LEARNING KINEMATIC MODEL OF TARGETS IN VIDEOS FROM FIXED CAMERAS

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
{x.cheng, sh_wang, chenyq}@fudan.edu.cn

ABSTRACT

Object tracking is a key step of video analysis, while a motion model is crucial for object tracking. Concerning videos captured with fixed cameras, a sequence of a target's motion data may suggest the target's kinematic model with respect to the imaging system. In this paper we model the target's kinematic model by learning a long short-term memory network. This kinematic model can serve as a discriminative model and determine the probability of a sequence of velocities. In order to improve the expressive ability of the kinematic model, we partition units of the network into groups and activate groups at different temporal resolutions. With this improvement the kinematic model can also describe the abrupt motion of targets. We have conducted experiments to evaluate the performance of the proposed method, using both a fish tracking method and state-of-the-art tracking methods.

***Index Terms*—** Video tracking, motion model, kinematic model, surveillance, long short-term memory network

1. INTRODUCTION

Consider a video taken by a fixed camera capturing multiple objects in the camera's field of view. We aim to automatically estimate each object's motion state, such as position and orientation, for every frame. Estimating an object's motion states in videos is known as object tracking, and that is one of key steps of video analysis [1]. The well known videos captured by fixed cameras are surveillance videos, in which cameras are usually fixed at a high position and take videos of a stationary scene. Fig. 1 shows the typical example of surveillance. There are plenty of literature that address the challenges of tracking objects in surveillance videos [2–5].

In addition, biological research also pose challenges to methods of tracking objects in videos captured with fixed cameras. The behavior patterns or interactions of groups of individuals, such as bird flocks [6], fish schools [7] and insect swarms [8] have aroused significant interest among scientists. The video tracking technology is the most effective gateway to acquire quantitative motion data for such research [9]. An example is shown in Fig. 1. The challenge of object tracking in these research 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 [6–10].

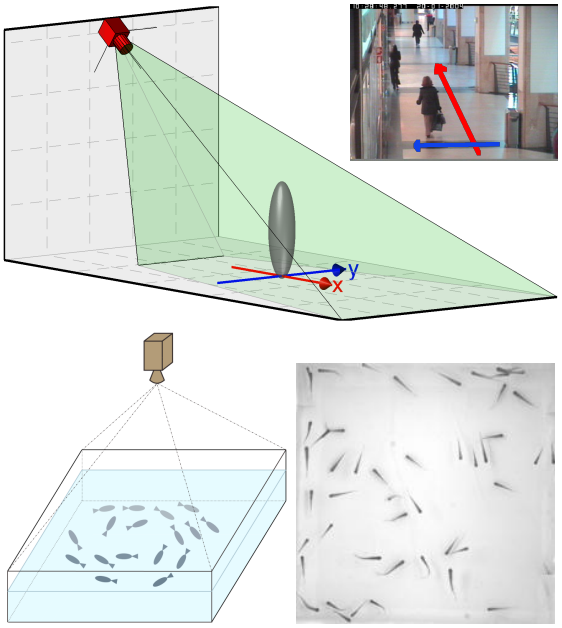

Fig. 1. Examples of typical applications and images. The upper panel shows the surveillance and the lower panel shows the biological research.

Motion model or dynamic model is crucial for object tracking [3], especially when a target takes abrupt motion. Here we do not rely on conventional Bayesian approaches (Section 2) for modeling the target's motion. In this work, we model the target's motion process by learning a long short-term memory (LSTM) network (Section 3). Modeling a target's motion process using the LSTM network is indeed modeling the kinematic model of the target. Moreover, the kinematic model is binding with the imaging system. That is, with unknown parameters of an imaging system videos captured by the cameras embed a target's kinematic pattern, and the LSTM network can learn the kinematic model using the tar-

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

get's sequential motion data obtained from these videos. The kinematic model serves as a discriminative model and can be applied in methods of tracking targets using videos captured by stationary cameras. This 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 LSTM network is a special kind of recurrent neural network (RNN), and it is introduced by Hochreiter and Schmidhuber [11] 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*. [12]. Though it is found that these variants are all about the same [13], 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.

