3D tracking based on possibilities rather than probabilities

Guillaume Chiron, Petra Gomez-Krämer, Michel Ménard

3D tracking based on possibilities rather than probabilities
Application to flying honeybees at the beehive entrance

Guillaume CHIRON[1], Petra GOMEZ-KRÄMER[1] and Michel MENARD[1]

[1]*L3i, University of La Rochelle, La Rochelle 17071 Cedex 9, France*

{*guillaume.chiron, petra.gomez, michel.menard*}*@univ-lr.fr*

Keywords: 3D tracking, Honeybees monitoring, Scene constraints, Trajectory database, Colony dynamics

Abstract: This article covers a tracking appoach for multiple simultaneously flying animals. The idea is to constraint the tracking to the physically possible space without any knowledge of the scene. Accordingly, we propose to use a naively made trajectory dataset as a base to track the targets into the possible space rather than the probable space. First, a trajectory database is computed under weak assumptions (permissive classic Kalman Filter). Then, for the effective tracking, we replace the prediction phase of the Kalman Filter to rely only on the dynamic stored in the database of previous trajectories. We show that performances can reach these obtained by other methods based on strong assumptions (e.g. the 3D model of the scene).

1 INTRODUCTION

Forced to observe the worldwide decline of honeybees (Apis mellifera), biologists began to study different hypotheses that could explain the phenomenon. Recently, the authors of [1] highlighted the evidence of behavioral alterations caused by pesticides. In that study, entrance and exit data was collected by a radio-frequency identification monitoring device placed at the beehive entrance. So far, no biological study has been conducted at a big scale on honeybees flight behaviors. In cause, the lack of suitable methods to collect trajectories of honeybees in flight. The only method used by biologists (harmonic radar) is intrusive and suffers from biases. We believe that computer vision can effectively achieve this task with the respect of the application constraints. We note however that some work exists on honeybees trajectoires inside the hive [2].

In a first paper [3], we laid the foundations for a stereo vision-based system for monitoring multiple simultaneously flying honeybees in 3D at the beehive entrance. The tracking follows a detect-before-track approach based on a Kalman Filter (KF) coupled with Global Nearest Neighbor for track assignments. The application is illustrated by Figure 1. Results were satisfying, but could possibly be improved by injecting more assumptions into the tracking process (e.g. for the KF [4]).

Thus, in [5], we proposed to constraint the prediction phase of the Kalman Filter according to the scene (namely the flight board). The idea was to reduce the maximum assignment distance relatively to the distance of the bee from the board. The advantage of the method was shown thanks to a ground truth made of semi-simulated trajectories. However, this method required the acquisition of a 3D model of the flight board, which was possible to compute automatically in that application. But this requirement is clearly a break for the adoption of the method.

Therefore, we propose in this paper a new method to constraint the tracking without having to model the scene. The idea is to use a naively made trajectory database as a base to track the targets into the possible space rather than the probable space as the Kalman Filter does. The trajectory database is computed under weak assumptions (classic KF with low constraints). We use the fact that even if some trajectories of the database are not well computed, the error is compensated by the huge amount of the data.

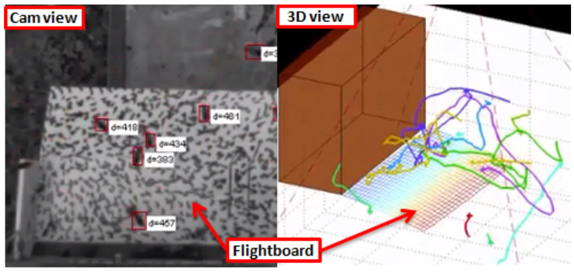

Figure 1: On the left, cropped top camera view of the flight board with detected flying bees. On the right, 3D reconstruction of the beehive with trajectories of bees being tracked at the entrance.

This article is organized as follows. First, we revisit in Section 2 the classic tracking approach based on the Kalman Filter used in [3]. Then we detail in Section 3 a new method for tracking with constraints based on a trajectory database. In conclusion, Section 5 proposes potential improvements and opens perspectives relative to behavioral analysis applications.

