Detecting and Tracking of Multiple Mice Using Part Proposal Networks

Zheheng Jiang, Zhihua Liu, Long Chen, Lei Tong, Xiangrong Zhang, Xiangyuan Lan, Danny Crookes, Senior Member, IEEE, Ming-Hsuan Yang, Fellow, IEEE, and Huiyu Zhou

Abstract—The study of mouse social behaviors has been increasingly undertaken in neuroscience research. However, automated quantification of mouse behaviors from the videos of interacting mice is still a challenging problem, where object tracking plays a key role in locating mice in their living spaces. Artificial markers are often applied for multiple mice tracking, which are intrusive and consequently interfere with the movements of mice in a dynamic environment. In this article, we propose a novel method to continuously track several mice and individual parts without requiring any specific tagging. First, we propose an efficient and robust deep-learning-based mouse part detection scheme to generate part candidates. Subsequently, we propose a novel Bayesian-inference integer linear programming (BILP) model that jointly assigns the part candidates to individual targets with necessary geometric constraints while establishing pair-wise association between the detected parts. There is no publicly available dataset in the research community that provides a quantitative test bed for part detection and tracking of multiple mice, and we here introduce a new challenging Multi-Mice PartsTrack dataset that is made of complex behaviors. Finally, we evaluate our proposed approach against several baselines on our new datasets, where the results show that our method outperforms the other state-of-the-art approaches in terms of accuracy. We also demonstrate the generalization ability of the proposed approach on tracking zebra and locust.

Index Terms—Bayesian-inference integer linear programming (BILP) model, geometric constraint, mouse part detection, Multi-Mice PartsTrack dataset.

I. INTRODUCTION

IN NEUROSCIENCE research, animal models are valuable tools to understand the pathology and development of neurological conditions such as Alzheimer’s and Parkinson’s diseases [1]–[3]. Visual tracking of animals [4]–[6] is an essential task for many applications and has been successfully used in mice behavior analysis [7]. Scientific experiments in laboratories with mice need long-term observations by researchers and other parties. However, manually annotating long video recordings is a time-consuming task. Furthermore, manual documentation suffers from a number of limitations such as being highly subjective and having scarce replicability. Hence, there is an increasing interest in the development of systems for automated analysis of mice social behavior from videos [8]–[10].

Definitions of social behaviors can vary to some extent, from all the behaviors that occur when two or more animals are present in the scene [11], to only those behaviors in which one influences another [12]. Despite its definition, to automatically analyze social behaviors, discriminative features are often required: to record behaviors, to track the positions (or parts) of the participants, to identify individuals across time and space, and to quantify animal interactions. For automated behavior analysis, a reliable and smart tracking method is required to associate the detected behaviors with correct individuals. Moreover, accurate locations of mouse parts obtained from a stable tracking method enable the representation of interactions and allow for behavioral classification. Many applications, such as [13]–[15], analyze mouse behaviors using the tracking results of mice. Simultaneous tracking of two or more individuals poses a challenge in the computer vision community. The fact is that mice are mostly identical and highly deformable objects. In addition, social interaction between individuals makes the tracking mission even more complicated due to frequent occlusions. A popular method to track individuals during interactions is to label each subject with a unique marker, e.g., by bleaching [16] and color [17]. Also, Galsworthy et al. [15] monitor multiple mice in a single cage using radio transmitters buried under the skin and then record their activities by the detection coils. However, these systems are invasive for the tested subjects, and the labeling method in these systems very likely influences an individual’s
social behaviors as it frequently provides an olfactory and/or visual stimulus [18]. Therefore, people generally prefer markerless identification.

Several markerless approaches have been proposed to tackle this challenging problem. A common approach is to count on suitable foreground segmentation, which separates the extracted foreground pixels into several spatially connected groups using clustering algorithms such as expectation--maximization for Gaussian mixture models (e.g., [16], [19], [20]) or watershed segmentation (e.g., [14]). However, if two mice are spatially close, in contact or occluding each other, it can be difficult to separate them only based on foreground detection. Moreover, the location estimation of mouse parts is unreliable when occlusion occurs. De Chaumont et al. [7] make use of prior knowledge when tracking the mouse parts and develop a model that connects a set of geometrical primitives under physics-based principles and adapts the model’s parts to the moving mouse based on the defined physical engines. However, this type of methods requires sophisticated skeleton models which are hard-coded in the system, and thus limit the flexibility of the methods.

