3D TRACKING TARGETS VIA KINEMATIC MODEL WEIGHTED PARTICLE FILTER

Xi En Cheng[1,2], *Shuo Hong Wang*[1], *Yan Qiu Chen*[1*]

[1] School of Computer Science, Shanghai Key Laboratory of Intelligent Information Processing, Fudan University, Shanghai, China [2] Jingdezhen Ceramic Institute, Jingdezhen, China
chenyq@fudan.edu.cn

ABSTRACT

Automatically and reliably tracking numerous flying objects in 3D space is of great significance for not only scientific researches such as collective behavior analysis, but also practical applications such as designing multi-agent robots. However, it remains a challenging task due to the large population, similar appearance, and severe occlusion happening in 2D images. This paper proposes a 3D tracking method that is capable of tracking individuals of a swarm of flying objects using the particle filtering technique. Each particle is not only weighted by the observation model but also weighted by the kinematic model. The kinematic model is modeled by learning a long short-term memory network on sequences of velocities. Experimental results show that the kinematic model significantly improves the efficiency of estimating target's motion state, and show that the proposed method outperforms the state-of-the-art methods.

Index Terms— 3D tracking, particle filter, kinematic model, long short-term memory network, observation model

1. INTRODUCTION

Given at least two cameras, being placed at different locations and filming the same flying objects, we aim to automatically estimate each object's motion states in 3D space using videos captured by these cameras. Estimating an object's motion states using videos is known as object tracking, and that is one of key steps of video analysis [1]. For aforementioned problem of object tracking, the imaging system has to be stationary during capturing videos and cameras have to be synchronized, and also the parameters of cameras are usually known, *i.e.* cameras are calibrated [2].

The aforementioned problem of object tracking is mostly introduced by biological researches. For many years, scientists have been interested in understanding the behavior patterns exhibited by large groups of flying animals, *e.g.* bird flocks [3, 4] and insect swarms [5, 6]. The object tracking technology is the most effective gateway to acquire quantitative motion data for such researches [7]. An example is shown in Fig. 1. The challenge of object tracking in these researches depend on several factors, such as the large number of objects (can be as many as tens of to hundreds of objects) and the similar appearance of objects [8–12].

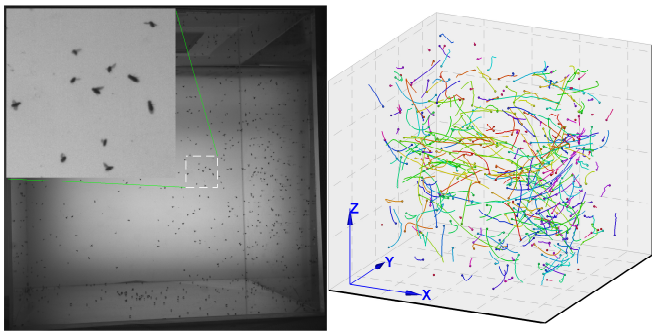

Fig. 1. An example of a camera image and the snapshot of objects with the 20 most recent positions in 3D space. Each object is color coded.

In order to obtain motion data of flying objects in 3D space, multiple synchronized and calibrated cameras are employed to capture videos. It needs methods to establish cross-view and cross-frame correspondence for methods to achieve the purpose. Due to severe occlusion and mutually similar appearance, finding correspondences across multiple views poses severe challenges. And moreover, since we are expected to film the entire motion process in the cameras' field-of-view (FOV), each object may takes up only a few pixels in the images. It makes the observation of objects very coarse. Therefore, there are little visual cues about each object's appearance and for distinguishing objects. From this viewpoint, most tracking methods treat targets as points [8–11, 13] or spheres [14], and thereby estimate a target's position over time. Recently, Cheng *et al*. [15] obtains both a target's position and orientation over time by treating targets as ellipsoids.

Currently, 3D trackers using the particle filtering technique obtain the state-of-the-art performance [10, 15]. These methods automatically generate many hypothetic targets in 3D space while new objects are detected, and adopt the particle filtering technique to inference the posterior distribution of each target's motion state at each moment. Observations from

*Corresponding author. Thanks to National Natural Science Foundation of China, Grant No. 61175036 for funding.