However, the expressive ability of the kinematic model is mostly about the smooth motion at this point. In order to improve the expressive ability of the kinematic model, we introduce a variant LSTM network (Section 4). We partition the GRUs of the network into groups, and activate and update each group at different temporal resolutions. Inspired by the works of [14] and [15], this variant LSTM network, while it models a target's kinematic pattern, has the ability to describe the target's abrupt motion. We have developed a fish tracking system to evaluate the effectiveness of the proposed method (Section 6.1). And experiments also show the results that the propose method being integrated into state-of-the-art tracking methods improves their performance (Section 6.2).

2. CONVENTIONAL BAYESIAN APPROACHES

The conventional Bayesian approaches for modeling target's motion usually treat targets as dynamic systems [1, 4]. In order to analyze and make inference about a dynamic system, at least two models are required:

$$\begin{aligned} X_t &= f(X_{t-1}, v_{t-1}) \\ Z_t &= h(X_t, n_t) \end{aligned} \tag{1}$$

referred to as the dynamic model and the observation model, respectively. The function f is a function of the state X_{t-1}, and v_{t-1} is the i.i.d. noise, where t denotes the discrete-time index. The function h is a function of the hidden state variable X_t, and n_t is the i.i.d. noise. In particular, we seek filtered estimates of X_t base on the set of all available measurements $Z_{1:t} = \{Z_k, k = 1..t\}$ up to moment t.

Suppose that the required pdf $p(X_{t-1}|Z_{1:t-1})$ at moment $t-1$ is available. That is, the motion process up to moment $t-1$ is known. Under the first-order Markov assumption and the Bayes rule, we can get the well-known equation of Bayesian filtering

$$\begin{aligned} p(X_t|Z_{1:t}) &\propto p(Z_t|X_t) \int p(X_t|X_{t-1}) p^- dX_{t-1} \\ p^- &\equiv p(X_{t-1}|Z_{1:t-1}) \end{aligned} \tag{2}$$

From a Bayesian perspective, the tracking method is to recursively calculate some degree of belief of the state X_t at moment t, taking different values, given the observations $Z_{1:t}$ up to moment t. Thus, it is required to construct the posterior $p(X_t|Z_{1:t})$. It can be solved recursively in two steps: Predict a target's state at moment t using its state at moment $t-1$; update the target's state using its observation at moment t. However, there is only a conceptual solution in general, the posterior $p(X_t|Z_{1:t})$ cannot be determined analytically. Depending on the constraints and on the *a priori* information available on the system, the solution to the posterior density can be optimal like in the case of the well known Kalman filter (KF) [16], suboptimal in the case of the extended Kalman filter (EKF) [16], and the particle filter (PF) [17], and *etc*.

3. MODELING KINEMATIC PATTERNS

The Bayesian inference approach of modeling a target's motion is indeed been designed in the context of estimating the state of a system that changes over time using a sequence of noisy observations made on the system, such as radar signal processing [16]. The hidden system states and the noisy observations both embed the physical dimensions (*e.g*. meters) in real world. However, the motion states in image world have no physical dimensions or at least have no consistent physical dimensions while the Bayesian inference approach is introduced.

One of the reasons is that the principle of imaging systems is perspective projection. For instance, an object taking uniform motion in the camera's field of view (FOV) is captured in videos speeding up (toward the camera) or slowing down (away from the camera) in the image dimensions (*e.g*. pixels). That is, identical motions may be explained as abrupt motions (near the camera) or common behaviors (far from the camera) if the analysis relies on the image dimensions. The same motions may be explained as the same behaviors while the distance between the object and the camera was varying in a small interval in the period when motions have been taken, without regard to image distortion. Many tracking methods, which realize the effect of perspective projection, append the scale component to the motion state [1, 3]. However, appending a scale component to the motion state can not bring significant advantage into the video analysis.

Suppose many sequences of a target's motion data in image dimensions are available, and include the target's motion process at different conditions, such as along different directions and at different distances to the camera. We aim to model the target's motion pattern with respect to the imaging system. This model produces the same responses on a