The above methods belong to the family of detection-free tracking methods, which requires manual initialization of mouse location in the first frame and then tracks mice and their parts in subsequent frames. Although these methods do not need pretrained object detectors, they are prone to drifts and identity switches (IDs). Recently, tracking-by-detection methods have become popular as they are efficient at handling deformation and occlusion. The essential idea is to first detect objects and then handle the data association problem over frames. This approach has been widely used for human tracking and pose estimation [21]–[25]. Most recently, Wang et al. [4] propose a motion-detection-based system to monitor small target such as insects.

To address the part tracking problem of multiple mice in the context of tracking-by-detection, we break the original problem into three interdependent and also associated subproblems. First, the positions of all the parts in each frame must be identified. Second, the detected parts must be assembled to form a physical mouse. Finally, the positions of the parts must be connected across image frames to generate mouse motion trajectories.

Detecting mouse parts in the first sub-problem is challenging due to the small size and subtle local interclass differences between image frames. To address this problem, we propose a mouse parts and body detection framework based on the features from deep neural networks. In fact, the use of deep neural networks has already obtained promising outcomes for part detection and pose estimation of humans in some challenging benchmarks [21], [22], [26]. This suggests that deep learning architectures can be applied to part detection of laboratory animals. For the second and third problems, we propose a Bayesian-inference integer linear program (BILP) model that resolves these problems by minimizing a joint objective function through Bayesian inference. We wish to handle target–candidate assignment and pair-wise part association within a single cost function. To solve the challenging problems in mouse part detection and tracking, we here design and evaluate a novel framework based on part association. Fig. 1 shows the flowchart of the proposed tracking system. In summary, we have the following novel contributions.

1) We propose a reliable detection framework that is efficient for identifying mouse body and parts and obtaining competitive performance on detection benchmarks.
2) With the detection candidates generated by the proposed detector, a BILP model is proposed to estimate the locations of all the mouse parts present in an image. The formulation is based on the targets’ assignment and pair-wise part association, subject to mutual consistency and exclusion constraints.
3) We formulate the parts’ localization as a Bayesian inference process that combines the output of the proposed detector with prior geometric models of mice. Unlike the previous work such as [7], our prior on the configuration of the mouse parts is not hard-coded but derived from our rich collection of the labeled training samples. In addition to geometric models, we also introduce motion cues to compensate for the missing appearance information.
4) Since there is no publicly available dataset that provides a foundation to quantitatively evaluate the Multi-Mice part tracking, we also introduce a new challenging Multi-Mice PartsTrack dataset that comprises a set of video recordings of two or three mice in a home cage with the top view. Several common behaviors are included in this dataset such as “approaching,” “following,” “moving away,” “nose contact,” “solitary,” and “pinning.” The true part and body locations of mice were manually labeled for each image frame.

II. RELATED WORK

In this section, we review the established approaches related to our proposed system. Section II-A reviews the existing methods for tracking multiple mice. Section II-B discusses the methods used for multipeople tracking.

A. Mouse-Based Tracking

Typically, the separation of foreground and background can be used as the first step of a multisubject tracking algorithm, e.g., [7], [14], [16], [19], [20], [27]–[29]. In mouse tracking, the knowledge of the foreground can be used to improve the accuracy of the tracking scheme.