2D images are incorporated to compute each particle's weight according to the observation model. The essential principle of these methods is that non-existing targets can not satisfy the observation model for a long term, and thus can be eliminated in the following few moments.

In this paper, we propose a 3D tracker that follows the particle filtering framework (Section 3). However, since there is indeed little cues that can be used to compute a particle's weight, we introduce the kinematic model (Section 2). Each particle is weighted not only by the observation model (Section 4) but also by the kinematic model (Section 3.1). The kinematic model outputs the probability of a sequence of motion data (usually the velocities from beginning to the current moment) being produced by targets. Since a target's motion process up to moment $t-1$ is supposed to be known during the period of tracking, this model thereby tells the probability of the target moving to a sample state at moment t.

The kinematic model is modeled by learning a long short-term memory (LSTM) network (Section 2). The LSTM network is a special kind of recurrent neural network (RNN), and it is introduced by Hochreiter and Schmidhuber [16] and works well on modeling the variable-length sequential signals. There are many variants on the network's structure and units. A dramatic variation on the LSTM network is replacing the network's units by the gated recurrent units (GRUs), introduced by Cho *et al*. [17]. Though it is found that these variants are all about the same [18], in this work we prefer using the GRUs to build the LSTM network for its less amount of parameters and simpler to compute and implement.

2. THE KINEMATIC MODEL

The LSTM networks are proved to produce promising performance on analyzing long sequential data [19], we prefer to model the motion process using the LSTM network. While there are numerous LSTM variants, we adopt the implementation proposed by Cho *et al*. [17]. Fig. 2 shows the gated recurrent unit (GRU) and the unrolled network structure. The GRUs at each time step t can be defined as a collection of vectors in $\mathbb{R}^d$: an udpate gate $\mathbf{z}_t$, a reset gate $\mathbf{r}_t$, and a hidden state (the unit's output) $\mathbf{h}_t$. d is the number of units in the network. The entries of the gating vectors $\mathbf{z}_t$ and $\mathbf{r}_t$ are all in $[0, 1]$. The transition equations are defined as:

$$\begin{aligned}
\mathbf{z}_t &= \sigma\left(\mathbf{W}_{zv}\mathbf{v}_t + \mathbf{W}_{zh}\mathbf{h}_{t-1}\right) \\
\mathbf{r}_t &= \sigma\left(\mathbf{W}_{rv}\mathbf{v}_t + \mathbf{W}_{rh}\mathbf{h}_{t-1}\right) \\
\tilde{\mathbf{h}}_t &= \tanh\left(\mathbf{W}_{hv}\mathbf{v}_t + \mathbf{W}_{hh}(\mathbf{r}_t \odot \mathbf{h}_{t-1})\right) \\
\mathbf{h}_t &= (1 - \mathbf{z}_t) \odot \mathbf{h}_{t-1} + \mathbf{z}_t \odot \tilde{\mathbf{h}}_t
\end{aligned} \tag{1}$$

where $\mathbf{v}_t$ is the input at each time step t, σ denotes the logistic sigmoid function and $\odot$ denotes elementwise multiplication. The weight matrix $\mathbf{W}_{zv}$ denotes the weights of connections between input $\mathbf{v}_t$ and update gate $\mathbf{z}_t$, and so on. Intuitively, the reset gate controls how to combine the new input with the previous hidden state, and the update gate defines how much of the previous hidden state to keep around.