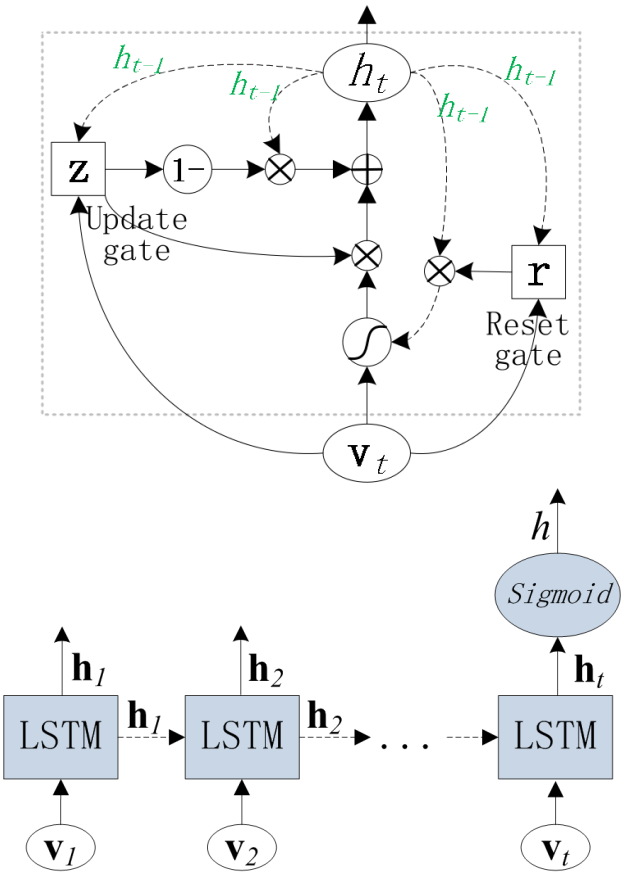

Fig. 2. Illustration of the LSTM network. The upper panel shows a GRU. The dashed line denote the recurrent connection. The lower panel shows the unrolled LSTM network.

target's motion process independent with the direction of target's movement or the distance between the target to the camera. The LSTM networks are proved to produce promising performance on analyzing sequential data [18], we prefer using it to model the motion process of a target.

While there are numerous LSTM variants, we adopt the implementation proposed by Cho *et al*. [12]. 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:

$$\begin{aligned}
\mathbf{z}_t &= \sigma\left(\mathbf{W}_{zv}\mathbf{v}_t + \mathbf{W}_{zh}\mathbf{h}_{t-1}\right) \\
\mathbf{r}_t &= \sigma\left(\mathbf{W}_{rv}\mathbf{v}_t + \mathbf{W}_{rh}\mathbf{h}_{t-1}\right) \\
\tilde{\mathbf{h}}_t &= \tanh\left(\mathbf{W}_{hv}\mathbf{v}_t + \mathbf{W}_{hh}(\mathbf{r}_t \odot \mathbf{h}_{t-1})\right) \\
\mathbf{h}_t &= (1 - \mathbf{z}_t) \odot \mathbf{h}_{t-1} + \mathbf{z}_t \odot \tilde{\mathbf{h}}_t
\end{aligned} \tag{3}$$

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.

The LSTM network models a target's kinematic pattern and can determine if a sequence of velocities is possible or not. The network contains a single LSTM layer followed by a sigmoid non-linear layer as depicted in the lower panel of Fig. 2. 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 motion state at moment $t-1$, they both are known. At moment t we have a motion state X_t which may be associated with the target, and thus the hypothetical motion velocity at moment t, $\tilde{\mathbf{v}}_t$, is computed as $\tilde{\mathbf{v}}_t = X_t - X_{t-1}$. Therefore, the hypothetical sequence of velocities is defined as

$$\{\mathbf{v}_{1:t-1}, \tilde{\mathbf{v}}_t\} \tag{4}$$

This sequence of velocities is the input of the LSTM network, and the output 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 $\tilde{\mathbf{v}}_t$ at moment t. That is, the learned LSTM network is indeed the target's kinematic model. We concentrate on using the velocity sequence because of the well-supported assumption that information obtainable from spatio-temporal data has considerably better accuracy than steady positional data.

4. IMPROVING MODEL'S EXPRESSIVE ABILITY

The LSTM network can model a target's kinematic model, and the kinematic model can determine the probability of a motion process. That is, according to the historical information about the characteristic of the target's motion pattern, we can determine if a motion process up to moment t is possible or not. However, as Fig. 3 shows, given the motion process up to moment t the kinematic model output a lower probability. The reason is the target has taken an abrupt motion in the period from moment $t-1$ to moment t. Therefore, the sequence of the target's velocities up to moment t is rejected by the kinematic model.