2 Tracking based on probabilities

In order to settle the base, the following briefly summarizes an application of the classic Kalman Filter for tracking on flying honeybees in 3D [3].

In a first independent step, the system detects flying bees thanks to a hybrid segmentation that takes advantage of both intensity and disparity images provided by a stereo camera. After projection, each observation is then defined by the 3D coordinates (x,y,z) in the camera coordinate system, which is a 3D Euclidean space with the camera located at $(0,0,0)$.

In a second step, each target is tracked by a Kalman Filter [6]. Despite the apparent rough dynamic of bees, frames were acquired at a sufficiently high frequency (about 47 fps) so that a constant speed model can be assumed. As assumptions on bee dynamic are weak, noise matrices (Q for the model evolution, R for the measure) are set with a relatively high covariance which allows the tracker to be permissive. Let us suppose $Y_{1:n}$ the series of observations corresponding to a target from time 1 to n. A Kalman Filter is instantiated with Y_1 and later destroyed when the step $k > n$. For a given step k, an observation is defined by the vector $Y_k = [x_m, y_m, z_m]^T$, and the estimated state of a target is modeled by a posterior Gaussian probability density with μ defined by the state vector $\hat{x}_k^+ = [x,y,z,\dot{x},\dot{y},\dot{z}]^T$ (combining the 3D position and velocity), and the uncertainty ε defined by the covariance matrix P_k^+. Figure 2 lays down the recursive mechanism of the Kalman Filter.

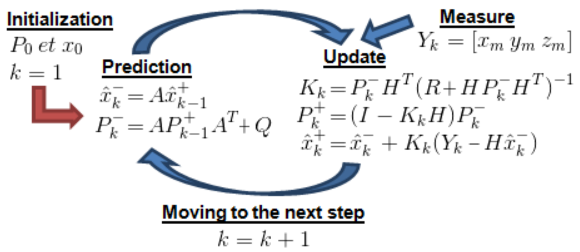

Figure 2: Recursive mechanism of the Kalman Filter for the estimation of x.

In this application, up to 20 targets can be observed at each step, which requires a multi-target tracking approach. The commonly used Global Nearest Neighbor (GNN) [7] method handles observation to track associations, track instantiations and destructions. In GNN, the assignment matrix $A[c_{i,j}]$ represents all the possible associations and the costs generated by those associations. A includes the possibility for each observation to be associated to an existing track, not to be associated to any track or to be associated to a new track. $c_{i,j}$ is the cost for the observation i to be assigned to the possibility j. The best configuration of associations is the solution that minimizes the global cost (e.g. solved by the Hungarian method). The association cost between an observation and a track is computed by the Mahalanobis distance given by:

$$\begin{aligned} d^2 &= (Y - H\hat{x}^-)'S^{-1}(Y - H\hat{x}^-) \\ S &= HP^-H' + R \end{aligned} \tag{1}$$

where Y is the measure vector corresponding to the observation, M the measurement matrix, $\hat{x}^-$ the predicted a priori position, S the innovation covariance with R as the measure noise matrix, and P^- the predicted noise covariance matrix.

Then, in later work, we proposed in [5] to add constraints to the tracking by adapting the uncertainty matrix according the location of the target from the flight board. Despite the better performances, this approach suffers from its need of a 3D model of the scene to be able to calculate distances.

3 Tracking based on possibilities

This section introduces our new approach for adding scene constrains into the tracking process, where contrary to [5], the modeling of the scene is no longer needed.

The application of monitoring honeybees at the beehive entrance is conceived for many hours of video, and enables to collect thousands of trajectories per hour. So the idea of the proposed approach is to take advantage of this huge amount of data that we consider to represent most of bee dynamic possibilities.