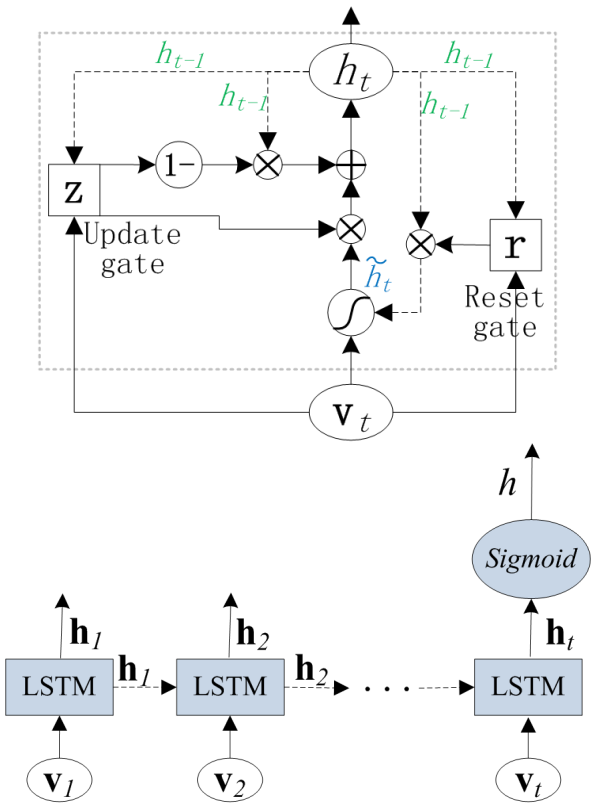

Fig. 2. Illustration of the LSTM network. The upper panel shows a GRU. The dashed line denote the recurrent connection. The lower panel shows the unrolled LSTM network. The box denotes a single LSTM layer, and the output $\mathbf{h}_i, i = 1..t$ is a vector of outputs of all GRUs in the layer.

Here a kinematic model usually contains a single LSTM layer followed by a sigmoid non-linear layer as depicted in the lower panel of Fig. 2. Thus, from an input sequence $\mathbf{v}_{1:t} = \{\mathbf{v}_k | k = 1..t\}$ up to moment t, the units in the LSTM layer will produce a representation sequence $\{\mathbf{h}_{1:t}\}$. This representation is fed to the sigmoid layer to determine the probability of the input sequence, h.

2.1. Training set

To learn the kinematic model of targets, we collect sequences of velocities as the training samples. A certain sequence of velocities is defined as $\mathbf{v}_{1:t} = \{\mathbf{v}_k | k = 1..t\}$, which denotes a certain target's velocities up to moment t. The training samples are classified into two classes: *good samples* and *bad samples*. A good sample is defined as $\{\mathbf{v}_{1:t}, y^g\}, y^g \in (0.9, 1]$; a bad sample is defined as $\{\{\mathbf{v}_{1:t-1}, \tilde{\mathbf{v}}_t\}, y^b\}, y^b \in [0, 0.1)$, in which $\tilde{\mathbf{v}}_t$ denotes the corrupted velocity of a certain target at moment t.

The LSTM network is probably over-fitting on training

samples. The easiest and most common method to reduce over-fitting is to artificially augmenting the training set. The augmentation approach for sequential samples is only performed on the last entry of a sequence. Let $\mathbf{v}_{1:t}$ denote the sequence of velocities of a certain target up to moment t, the augmentation is only performed on the velocity at the last moment, $\mathbf{v}_t$. That is, the samples augmented from the original sequence is defined as

$$\{\mathbf{v}_{1:t-1}, \mathbf{v}_t + \nu \mid \nu \sim \mathcal{N}(\mu, \Sigma)\} \tag{2}$$

where $\mathbf{v}_{1:t-1}$ denotes the actual velocities of a certain target up to moment $t-1$ and $\mathbf{v}_t$ denotes the target's actual velocity at moment t. For good samples, the parameters of the Gaussian noise is set to $\mu = \mathbf{0}$ and $\Sigma = 0.1 * \mathbf{I}$ where $\mathbf{I}$ denotes the identity matrix, therefore ν denotes the white noise. The bad samples are those been augmented by corrupting the velocity $\mathbf{v}_t$ with a biased noise, *e.g.* $\mu = \mathbf{1}$ and $\Sigma = \mathbf{I}$.