Some existing approaches for tracking multiple mice focus on modeling the appearance of a mouse. For example, Hong et al. [29] first apply background subtraction and image segmentation from the top-view camera. They then fit an ellipse to each mouse in the foreground. Thus, the position and body orientation of each mouse are described by the fit ellipse. Twining et al. [27] find mice in a new image by fitting active shape models to different locations in the image and selecting the instance that best fits the models. Similarly, de Chaumont et al. [7] manually define shape models based on geometrical primitives and fit these models
Fig. 1. Architecture of the proposed tracking system. First, we propose an efficient and robust deep-learning-based mouse part detection scheme, which consists of a multistage part proposal network for generating high-quality part and body proposals and two network branches for generating part and body candidates. Subsequently, we propose a novel Bayesian-inference integer linear programming model that jointly assigns the part candidates to individual targets with necessary geometric constraints while establishing pair-wise association between detected parts.

In general, motion is a useful cue for dealing with occlusion. Model-based tracking approaches can benefit the identification by incorporating motion cues into the used model. A frequently adopted tracking method that combines both appearance and motion is based on particle filtering. For example, Pistori et al. [28] extend a particle filtering approach with certain variations on the observation model to track multiple mice from the top view. Branson and Belongie [19] present a particle filtering algorithm by tracking the contours of multiple mice and acquire the images of the targets from a side view. The ability of a particle filter to correctly track mice depends on how well the observation model works with mice. Due to the highly deformable shapes of mice, it is very difficult to explicitly model the entire mouse.

Another solution toward the problem of occlusions is to mark mice. Shemesh et al. [31], for example, apply fluorescent paints that light up in different colors under UVA light, and Ohayon et al. [16] dye the mouse fur with different patterns of strokes and dots. An alternative to visible markers are radio frequency identifiers (RFIDs), which are used by Weissbrod et al. [32] to identify individuals in combination with video data. However, these markers must be placed before recording and may be a potential distraction to mice. To deal with these issues, Giancardo et al. [14] use a high-resolution thermal camera to detect minor differences in body temperature. 3-D tracking can also disambiguate occlusions using depth cameras (e.g., Hong et al. [29]) or multiple video cameras at different viewpoints (e.g., Sheets et al. [33]), which is a challenging solution requiring additional equipment and calibration work. The summary of mouse-based tracking is shown in Supplementary A.

B. People-Based Tracking

Human tracking is also a well-studied topic in computer vision. Some of the techniques developed for human tracking may be applied for mouse tracking. Most filter-based tracking methods such as Kalman filter, particle filter, and correlation filter are well-suited for online applications due to their recursive nature. Some approaches of this category focus on tracking a single target by model evolution [34], [35].
Some of the others aim at training better object classifiers [36] or learning better target representations [23]. However, these methods cannot guarantee a global optimum as they conduct tracking on each target individually.

To alleviate the problems of filter-based tracking methods, tracking-by-detection methods have been used in many applications. The fundamental idea is to first detect objects in each frame and then address the data association problem. Recent approaches in this category have been focused on improving the performance of the developed object detectors or designing better data association techniques to improve the tracking performance. For example, Shu et al. in [37] propose an extension to deformable part-based human detector [38] and use the visible part to infer the state of the whole object. A number of approaches rely on data association methods such as the Hungarian algorithm [39], [40], global objective function optimization [41]–[45], and recurrent neural network [46]. Xiang et al. [39] propose to formulate tracking as a Markov decision process with a policy learned on the labeled training data. They aim to learn similarity scores between targets and detections, which are used in the Hungarian algorithm to obtain the targets’ assignment. In [41], the authors propose a continuous formulation that analytically models mutual occlusions, dynamics, and trajectory continuity, but uses a simple support vector machine (SVM) detector in their appearance model. Hamid Rezatofighi et al. [45] propose an efficient approximation of joint probabilistic data association (JPDA) to reduce computational complexity in data association. Recent approaches have formulated data association as a network flow problem and then seek the solution by optimizing a global objective function. Zhang et al. [42] show that a promising solution toward the network flow problem can be found in polynomial time and is highly efficient in practice. Pirsiavash et al. in [43] also adopt network flows and use dynamic programming to search for a high-quality suboptimal solution. Shirlit et al. [44] propose a multicommodity network flow to incorporate the appearance consistency between groups of people. Milan et al. [46] propose data-driven approximations to deal with the data association problem using long short-term memory (LSTM) networks while using RNNs for targets’ prediction and updating. Compared with these methods, we consider targets’ assignment and pair-wise part association in our problem formulation. Also, we use Bayesian inference to construct correct correspondences between the proposals and the measurements. Moreover, we introduce a geometric constraint in Bayesian inference to help reduce the ambiguity caused by the nearby targets with similar appearance.