3.1 Making of the trajectory database

In a preliminary step, a rough trajectory database is made by using a naive tracking. As we have no knowledge about the scene, the tracker is configured to be permissive, as mentioned in Section 2. Namely, noise matrixes Q and R are set with high values to allow tracks to be associated with relatively farther observations. Naturally, this leads to tracking errors, such as missing tracks, track swaps or track end failures. But even with an high error rate, the chances for an error to occur twice or more times at the same point of the 3D space are low. So, the huge amount of data compensates the potential errors of the trajectory database. Finally, the database represents globally all the dynamic of the bees in flight at an exhaustive number of points of the space. Figure 3 shows a sample of the trajectories collected in the database.

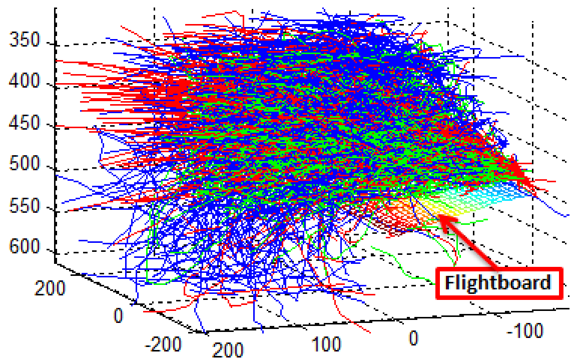

Figure 3: 0.1% of the trajectories constituting the database, with the flight board displayed as reference. Colors are given to trajectories according to their action; red for departure, green for arrival, blue for crossing.

3.2 Kalman Filter prediction based on the trajectory database

Our new tracking method consists in replacing the tracking prediction phase of the Kalman Filter in order to base the state estimation directly on the trajectory dynamic database rather than relying on probabilities. Formally, let Q be a query subtrack for which we want to predict the further state. $DB_{trajectories}$ is the database of trajectories. $DB_{subtracks}$ is the set of all the subtracks of the same length of Q that can be considered from $DB_{trajectories}$. Using a weighted Euclidian distance (letting more importance to the beginning of the tracks), a set C of n closest subtracks of Q is found from $DB_{subtracks}$. Then, considering the set of next moves associated to each subtracks of C, we estimate the further state of Q. The following details the different steps of our approach.

1) Definition of a query subtrack Q of length l corresponding to a track T that we want to estimate the next step (already estimated over the steps $1 \rightarrow k-1$) :

$$Q \leftarrow T_{k-l:k} \quad \text{with} \quad l \leftarrow min(l_Q, T_{length}) \tag{2}$$

with l_Q the maximum length of a query (e.g. 15 steps).

2) Construction of the database of subtracks $DB_{subtracks}$ from $DB_{trajectories}$:

Data: $DB_{subtracks} = \varnothing$
for $T \in DB_{trajectories}$ ***having*** $T_{length} > l$ **do**
  for $i \leftarrow 1$ **to** $T_{length} - l$ **do**
    add $T_{i \rightarrow i+l}$ **in** $DB_{subtracks}$
  end
end

3) Find from $DB_{subtracks}$, the set C of n closest subtracks of Q. Or more specifically the corresponding indexes:

$$\begin{gathered} C\text{-}indexes \leftarrow sort(dist(Q,X),\ \forall X \in DB_{subtracks})_{1:n} \\ \text{using } dist(Q,X) = \textstyle\sum_{i=1}^{l} \omega_i (X_i - Q_i)^2 \end{gathered} \tag{3}$$

with ω proportional to i for example.

4) Replacement of prediction phase of the KF, the mean state and the uncertainty associated by a covariance matrix are given by :

$$\begin{aligned} \hat{x}_k^- &= [\mu_x, \mu_y, \mu_z, ...]^T \\ \hat{P}^-_{k\,(m,n)} &= cov\left(Cx_{(m)}, Cx_{(n)}\right) \end{aligned} \tag{4}$$

with $C\text{-}coords_{(d)}$ is the set of coordinates for each point corresponding to the next step of each subtrack of C, given the dimension d (namely x, y, z, ...). The mean point is given by $\mu_d = \mathbb{E}\left[C\text{-}cords_{(d)}\right]$. Let us notice that we previously applied to each point of C-$coords$ a translation to recenter this potential move in the context of the query track Q, as:

$$\begin{gathered} C\text{-}coords_{(d,i)} = T_{(d,i)_{l+1}} - (T_{(d,i)_l} - Q_{(d,l)}) \\ \text{with } T_{(d,i)} = DB_{subtracks}(C\text{-}Indexes_{(i)}) \end{gathered} \tag{5}$$

with d the coordinate dimension, i the track index and l the point index. Figure 4 illustrates an example of query and associated subtracks for the prediction.