Why this happens? In order to compensate the bias induced by the imperfect detection or state estimation, the target's trajectory is smoothed before we compute the sequence of velocities. Therefore, most of samples in the training set probably satisfy the smooth motion assumption. That is, the network fits to model the smooth motion pattern. The discriminative ability of this model is comparable with the KF in real world. Inspired by the works of [14] and [15], which use delayed connections and units operating at different timescales to improve the simple RNN, we partition the GRUs of the LSTM network into groups. Different groups capture kinematic patterns at different temporal resolutions.

More formally, the GRUs of this variant LSTM network are partitioned into g groups $\{G_1, ..., G_g\}$. Each group

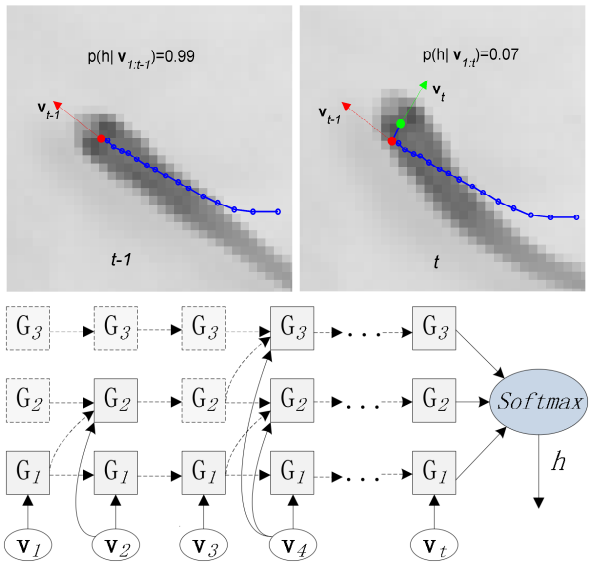

Fig. 3. Modeling abrupt motion using the LSTM network with grouping units. The upper panel shows a typical abrupt motion. The lower panel shows the unrolled LSTM network, dashed groups are non-activated groups at a certain time step and dashed arrows denote the recurrent direction.

$G_k, k \in \{1, ..., g\}$ is activated at a certain temporal resolution T_k. The choice of the set of temporal resolutions $T_k \in \{T_1, ..., T_g\}$ is arbitrary. From this view point, the conventional LSTM network is the network with just one group. Fig. 3 shows the unrolled structure of this variant LSTM network

At each time step t, only the group G_k that satisfy $t = n * T_k, n \in \mathbb{N}$ are activated, and $\mathbb{N}$ denotes the set of natural numbers. Here, we use the exponential series of periods, group G_k has the temporal resolution $T_k = 2^{k-1}$ [15]. Threfore, the group G_1 is the fasted one and can be activated at every time step, which works like the conventional LSTM network. The groups $G_k, k > 1$ are slowing down as k increases.

At moment t, the parameters and hidden states of units in group G_k are calculate in two cases:

(I) G_k **is activated**. When group G_k is activated at time step t, the parameters of GRUs of this group are calculated as:

$$\begin{aligned}
\mathbf{z}_t^k &= \sigma\left(\mathbf{W}_{zv}^k \mathbf{v}_t + \sum_{j=1}^{g} \mathbf{W}_{zh}^{k \leftarrow j} \mathbf{h}_{t-1}^j\right) \\
\mathbf{r}_t^k &= \sigma\left(\mathbf{W}_{rv}^k \mathbf{v}_t + \sum_{j=1}^{g} \mathbf{W}_{rh}^{k \leftarrow j} \mathbf{h}_{t-1}^j\right) \\
\tilde{\mathbf{h}}_t^k &= \tanh\left(\mathbf{W}_{hv}^k \mathbf{v}_t + \sum_{j=1}^{g} \mathbf{W}_{hh}^{k \leftarrow j} (\mathbf{r}_t^k \odot \mathbf{h}_{t-1}^j)\right) \\
\mathbf{h}_t^k &= (1 - \mathbf{z}_t^k) \odot \mathbf{h}_{t-1}^k + \mathbf{z}_t^k \odot \tilde{\mathbf{h}}_t^k
\end{aligned} \tag{5}$$

where $\mathbf{z}_t^k$ and $\mathbf{r}_t^k$ are the vectors of update gates and reset gates of group G_k at time step t respectively; $\mathbf{h}_t^k$ denotes the hidden state vector of group G_k at time step t.