3. 3D TRACKER USING PARTICLE FILTER

In this work, we represent a target as a sphere in 3D space, and thus the target at moment t is defined as $\{(x, y, z)^{\mathrm{T}}, \tilde{l}\}$ where the constant scalar $\tilde{l}$ denotes the diameter of the sphere (*e.g.* in the experiments, $\tilde{l} = 2.73$ mm is the average body length of fruit flies). By treating a target moving in 3D space as a dynamic system, we utilize X_t to denote the target's state at moment t and defined as

$$X_t = (x_t, y_t, z_t, x_{t-1}, y_{t-1}, z_{t-1})^{\mathrm{T}} \tag{3}$$

We adopt the first-order linear extrapolation (FLE) as the dynamic model. The FLE model assumes that the next state is defined by the linear extrapolation of the last two positions. The transition function is defined as:

$$X_t = \begin{bmatrix} 2\mathbf{I}_3 & -\mathbf{I}_3 \\ \mathbf{I}_3 & \mathbf{0}_3 \end{bmatrix} X_{t-1} + n_t \tag{4}$$

where $\mathbf{I}_3$ is a 3×3 identity matrix and $n_t \sim \mathcal{N}(0, \Sigma)$ is the transition noise.

Let $Z_{1:t} = \{Z_k | k = 1..t\}$ denote the set of all available observations up to moment t, the Bayesian inference can be formulated as a problem of estimating the posterior probability $p(X_t|Z_{1:t})$ [20]. Under the first-order Markov assumption and the Bayes' rule, we can get the well-known equation of Bayesian filtering

$$\begin{array}{cc} p(X_t|Z_{1:t}) & \propto p(Z_t|X_t) \int p(X_t|X_{t-1}) p^- dX_{t-1} \\ p^- & \equiv p(X_{t-1}|Z_{1:t-1}) \end{array} \tag{5}$$

where $p(Z_t|X_t)$ is called the observation model.

According to the particle filtering solution, the posterior of a target's state at moment t, $p(X_t|Z_{1:t})$, is approximated by a set of weighted particles: $\{(X_t^i, w_t^i) | i = 1..N\}$. The target's state, $\hat{X}_t$, is computed as the expectation

$$\hat{X}_t = E(X_t|Z_{1:t}) = \sum_{i=1}^{N} w_t^i X_t^i \tag{6}$$

where each weight w_t^i is proportional to the likelihood of the sampled state given the observation at moment t, *i.e.* $w_t^i \propto p(Z_t|X_t^i)$. That is, the observation model is used to measure the weight of particles. The observation model is an essential issue for object tracking methods using particle filter.

3.1. Weighting particle using the kinematic model

Let $\mathbf{v}_{1:t-1} = \{\mathbf{v}_k | k = 1..t-1\}$ denote the target's velocities up to moment $t-1$, and X_{t-1} denote the target's state at moment $t-1$. At moment t we have a sample state X_t^i, and thus the hypothetical velocity at moment t, $\tilde{\mathbf{v}}_t$, is computed as $\tilde{\mathbf{v}}_t^i = X_t^i - X_{t-1}$. Therefore, the hypothetical sequence of velocities is defined as

$$\{\mathbf{v}_{1:t-1}, \tilde{\mathbf{v}}_t^i\} \tag{7}$$

This sequence of velocities is the input of the LSTM network, and the output, h_t^i, is the probability of the target that has taken velocities $\mathbf{v}_{1:t-1}$ up to moment $t-1$ and then takes the velocity $\tilde{\mathbf{v}}_t^i$ at moment t. This gives us an approach to weight a particle independent with the observation model. That is, the set of particles is thereby defined as $\{(X_t^i, w_t^i, h_t^i) | i = 1..N\}$ and the estimated motion state, $\hat{X}_t$, is thereby computed as

$$\hat{X}_t = \sum_{i=1}^{N} w_t^i h_t^i X_t^i \tag{8}$$

In practice, a particle is not only weighted by h_t^i but also screened by h_t^i, *e.g.* a particle is ignored while the corresponding h_t^i is less than 0.5.

