generated by a convolutional neural network. These data are used to test the tracking motion estimation methods. Tracking has been run over the reference data set and then over the test data set for the six following tracking methods which differs by the way the motion is estimated:

- Nearest Neighbor (Method 1)
- Regular motion (Method 2)
- Weighted Average (Method 3)
- Cartesian Least Squares (Method 4)
- Polar Least Squares (Method 5)
- Least Squares Under Constraints (Method 6)

The objective is to have common coefficients for all data sets. A method has to be robust, to the density and to the broiler activities. As a first step, coefficients have been evaluated for each data set of reference corresponding to broilers excursively in movement. As most part of them are generally resting, the idea is to avoid to have parameters biased by too much resting broilers. The objective being to predict motion. The Table 2 gives these parameters estimations given by the least squares method. An analyze has been initiated in order to determine the

Table 2
Least square parameters estimation .

Model	Parameters	Day 21	Day 26	Day 37
Cartesian	$\widehat{\gamma}$	$(0.43 \quad 0.31)$	$(0.33 \quad 0.25)$	$(0.27 \quad 0.17)$
Polar	$\begin{pmatrix}\widehat{\alpha}\\ \widehat{\beta}\end{pmatrix}$	$\begin{pmatrix}0.38 & 0.28 & 0.17\\ -0.07 & -0.09 & -0.07\end{pmatrix}$	$\begin{pmatrix}0.35 & 0.27 & 0.21\\ -0.05 & 0.03 & -0.02\end{pmatrix}$	$\begin{pmatrix}0.34 & 0.27 & 0.18\\ -0.06 & -0.04 & -0.007\end{pmatrix}$