(II) G_k **is non-activated**. When group G_k is non-activated at time step t, the parameters of its GRUs keep unchanged, and the hidden states are just copy of the previous states.

$$\mathbf{h}_t^k = \mathbf{h}_{t-1}^k \tag{6}$$

5. TRAINING SAMPLES AND TRAINING

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.

5.1. Augmentation of training set

The LSTM network is probably over-fitting on training samples. The easiest and most common method to reduce overfitting is to artificially enlarge the training set using label-preserving augmentations. 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

$$\{\mathbf{v}_{1:t-1}, \mathbf{v}_t + \nu \mid \nu \sim \mathcal{N}(\mu, \Sigma)\} \tag{7}$$

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 $\Sigma = 0.1 * \mathbf{I}$, 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 $\Sigma = \mathbf{I}$.

In a video tracking system, the velocity of a target is usually the hidden motion variable of the target. The observation of the target is usually an image patch (a blob), the blob's barycenter (the position) *etc.* which are all data being detected and being computed directly. That is, we can not obtain the velocities from observations directly. Fortunately, the velocity of a target at each moment can be computed using its states at consecutive moments, with help of the target's trajectory ground truth (smoothed version).

5.2. Training

In order to predict the the probability distribution of a motion sequence, the network is trained with back-propagation

through time (BPTT [19]) and the gradient-based optimization is performed using the Adagrad update rule [20].

To improve the efficiency of the training stage, we applied the mini-batch stochastic gradient descent strategy. The mini-batch size set to 10 samples for each iteration. The value of loss function is summed upon each mini-batch and then back-propagates from output layer to input layer.

Concerning the network presented in Section 4, its back propagation of errors is similar to the conventional LSTM network as well. The only difference is that the error propagates only from groups that were activated at moment t. The error of non-activated groups gets copied back in time (similarly to copying the activations of nodes not activated at moment t during the corresponding forward pass), where it is added to the back propagated error.

6. EXPERIMENTS

6.1. Experiments 1

To validate the effectiveness of the proposed method of modeling the kinematic model, we developed a tracking method for tracking multiple zebrafish in a water tank (Fig. 1 shows a typical image). The tracking method follows "tracking by detection" paradigm and is an updated version of the fish tracking method proposed by Qian *et al*. [7]. The method [7] proposed a fish head detection approach and adopted the Kalman filtering technique to model the dynamic motion of each target, and it serves as the *baseline* for the performance comparison. The performance of our fish tracking method is also compared with the popular fish tracking method, idTracker [10].

We create 3 dataset, *D1-D3*, at different populations (10, 26, and 49 respectively). Each dataset contains three video clips and each video clip consist of 2000 frames and each has been manually annotated the ground truth.

6.1.1. Results

Each experiment repeats three times, the reported results are the average score for each method. Table 1 shows the evaluation results, in which P denotes precision and R denotes recall. Unfortunately, idTracker [10] fails running on *D3* because of the large population.

6.2. Experiments 2

We have integrated the kinmatic model into the state-of-the-art video tracking methods, including the CPF [21], IVT [22], MIL [23], and CT [24]. The performance is evaluated on 3 videos: "Walking", "Walking2", and "Subway" [3].

The first phase of integration is to learn the kinematic model of targets for each video. Then the second phase is performed on each method with two steps: (i) Generate a hypothetic sequence of velocities of a target while a sample is going to associated with it. (ii) Multiply the probability produced by the kinematic model to the constraint adopted by each method to associate observations with targets.

Table 1. Performance comparison on zebrafish dataset

	Method	P	R	F1	Frag	IDS
D1	*baseline*	0.976	0.973	0.970	6.3	2.6
	idTracker	0.929	0.833	0.878	3.0	0.6
	Ours	0.991	0.986	**0.989**	**1.1**	**0.4**
D2	*baseline*	0.951	0.937	0.950	8.5	3.1
	idTracker	0.889	0.671	0.765	5.5	0.8
	Ours	0.970	0.962	**0.966**	**1.4**	**0.5**
D3	*baseline*	0.942	0.886	0.913	11.3	5.2
	idTracker	–	–	–	–	–
	Ours	0.963	0.931	**0.947**	**1.9**	**0.7**

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

6.2.1. Results

Fig. 4 shows the results of one-pass evaluation (OPE) [3]. Though parameters for each original method is tuned for the best performance, integrating the learned kinematic model improves the performance.