4. THE OBSERVATION MODEL

The observation model is used to measure the likelihood of particles. However, in the proposed 3D tracking method, particles represent a target's states in 3D space, while the observations are in 2D images. Here matching particles and observations, *i.e.* data association, is utilized by employing the projection matrices of cameras.

4.1. Probabilistic occupancy map

The Gaussian background model is capable to output for each frame a probabilistic occupancy map, the pixel value of which indicates the probability of each pixel belonging to the foreground. Particularly, for each camera c, let μ^c be the mean image and σ^v be the standard deviation image. In this work,

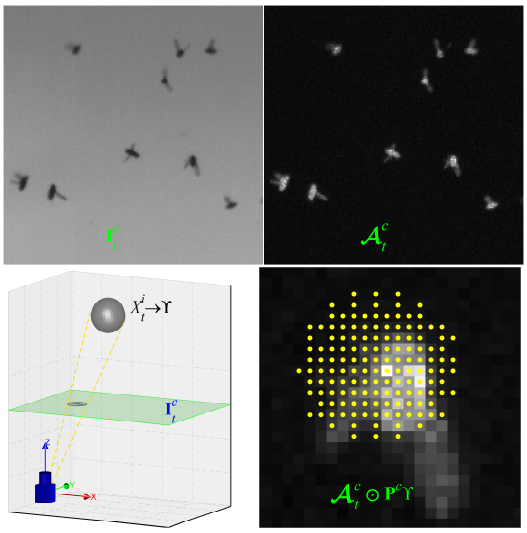

Fig. 3. The observation model on the occupancy map relies on the probability gate that is determined by projecting the sphere into the camera image.

all images are gray scale images. Therefore, for each camera image $\mathbf{I}_t^c$ at moment t, the occupancy map is computed as

$$\mathcal{A}_t^c \propto \frac{\mathbf{I}_t^c - \mu^c}{\sigma^c} \tag{9}$$

Fig. 3 shows a typical camera image and the corresponding occupancy map. At each moment t, the occupancy maps of all camera images can be back-projected into 3D space and determine the occupancy map of objects in a volume. This is a standard visual hull procedure. Here we adopt a discriminative strategy to apply the occupancy map.

As aforementioned, we treat an object in 3D as a sphere. Thus a particle that represents a sample state can be represented by a sphere. We project the sphere onto each camera image and collect the occupancy probability of pixels that are overlapped by the projected sphere image, as shown in Fig. 3. The observation model on the occupancy map is thereby defined as

$$p(Z_t|X_t^i) \propto \sum \mathcal{A}_t^c \odot [\mathbf{P}^c \Upsilon] \tag{10}$$

where Υ denotes the sphere that represents the particle and $\mathbf{P}^c$ is the projection matrix of camera c, and $\odot$ denotes the pixelwise multiplication.

4.2. Body model

The flying animals can make wing-strokes very fast (*e.g.* the fruit fly takes less than 4 ms [21] for one wing-stroke), meaning the wings' positions between consecutive frames are inconsistent (see Fig. 4) unless the camera's frame-rate is very high, such as 8,000 frames per second (FPS) [21]. This is unusual for conventional multi-camera system. On the other hand, as shown in Fig. 4, a target's body part is consistent.

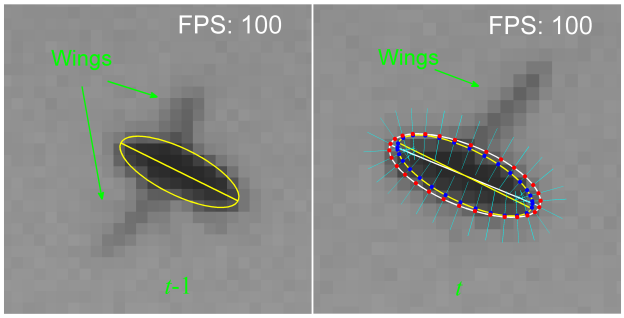

Fig. 4. The body model. The distance D_2 between two ellipses is defined as the sum of all pairwise distances between red points and blue points. The cyan lines are perpendicular to the white ellipse. Yellow ellipses are ellipses at moment $t-1$.