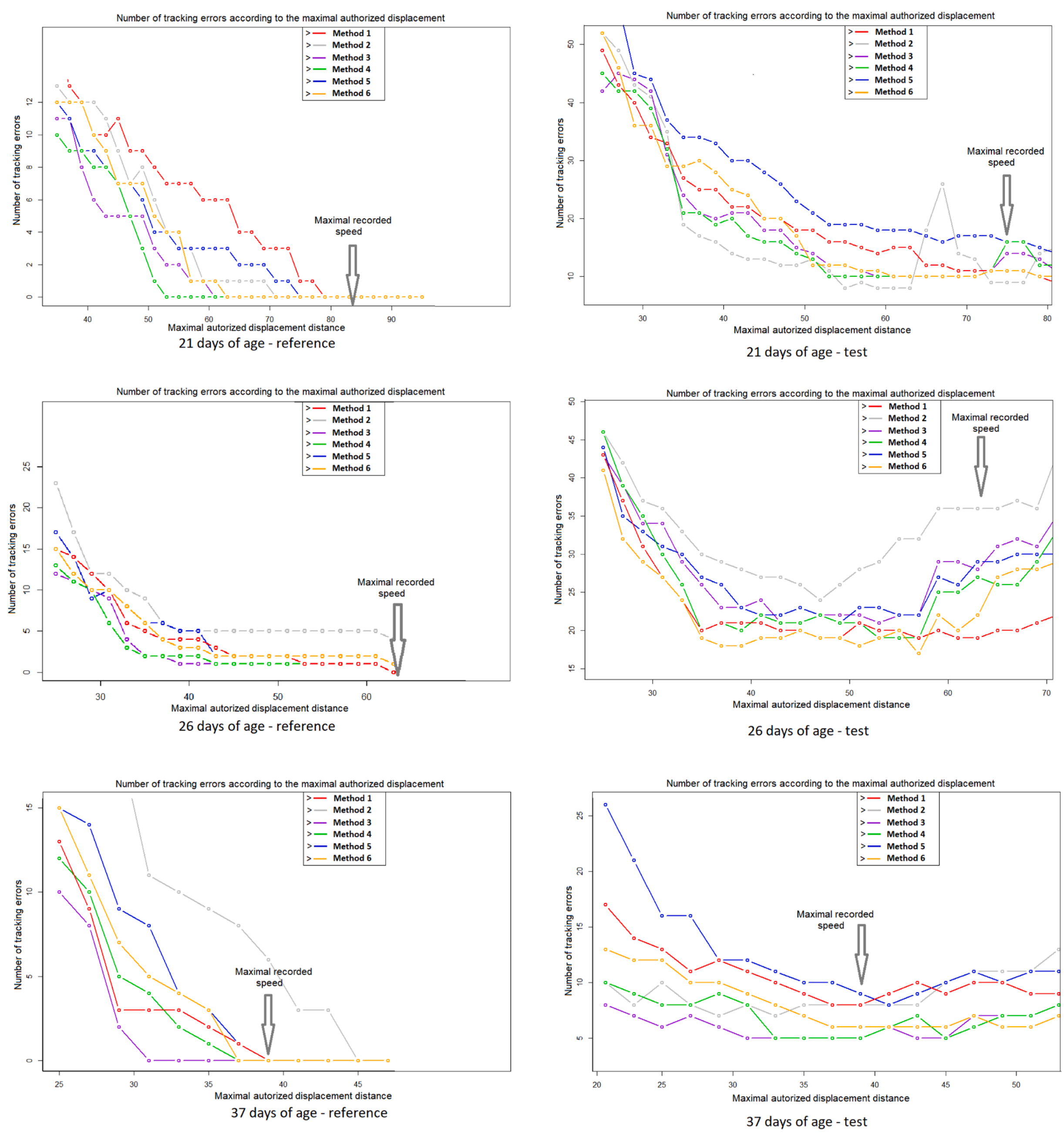

Fig. 9. Number of tracking error according to the authorized displacement distance.

parameter P in the model (13), which is the number of past relevant velocity vectors. By estimating these γ_s parameters, it turned out that they are higher as recorded measures are more recent. The minimum parameter value accepted is then defined such that the model does not take into account value whose contribution leads to a movement lower to one pixel for highest speed values.

Based on these results, the final parameters used for the test data sets are the mean of the three reference data sets. These estimations have not been estimated on a whole data set merging those three ones in order to avoid that one of them prevail over the two others.

$$\hat{\gamma} = \begin{pmatrix} 0.34 \\ 0.24 \end{pmatrix} \tag{29}$$

$$\begin{pmatrix} \hat{\alpha} \\ \hat{\beta} \end{pmatrix} = \begin{pmatrix} 0.36 & 0.27 & 0.19 \\ -0.06 & -0.03 & -0.03 \end{pmatrix} \tag{30}$$

Table 3 gives then a global information on the movement estimation. This test consists in measuring the standard deviation of $\hat{\epsilon} = \hat{\mathbf{V}}^t - \mathbf{V}^T$ for each model. Smaller is the variance values, better is the model prediction. It has to be noticed that it is based on reference data sets with broilers in movement. The method 1 (Nearest Neighbor) seems to be the less adapted one to broilers high speeds and the hybrid the most adapted one. However, in data test most of broilers are resting and detection is not perfect.

4. Results and discussion

One last parameter to take into account is the maximal speed reach by broilers. The fact is that if the tracking leads to wrong detections, wrong identification assignments occurs. They may appear between false positives and miss detected or occluded broilers. There is a compromise to be find between to enable large speed displacements in order to track fast broilers, and to enable only low speed displacement in order to avoid wrong assignments with occluded and miss detected broilers. Testing all the six methods on the reference data sets (no occlusion nor miss detection) may give an idea of the highest speed displacement that can be accepted. The Fig. 5 gives the number of tracking error according to the threshold of largest accepted speed. This threshold is expressed as a ratio of broilers size. A value of 100% refers to the median of the larger bounding box size of every broilers (cf: Table 1). The best method is the one which reaches the minimum of error for the smallest speed threshold possible. Being able to track broilers with a small speed threshold express a good estimation of motion and leads to less wrong identification assignments. Where it might be possible to correct errors due to too high speeds in post processing, it is quite impossible to correct wrong assignment identifications for ID passing from one broiler to another due to miss detections. This is why it is really important to reach the minimum of error as quickly as possible (Fig. 9).