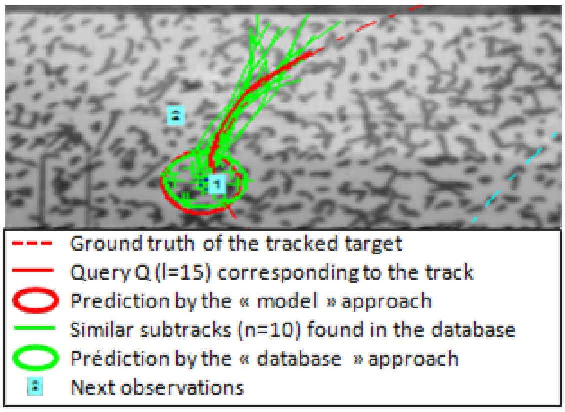

Figure 4: Top view comparison of both constrained approaches. The ellipses represents the prior estimated Gaussian probability densities.

4 Results

The success rate of our approach is correlated to the size of the database. Figure 5 shows that in a hypothetical 2D situation, for a database made of at least 25k trajectories, the mean error for state prediction reaches a value comparable to other methods. Naturally, when adding a dimension (such as in our application which is in 3D), it seems more difficult to reach this performance. It may be possible with much more data. Concerning the prediction of the uncertainty, the more samples are available in the database, the lower the incertitude of prediction would potentially be, which appears logical.

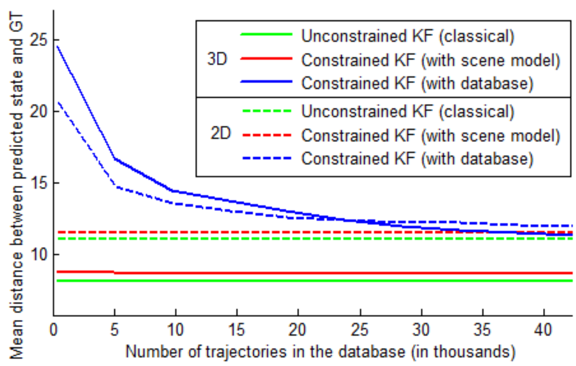

Figure 5: State prediction error relatively to the size of the database. A 2D situation (simpler) is shown as a comparison to 3D.

Similarly to [3], the comparative evaluation of the methods is made thanks to a pseudo-simulated ground truth. Namely, in this case, scenarios were built by time shifting random picked real trajectories from a base of 200 manually annotated trajectories. For the creation of the scenarios, we focused on the problems highlighted in [3, 5]: the simultaneous (or close) departures and arrivals at the flight board. Thus, in this current evaluation, tracks were simulated in couples (a landing track followed by a near take off).

We evaluated the performance of both constrained methods, "Constrained KF with scene model" introduced in [5], and "Constrained KF with database" proposed in this paper, over 100 random generated scenarios, under each of the 6 following configurations: 2, 4, 6, 8, 10 and 12 simultaneous couples. Also, as a reference, we compared the results with the classical "Unconstrained KF" introduced in [3]. The metric used for the evaluation is the ratio of well recovered tracks. A track is considered well recovered if at least 90% of its associated observations match one of the ground truth tracks. Figure 6 show the advantage of both constrained methods which increase the performance by around 10%. But, among these two methods, the one proposed in this paper is especially interesting as it doesn't require any scene model. The weakness of the proposed method is the computation time, as it requires browsing the entire database at each step and for each target. However, even if it is currently taking a few seconds per step (on non-optimized code), distance calculations on this kind of data is highly parallelizable.