That is, blobs that represent a target's body part in all images are consistent between consecutive moments. Moreover, the intensities of all pixels in blobs jointly encode features of a target's intrinsic appearance, depth, background and illumination. These factors are approximately constant for consecutive frames.

Inspired by [15], here the body model is defined as two components $(\mathbf{E}_t^c, \mathbf{q}_t^c)$, where $\mathbf{E}_t^c$ is a positive definite symmetric matrix and denotes the ellipse that fits on the blob, and $\mathbf{q}_t^c$ denotes the color histogram of the blob. The observation model on the body model is thereby defined as:

$$p(Z_t|X_t^i) \propto \exp\left(-D_1[\mathbf{q}_t^c, \mathbf{q}_{t-t}^c]\right) \exp\left(-D_2[\mathbf{E}_t^c, \mathbf{E}_{t-1}^c]\right) \tag{11}$$

where D_1 measures the distance between two histograms and is derived from the Bhattacharyya coefficient [22], D_2 measures the distance between two ellipses and its computation is depicted in Fig. 4.

5. EXPERIMENTS

Although in principle, two cameras are sufficient for stereo imaging. Three or more cameras are typically required to resolve the ambiguities between targets and to avoid false identifications. We house fruit flies in a transparent acrylic box of size 360 mm, and illumination is provided by planar fluorescent lights. Three monochrome CMOS cameras which have been geometrically calibrated and temporally synchronized were used to capture videos of the flying fruit flies. The resolution of camera was $2048v \times 2040h$ and the frame rate was 100 FPS. We compare the performance with the state-of-the-art methods: Wu2009 [8], Liu2012 [10].

5.1. Ground truth

We have collected 4 dataset (groups of videos) (*DF1-DF4*) with the populations $\{1, 2, 5, 10\}$ respectively. Each group

contains 3 videos and each video contains 1000 frames. The ground truth is created in two phase. At the first phase, by using straight forward application of "track-by-detection" we generate the ground truth of *DF1*. A target's 3D position is reconstructed using the blobs detected in images. Since the parameters of cameras are known, these 3D positions embed the physical dimensions (*e.g.* in millimeters). Therefore the velocities computed using positions of consecutive moments are the target's real-world velocities. It should be noted that the target's trajectory is smoothed before computing velocities.

Then at the second phase, we can generate the initial training set and train the LSTM network. Following the bootstrapping strategy, the ground truth of *DF2-DF4* is created by running the propose 3D tracker and manually validate the results one by one. And finally, the training set for training the LSTM network contains samples from all datasets, and we have also created the ground truth for all datasets.

5.2. Experiments 1

We have collected recordings of flying fruit flies and the population varies in several hundreds. However, manually creating the 3D ground truth is infeasible: (i) each target only has small image area and resembles each other; (ii) there are only 2D observations. Therefore, in Table 1 we only present the quantitative comparison result on *DF4*, in which "Ours-" denotes our 3D tracker but the kinematic model is disabled. We show the other typical results in Fig. 5.

Table 1. Performance comparison on fruit fly dataset

	Method	P	R	F1	Frag	IDS
	Wu2009	0.929	0.883	0.905	6.3	2.6
DF4	Liu2012	0.876	0.873	0.875	3.0	1.8
(10)	Ours-	0.941	0.923	0.932	2.8	1.6
	Ours	0.991	0.986	**0.989**	**0.9**	**0.3**

F1: F1-measure; Frag: Fragments; IDS: ID Switches

5.3. Experiments 2

Since aforementioned reasons, existing methods prefer using simulation data to evaluate performance [8, 10, 13]. As a complement of experiments, we also evaluate performance on simulation data. We adopt the CLEAR MOT metrics [23] to measure the performance of methods and choose the sphere diameter $\tilde{l} = 2.73$ mm as the matching criterion. We create five simulation datasets (*DS1-DS5*) with the number of targets $N = \{50, 100, 150, 200, 250\}$, respectively. The detail of simulation system is presented in supplemental materials.