4.1. Analyze of performance for each method

Method 1 ***Nearest neighbor*** As expected Nearest Neighbor method is one of the worsts as predict high speed motions. Even with the reference data (perfect detection), it is one of the last to reach a minimum of errors. Its best performance is found for high density video in test situation. *Method 2* ***Regular motion*** based on [11], it was expected to have good performances for straight movements and good quality of detection. Straight movements are mainly presents in the video of 21 days of age broilers as speeds are the higher. In practice its best performance is effectively find the 21 days of age broilers data test where it reaches quickly the minimum of errors. However performances decrease for low broiler activities both in reference and test data sets. *Method 3* ***Weighted average*** the Weighted Average method from [32] was expected to be robust to miss detection as it uses up to five input variables, however it was said it probably does not use the most adapted coefficients. When it is compared to other methods, it seems to have better performance for reference data set compare to test data test. That means it is dependent on the detection quality. *Method 4* ***Cartesian least squares*** In theory, this method should follow the WA method performance as they are based on the same model. However, it coefficients have been tuned on reference data and it uses less coefficients making it more sensitive to poor detection. In practice, the XY method is always among the two fastest methods to reach the minimum of error. It looks like the best compromise according to the three situations. *Method 5 & 6* ***Polar least squares & least squares under constraints*** Even though their coefficients have been tuned with reference data set, the Polar method remain one of the worst one both for reference and test data sets. However these methods did present the lowest standard deviation in Table 3. It means they are too sensitive to movements due to imprecision of detection. Nevertheless, the limit of outliers by the Hybrid method seems really efficient when the two methods are compared together. Even though the Hybrid method is quite slow to reach the minimum for low densities video (21 and 37 days of age), it is among those which reach the lowest number of tracking errors and seems to reach the best performance for high density video.

A successful method of motion prediction has to be robust to: the density of the broilers, to their activities, to motions derived from the detection imprecision, to the occlusions and miss detections. In addition, it has to reach the best performances for the lowest threshold of authorized speed motion. In this sens the XY method seems to be the best compromise among all the six methods investigated.

5. Conclusion

In this paper, three estimations of motion applied to broilers tracking have been presented. They have been compared to three methods of the literature through three data sets created by our own. The XY method stands out of the group as the best adapted to all situations. However, even though one of them seems to be most adapted to all kind of situation, it is still dependent of maximum authorized speed threshold that seems to vary a lot according to the video. An automatic way to fix max displacement has to be set in order to have a fully robust method.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] W. Luo, J. Xing, A. Milan, X. Zhang, W. Liu, T.-K. Kim, Multiple object tracking: a literature review, Artif. Intell. 293 (2021) 103448.

[2] M. Han, A. Sethi, W. Hua, Y. Gong, A detection-based multiple object tracking method. 2004 International Conference on Image Processing, 2004. ICIP'04. vol. 5, IEEE, 2004, pp. 3065–3068.

[3] Z.-Q. Zhao, P. Zheng, S.-t. Xu, X. Wu, Object detection with deep learning: a review, IEEE Trans. Neural Netw. Learn. Syst. 30 (11) (2019) 3212–3232.

[4] A. Dhillon, G.K. Verma, Convolutional neural network: a review of models, methodologies and applications to object detection, Prog. Artif. Intell. 9 (2) (2020) 85–112.

[5] S.R. Balaji, S. Karthikeyan, A survey on moving object tracking using image processing. 2017 11th International Conference on Intelligent Systems and Control (ISCO), 2017, pp. 469–474, https://doi.org/10.1109/ISCO.2017.7856037.

[6] S. Kothiya, K. Mistree, A review on real time object tracking in video sequences, 2015, pp. 1–4. 10.1109/EESCO.2015.7253705.

[7] Y. Hua, K. Alahari, C. Schmid, Online object tracking with proposal selection. Proceedings of the IEEE International Conference On Computer vision, 2015, pp. 3092–3100.

[8] Q. Wang, L. Zhang, L. Bertinetto, W. Hu, P.H. Torr, Fast online object tracking and segmentation: aunifying approach. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 1328–1338.

[9] J. Luiten, I.E. Zulfikar, B. Leibe, Unovost: unsupervised offline video object segmentation and tracking. Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2020, pp. 2000–2009.

[10] Y. Wei, J. Sun, X. Tang, H.-Y. Shum, Interactive offline tracking for color objects. 2007 IEEE 11th International Conference on Computer Vision, IEEE, 2007, pp. 1–8.

[11] I.K. Sethi, R. Jain, Finding trajectories of feature points in a monocular image sequence, IEEE Trans. Pattern Anal. Mach. Intell. (1) (1987) 56–73.

[12] C.J. Veenman, M.J. Reinders, E. Backer, Resolving motion correspondence for densely moving points, IEEE Trans. Pattern Anal. Mach. Intell. 23 (1) (2001) 54–72.