C. Pose Estimation

Recent methods for multiperson pose estimation can be broadly divided into two categories, i.e., top–down and bottom–up. The top–down methods [47]–[49] aim to first detect people in the image and then estimate body pose independently. For example, Pishchulin et al. [47] follow this paradigm by incorporating a trained deformable part model (DPM) into a pictorial structure framework. Gkioxari et al. [48] use a person detector modeled after poselets, which is more robust to occlusions. The very recent Mask-RCNN method [49] extends the standard Faster R-CNN by adding a branch for predicting object masks, which shows satisfactory results in person keypoint estimation. However, such approaches are applicable only if people appear well-separable. Moreover, these pose estimation methods always output a fixed number of body joints without accounting for occlusion. The bottom–up methods first detect body parts instead of full persons, and then associate these parts to individual instances. Pishchulin et al. [21] and later Insafutdinov et al. [22] generate body parts’ hypotheses using CNN-based part detectors, and then jointly solve the problems of people detection and pose estimation by formulating these problems as part grouping and labeling via an integer linear program. Cao et al. [50] combine the unary joint detector modified from [51] with a part affinity field and greedily generate person instance proposals. In view of the ability of bottom–up approaches, they are more robust to occlusion as they assemble parts to the corresponding individuals. However, these methods do not incorporate any part tracking and therefore only work on single images.

In the case of mice, previous studies commonly use the top–down scheme. For example, Cootes et al. [52] and Twining et al. [27] find mice in a testing image by fitting active shape models to different locations in the image and selecting the instance that best fits the models. Similarly, de Chaumont et al. [7] manually define shape models based on geometrical primitives and fit these models to the image using a physical engine. Although these methods work effectively, the flexibility of such methods is limited due to the requirement of sophisticated skeleton models. Most recently, based on previous designs of neural networks for human pose estimation, Mathis et al. [53] develop their pose estimation model by deploying transfer learning that can reduce the number of required training examples. Pereira et al. [54] incorporate the previous work with a graphical user interface for labeling of body parts while training the network. However, these animal pose estimation systems are based on the single-image pose estimation methods.

III. PROPOSED MOUSE PARTS AND BODY DETECTION

As stated before, a powerful part and body detector supplies the solid foundation for successful body tracking. Driven by the recent development in deep neural networks with “attention” mechanisms, we here propose a novel mouse part and body detection framework which consists of two components: a multistage part proposal network (PPN) that generates suitable proposals of mouse parts and body, followed by a fully connected network that classifies these proposals using higher resolution convolutional and local binary pattern (LBP) features extracted from the original images.

A. Part Proposal Network (PPN) for Mouse Parts and Body Detection

Detecting mouse parts is nontrivial due to their small size and subtle local interclass differences across images. To develop a strong detector, we first create a proposal generation network that directs the downstream classifier
where to look. The region proposal network (RPN) in the Faster R-CNN [55] and selective search [56], [57] are two state-of-the-art methods for object proposal generation. These methods were developed as a class-agnostic proposal generator aiming to create the bounding boxes that may contain objects. Without specific knowledge, these region proposals may not be accurate. DPM [38] is another frequently used algorithm to generate small object proposals. DPM and its variants are a class-specific method, which is based on the histogram of oriented gradient (HOG) features. In addition, most negative examples generated by these methods belong to the background. Unlike these methods, we adopt a multistage architecture to perform effective hard negative mining. Suppose we can have finer region proposals, and the accuracy of the proposed classifier can be further improved. In this section, we describe the proposed multistage PPN as shown in Fig. 2, which can generate high-quality object proposals from convolutional feature maps.