Each experiment repeats three times, the reported results are the average score for each method. Compared to the proposed method, Fig. 6a shows that Wu2009 obtains nearly same scores (low MOTP score) on *DS1* and *DS2* but worse results on *DS3-DS5*. Wu2009 tracks targets using 2D observations and then reconstructs 3D trajectories using 2D trajectories across views. Its precision mostly depends on the target detection at the first step, and thereby decreases as the population increases causing severe occlusions in 2D images.

Fig. 6b shows Liu2012 obtains almost same scores (high MOTA score) on *DS1* and *DS2*, as the proposed method does; but Liu2012 shows fastly decreasing as the population increases. Liu2012 accepts particles even if they are only "observed" in one camera view, and thus obviously increases the possibility of trackers being distracted by other targets and thus decreases its robustness. Concerning the proposed method, those particles are rejected by the kinematic model.

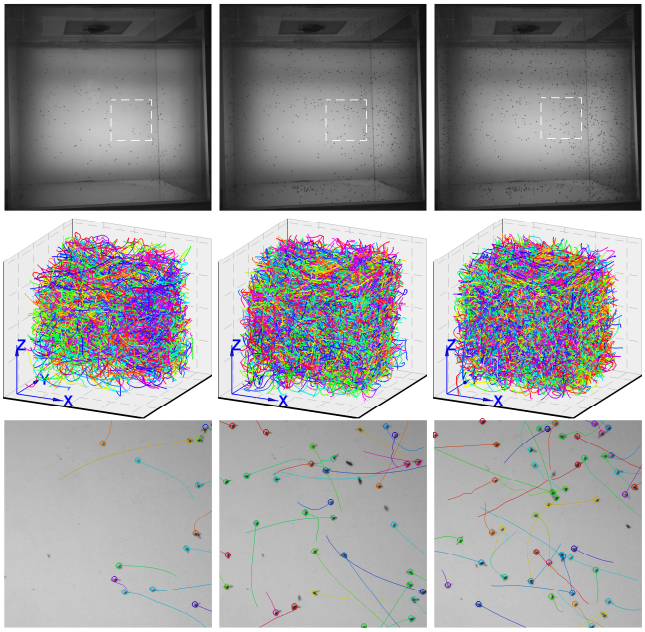

Fig. 5. Typical results of the proposed method on tracking fruit flies. Top row: Snapshots of camera images. From left to right, the population is ≈ 150, ≈ 400, and ≈ 600 respectively. Middle row: The full trajectories of each recording. Trajectories are color coded. Bottom row: Snapshots of the positions projected onto the camera image (only show the patch marked by the rectangle in the top row), tails represent the twenty recent positions of targets.

6. CONCLUSION

The kinematic model of flying objects is a strong prior for object tracking. In this paper, we propose that the kinematic model can be learned by a LSTM network using the sequences of velocities. We thereby propose a 3D tracking method that can accurately and reliably tracking numerous flying targets, and it performs well even though the raw observation is very limited image areas.

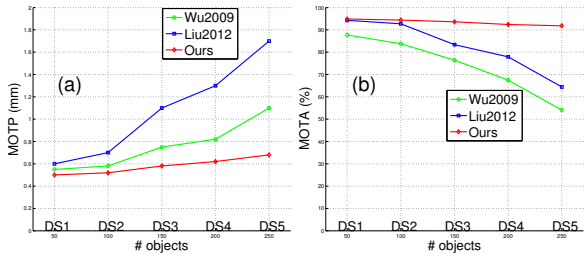

Fig. 6. Performance of the proposed method evaluated on simulation data, compared with state-of-the-art methods.

7. REFERENCES

[1] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Comput. Surv.*, vol. 38, no. 4, pp. 13, 2006.

[2] Z. Zhang "A flexible new technique for camera calibration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 11, pp. 1330–1334, 2000.