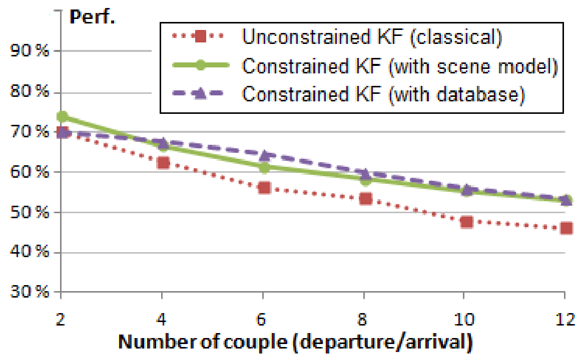

Figure 6: Compared mean performances over 100 scenarios of the unconstrained and constrained methods according to the number of simultaneous targets

5 Conclusion

We proposed in this article a method for tracking simultaneously multiple flying insects considering the possible space, namely by constraining the prediction relatively to the historical moves of other members of the colony. The dynamic of motion is stored in a massive trajectory database, which is naively established in a preliminary step thanks to a permissive tracker. The base contains errors such as swaps and track end failures. However, the massive amount of tracks offers some redundancy that avoids those errors to propagate through the system. Results are satisfying on the generated scenario, which highlighted one difficulty of our application, namely the creation and destruction of tracks around the flight board. Nevertheless, when online realtime tracking is needed, classical methods could still be preferable for simpler scenarios.

For a further version, a first idea would be to take into account the distance between the query track Q and the subtracks in order to weight the calculation of the mean and the covariance. A second idea would be to avoid using the Kalman Filter which is monomodal. It would be interesting to support multi-modal predictions, for example by using a particle filter where each particle would be associated to a trajectory of the database.

Concerning perspectives, this work is a preliminary for a behavioral analysis. Based on 3D trajectories collected, we will be able to build a database of bee behaviors in front of the beehive entrance. Some methods based on non-parametric Bayesian approaches [8] seem to be interesting candidates for clustering behaviors.

6 Acknowledgment

This work was supported by the European Regional Development Fund (contract: 35053) and the Poitou-Charentes Region. The videos used in this work were taken during autumn 2012 on the INRA site (National Institute for Agronomy Research) at Magneraud. We would like to thank INRA's biologists for their support and expertise in helping to collect exploitable data under different conditions (e.g., activity, weather).

REFERENCES

[1] Henry, M., Beguin, M., Requier, F., Rollin, O., Odoux, J.F., Aupinel, P., Aptel, J., Tchamitchian, S., Decourtye, A.: A common pesticide decreases foraging success and survival in honey bees. Science **336**(6079) (2012) 348–350

[2] Veeraraghavan, A., Chellappa, R., Srinivasan, M.: Shape-and-behavior encoded tracking of bee dances. Pattern Analysis and Machine Intelligence, IEEE Trans. **30**(3) (2008) 463–476

[3] Chiron, G., Gomez-Krämer, P., Ménard, M.: Detecting and tracking honeybees in 3D at the beehive entrance using stereo vision. EURASIP Journal on Image and Video Processing (1) (2013) 59

[4] Simon, D., Chia, T.L.: Kalman filtering with state equality constraints. Aerospace and Electronic Systems, IEEE Transactions on **38**(1) (2002) 128–136

[5] Chiron, G., Gomez-Krämer, P., Ménard, M.: 3d tracking of honeybees enhanced by environmental context. In: Image Analysis and Processing - ICIAP 2013. Volume 8156. (2013) 702–711

[6] Kalman, R., et al.: A new approach to linear filtering and prediction problems. Trans. of the ASME – Journal of basic Engineering **82**(1) (1960) 35–45

[7] Blackman, S., Popoli, R.: Design and analysis of modern tracking systems. Volume 685. Artech House Norwood, MA (1999)

[8] Wang, X., Ma, X., Grimson, W.E.L.: Unsupervised activity perception in crowded and complicated scenes using hierarchical bayesian models. IEEE, PAMI **31**(3) (2009) 539–555