We first adopt the VGG-19 net [58] as the backbone network, which is pretrained on the ImageNet dataset [59]. The input image is first analyzed by a convolutional network initialized by the first 15 layers of VGG-19 net [58] and fine-tuned for the region proposal task to generate a set of feature maps as the input to our proposed PPN. We use a partial VGG-19 network and add another two trainable convolutional networks after the VGG-19 network, aiming to extract discriminative features and generate feature maps \( F \) for subsequent proposal generation tasks. To produce region proposals, we construct a multistage class-specific proposal generation network. In each stage, we create a new network including an intermediate \( 3 \times 3 \) convolutional layer sliding on feature maps \( F \), followed by two sibling \( 1 \times 1 \) convolutional layers for classification and bounding box regression, respectively. For each sliding window in the \( 3 \times 3 \) convolutional layer, we simultaneously predict multiple region proposals (called root boxes) in the original image. In our method, the multiple scales and aspect ratios of the root boxes are different for individual objects, dependent on the object size in the image.

The classification layer provides the confidence scores of the root boxes, as used to mine and add negative root boxes to a mini-batch of the next stage. Each mini-batch contains several positive and negative example root boxes from the image. Initially, all the positive root boxes in the image are selected, and the negative root boxes are randomly sampled. In each stage, hard negative root boxes are used to replace the previous negative root boxes in the mini-batch. Here, we define \( D_{\text{pos}}^z \subset D^z \) and \( D_{\text{neg}}^z \subset D^z \), consisting of all the positive and negative root boxes, respectively, in a mini-batch at stage \( z \). \( D^z \) is a universal set containing all the positive, negative, and unlabelled root boxes in the image.

With the above definitions, we infer a multitask loss function for the image at stage \( z \) as follows:

\[
L^z = -\frac{1}{N} \sum_j \left( \sum_{i \in D_{\text{pos}}^z} \log S_j^z(i) + \sum_{i \in D_{\text{neg}}^z} \log \hat{S}_j^z(i) \right) + \sum_{i \in D_{\text{pos}}^z} L_{\text{reg}}(B_j^z(i) - \hat{B}_j^z(i)).
\]  

Here, \( S_j^z(i) \) denotes the predicted probability of the root box \( i \) being a positive object of the mouse body \( p_0 \) or a mouse part \( p_j \in \{p_j\}_{j=1}^z \) at stage \( t \). The negative predicted probabilities are represented as \( \hat{S} \). For the regression loss \( L_{\text{reg}} \), we use the robust loss function (smooth \( L_1 \)) defined in [55]. \( B_j^z(i) \) is a vector which represents the four parameterized coordinates of the predicted bounding box associated with the root box \( i \) of \( p_j \) at stage \( z \), and \( \hat{B}_j^z(i) \) is that of the ground-truthed box. All the terms are finally normalized by the mini-batch size (i.e., \( N = 256 \)).

To utilize our RPNs trained at each stage, we assign the weights to the output of the classification–regression networks (represented by \( E^z \)) based on their pseudo-loss

\[
E^z = \frac{1}{\text{num}(D^z)} \sum_j \left( \sum_{i \in (D^z - D_{\text{pos}}^z)} S_j^z(i) + \sum_{i \in D_{\text{pos}}^z} S_j^z(i) + \sum_{i \in D_{\text{neg}}^z} \hat{S}_j^z(i) \right)
\]  

\[
\alpha^z = \frac{1 - E^z}{\sum_{z=1}^Z (1 - E^z)}
\]

where \( N \) is a number to count the number of the root boxes. After \( Z \) stages of the training, the confidence scores \( S_j \) and the bounding boxes \( B_j \) of the root boxes with respect to object \( p_j \) have the following form: \( S_j = \sum \alpha^z S_j^z \) and \( B_j = \sum \alpha^z B_j^z \).

Fig. 3 shows the refinement of the confidence maps across different stages. To reduce the number of proposals for efficiency, we remove all the bounding boxes that have an intersection-over-union (IoU) [55] ratio over 0.7 with another bounding box that has a higher detection confidence. Supplementary B summarizes the training steps of our PPN.