[13] H.W. Kuhn, The hungarian method for the assignment problem, Naval Res. Logist. Q. 2 (1–2) (1955) 83–97.

[14] V. Salari, I.K. Sethi, Feature point correspondence in the presence of occlusion, IEEE Trans. Pattern Anal. Mach. Intell. 12 (1) (1990) 87–91, https://doi.org/10.1109/34.41387.

[15] K. Rangarajan, M. Shah, Establishing motion correspondence, CVGIP 54 (1) (1991) 56–73.

[16] X. Li, K. Wang, W. Wang, Y. Li, A multiple object tracking method using Kalmanfilter. The 2010 IEEE International Conference on Information and Automation, IEEE, 2010, pp. 1862–1866.

[17] S.-K. Weng, C.-M. Kuo, S.-K. Tu, Video object tracking using adaptive Kalman filter, J. Vis. Commun. Image Represent. 17 (6) (2006) 1190–1208.

[18] L. Zhang, Y. Li, R. Nevatia, Global data association for multi-object tracking using network flows. 2008 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2008, pp. 1–8.

[19] J. Berclaz, F. Fleuret, E. Turetken, P. Fua, Multiple object tracking using k-shortest paths optimization, IEEE Trans. Pattern Anal. Mach. Intell. 33 (9) (2011) 1806–1819.

[20] A. Milan, L. Leal-Taixé, K. Schindler, I. Reid, Joint tracking and segmentation of multiple targets. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 5397–5406.

[21] S. Hamid Rezatofighi, A. Milan, Z. Zhang, Q. Shi, A. Dick, I. Reid, Joint probabilistic matching using m-best solutions. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 136–145.

[22] S.S. Blackman, Multiple hypothesis tracking for multiple target tracking, IEEE Aerosp. Electron. Syst. Mag. 19 (1) (2004) 5–18.

[23] A. Yilmaz, O. Javed, M. Shah, Object tracking: a survey, ACM Comput. Surv. (CSUR) 38 (4) (2006) 13.

[24] D. Comaniciu, V. Ramesh, P. Meer, Kernel-based object tracking, IEEE Trans. Pattern Anal. Mach. Intell. 25 (5) (2003) 564–577.

[25] O. Pele, M. Werman, The quadratic-chi histogram distance family. European Conference on Computer Vision, Springer, 2010, pp. 749–762.

[26] M. Tuceryan, A.K. Jain, Texture analysis. Handbook of Pattern Recognition and Computer Vision, 1993, pp. 235–276.

[27] H. Schweitzer, J.W. Bell, F. Wu, Very fast template matching. European Conference on Computer Vision, Springer, 2002, pp. 358–372.

[28] D.P. Huttenlocher, J.J. Noh, W.J. Rucklidge, Tracking non-rigid objects in complex scenes. 1993 (4th) International Conference on Computer Vision, 1993, pp. 93–101, https://doi.org/10.1109/ICCV.1993.378231.

[29] M. Bertalmio, G. Sapiro, G. Randall, Morphing active contours, IEEE Trans. Pattern Anal. Mach. Intell. 22 (7) (2000) 733–737, https://doi.org/10.1109/34.865191.

[30] R. Vaddi, L. Boggavarapu, H. Vankayalapati, K. Anne, Contour detection using freeman chain code and approximation methods for the real time object detection, Asian J. Comput. Sci. Inf. Technol. 1 (1) (2011) 15–17.

[31] Y. Bar-Shalom, F. Daum, J. Huang, The probabilistic data association filter, IEEE Control Syst. Mag. 29 (6) (2009) 82–100.

[32] H. Karunasekera, H. Wang, H. Zhang, Multiple object tracking with attention to appearance, structure, motion and size, IEEE Access 7 (2019) 104423–104434.

[33] M. Fletcher, K. Warwick, R.J. Mitchell, Application of a hybrid tracking algorithm to motion analysis. CVPR, 1991, pp. 84–89.

[34] K. Godfrey, P. Jones, Signal Processing for Control vol. 79, Springer, 1986.

[35] S. Bozic, Digital and KalmanFiltering, Arnold, London, 1979.

[36] R. Turetsky, The least squares solution for a set of complex linear equations, Q. Appl. Math. 9 (1) (1951) 108–110.