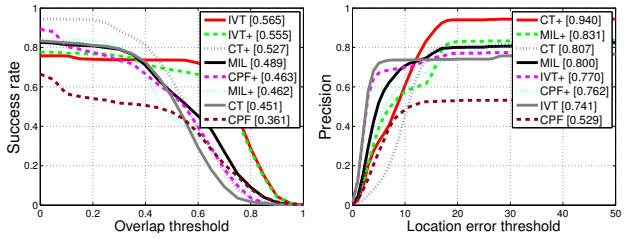

Fig. 4. The overall success plot and precision plot. Methods adapted by us are indicated by the postfix "+".

It shows the performance of CPF is significantly improved. The CPF compute the target's motion states using weighted particles [21]. Therefore, multiplying the probability produced by the kinematic model significantly improves the efficiency of computing each particle's weight. It also shows the success rate of CT is significantly improved. The CT compute the target's motion states using the sample with the maximal classification score [24]. Multiplying the kinematic probability helps to resolve samples while two or more samples having equivalent scores.

7. CONCLUSION

Using videos captured by fixed cameras, a sequence of a target's motion data (usually a sequence of velocities) probably encodes the target's kinematic pattern. We proposed in

this paper that the kinematic model can be learnt by a LSTM network. Videos captured by fixed cameras are common in biological research. The kinematic model is reusable for experiments on same kind of animals while the imaging system is fixed (for example, frame rate).

8. REFERENCES

[1] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Comput. Surv.*, vol. 38, no. 4, pp. 1–45, 2006.

[2] S. Calderara, R. Cucchiara, and A. Prati, "Bayesian-competitive consistent labeling for people surveillance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 354–360, 2008.

[3] Y. Wu, J. Lim, and M-H. Yang, "Online object tracking: A benchmark," in *CVPR*, 2013, pp. 2411–2418.

[4] A.W.M. Smeulders, D.M. Chu, R. Cucchiara, S. Calderara, et al., "Visual tracking: An experimental survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 7, pp. 1442–1468, 2014.

[5] S. Raghuraman, K. Bahirat, and B. Prabhakaran, "Evaluating the efficacy of rgb-d cameras for surveillance," in *ICME*, 2015, pp. 1–6.

[6] 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.

[7] Z-M. Qian, X.E. Cheng, and Y.Q. Chen, "Automatically detect and track multiple fish swimming in shallow water with frequent occlusion," *PLoS ONE*, vol. 9, no. 9, pp. e106506, 2014.

[8] K. Branson, A.A. Robie, J. Bender, P. Perona, and M.H. Dickinson, "High-throughput ethomics in large groups of *Drosophila*," *Nat Meth*, vol. 6, no. 6, pp. 451–457, 2009.

[9] AI. 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.

[10] A. Pérez-Escudero, J. Vicente-Page, R.C. Hinz, S. Arganda, and G.G. de Polavieja, "idTracker: tracking individuals in a group by automatic identification of unmarked animals," *Nat Meth*, vol. 11, no. 7, pp. 743–748, 2014.

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

[12] 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.

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

[14] S.E. Hihi and Y. Bengio, "Hierarchical recurrent neural networks for long-term dependencies," in *NIPS*, 1996, pp. 493–499.

[15] J. Koutnik, K. Greff, F. Gomez, and J. Schmidhuber, "A clockwork rnn," in *ICML*, 2014, vol. 32, pp. 1863–1871.

[16] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*, Artech House, 1999.

[17] 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.

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

[19] R.J. Williams and D. Zipser, "Gradient-based learning algorithms for recurrent networks and their computational complexity," in *Back-propagation: Theory, architectures and applications*, pp. 433–486, 1995.

[20] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *J. Mach. Learn. Res.*, vol. 12, pp. 2121–2159, 2011.

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

[22] D. Ross, J. Lim, R-S. Lin, and M-H. Yang, "Incremental learning for robust visual tracking," *Int. J. Comput. Vision*, vol. 77, no. 1-3, pp. 125–141, 2008.

[23] B. Babenko, M-H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1619 – 1632, 2011.

[24] K. Zhang, L. Zhang, and M-H. Yang, "Real-Time Compressive Tracking," in *ECCV*, 2012, pp. 864–877.