[3] M. Ballerini, N. Cabibbo, R. Candelier, A. Cavagna, E. Cisbani, I. Giardina, et al., "Interaction ruling animal collective behavior depends on topological rather than metric distance: Evidence from a field study," *Proc. Natl Acad. Sci.*, vol. 105, no. 4, pp. 1232–1237, 2008.

[4] M. Nagy, Z. Ákos, D. Biro, and T. Vicsek, "Hierarchical group dynamics in pigeon flocks," *Nature*, vol. 464, no. 7290, pp. 890–893, 2010.

[5] S. Butail, N. Manoukis, M. Diallo, J.M. Ribeiro, T. Lehmann, and D.A. Paley, "Reconstructing the flight kinematics of swarming and mating in wild mosquitoes," *J. R. Soc. Interface*, 2012.

[6] J.G. Puckett, D.H. Kelley, and N.T. Ouellette, "Searching for effective forces in laboratory insect swarms," *Sci. Rep.*, vol. 4, 2014.

[7] A.I. Dell, J.A. Bender, K. Branson, I.D. Couzin, G.G. de Polavieja, et al., "Automated image-based tracking and its application in ecology," *Trends in Ecology & Evolution*, vol. 29, no. 7, pp. 417 – 428, 2014.

[8] Z. Wu, N.I. Hristov, T.L. Hedrick, T.H. Kunz, and M. Betke, "Tracking a large number of objects from multiple views," in *ICCV*, 2009, pp. 1546–1553.

[9] A.D. Straw, K. Branson, T.R. Neumann, and M.H. Dickinson, "Multi-camera real-time three-dimensional tracking of multiple flying animals," *J. R. Soc. Interface*, vol. 8, no. 56, pp. 395–409, 2011.

[10] Y. Liu, H. Li, and Y.Q. Chen, " Automatic Tracking of a Large Number of Moving Targets in 3D," in *ECCV*, 2012, pp. 730–742.

[11] R. Ardekani, A. Biyani, J.E. Dalton, J.B. Saltz, M.N. Arbeitman, J. Tower, et al., "Three-dimensional tracking and behaviour monitoring of multiple fruit flies," *J. R. Soc. Interface*, vol. 10, no. 78, pp. 1–13, 2013.

[12] D.H. Kelley and N.T. Ouellette, "Emergent dynamics of laboratory insect swarms," *Sci. Rep.*, vol. 3, pp. 1–7, 2013.

[13] D. Zou, Q. Zhao, H.S. Wu, and Y.Q. Chen, "Reconstructing 3d motion trajectories of particle swarms by global correspondence selection," in *ICCV*, 2009, pp. 1578–1585.

[14] Z. Wu, A. Thangali, S. Sclaroff, and M. Betke, "Coupling detection and data association for multiple object tracking," in *CVPR*, 2012, pp. 1948–1955.

[15] X.E. Cheng, Z-M Qian, S.H. Wang, N. Jiang, A. Guo, and Y.Q. Chen, "A novel method for tracking individuals of fruit fly swarms flying in a laboratory flight arena," *PLoS ONE*, vol. 10, no. 6, pp. e0129657, 2015.

[16] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Compt.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[17] K. Cho, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," in *EMNLP*, 2014, pp. 1724–1734.

[18] K. Greff, R.K. Srivastava, J. Koutník, B.R. Steunebrink, and J. Schmidhuber, "LSTM: A search space odyssey," *CoRR*, vol. abs/1503.04069, 2015.

[19] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[20] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, 2002.

[21] A.J. Bergou, L. Ristroph, J. Guckenheimer, I. Cohen, and Z.J. Wang, "Fruit flies modulate passive wing pitching to generate in-flight turns," *Phys. Rev. Lett.*, vol. 104, no. 14, pp. 1–4, 2010.

[22] P. Prez, C. Hue, J. Vermaak, and M. Gangnet, " Color-Based Probabilistic Tracking," in *ECCV*, 2002, pp. 661–675.

[23] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: The clear mot metrics," *EURASIP J. Image Video Processing*, 2008.