**B. Part Candidates’ Generation**

With the proposals generated by PPN, inspired by the Faster R-CNN layer [55], we also crop the proposals and extract fixed-length features from the feature maps. However, the feature maps of the faster R-CNN are of low resolution.
for detecting small objects (e.g., mouse heads and tails). Generally, a small proposal box is mapped onto only a small region (sometimes \(1^*1^*\)) at the last pooling layer. Such small feature map lacks discriminative information, and thus degrades the following classifier. We address this problem using the pooling features from shallower layers, and by additionally extracting texture features (e.g., LBP features) from the original image. For classifier training, we construct the training set by selecting the top-ranked 200 proposals (and ground truths) of each image for each class. At the testing stage, we use a trained classifier to classify all the proposals in an image, and then perform nonmaximum suppression independently for each class to obtain part candidates.

IV. PARTS’ TRACKING OF MULTI-MOUSE

To achieve successful body and part tracking for all the mice freely interacting in a home cage, we here propose a BILP formulation. Solving this BILP will yield an optimal solution to provide the locations of mouse parts (i.e., head and tail) in continuous videos, while fulfilling mouse part association over time. Then the mouse body is obtained by connecting tracked mouse parts.

A. Problem Formulation

Given a video sequence \(\mathcal{F}\) containing \(N^* = \{1, \ldots, N\}\) parts, we generate a set of detection candidates \(M^*_t = \{1, \ldots, M\}\) using the proposed detector at time \(t\). For the task of multimouse part tracking, our goal is to jointly address two problems: 1) successful assembling of the detected parts to represent each individual mouse and 2) correct alignment of the mouse parts to form motion trajectories.

For the benefit of representation, similar to [21], [22], we define two binary vector spaces \(\mathbb{B}^{M_s \times M_t}\) and \(\mathbb{B}^{N_s \times (M_t+1)}\) and encode these two problems through the two binary vectors of \(\gamma_t \in \Gamma_t \subseteq \mathbb{B}^{M_s \times M_t}\) and \(\theta_t \in \Theta_t \subseteq \mathbb{B}^{N_s \times (M_t+1)}\)

\[
\gamma_t \in \Gamma_t = \left\{ s^m, s^m' \in \{0, 1\} \quad m, m' \in M^*_t \right\}
\]

\(\theta_t \in \Theta_t = \{ a^n_m, a^n_m' \in \{0, 1\} \quad m \in M^*_0, n \in N^* \}\) (5)

where \(s^m_m\) suggests whether the detected parts \(m\) and \(m'\) can be used to form an individual mouse, i.e., \(s^m_m = 1\) with \(m, m' \in M^*_t\).

Our approach to the second problem is to assign correct identities to detection candidates. In an easy case, the relationship between the targets and the detection candidates is bijective, which means that all the tracked targets are also observed and each measurement was generated by the tracked targets. This is an unrealistic scenario as it does not consider false detection and targets’ occlusions. To solve this assignment problem, we introduce a placeholder for a “fake” (or missed) detection and also define \(m \in M^*_0 = \{0, 1, \ldots, M\}\) to include “fake” detection and all the detection candidates. Similar to \(s^m_m\), a binary variable \(a^n_m\) represents that the detection candidate index \(m \in M^*_0\) which is generated by target \(n \in N^* = \{1, \ldots, N\}\) at time \(t\). Here, we have \(N^* (M + 1)\) paired with \(a^n_m = 1\) if candidate \(m\) is assigned to target \(n\) and \(a^n_m = 0\) otherwise. By definition, \(\Gamma_t\) and \(\Theta_t\) consist of all possible solutions at time \(t\).

To ensure that every solution is physically available, we add several constraints that 1) each candidate (except for the fake hypothesis \(m = 0\)) is assigned to at most one target, i.e., (a) for \(\forall m \in M^*_t, \sum_{n=1}^{N} a^n_m \leq 1\); 2) each target is uniquely assigned to a candidate, i.e., (b) for \(\forall n \in N^*, \sum_{m=0}^{M} a^n_m = 1\); and 3) only the candidates and targets of the same type (e.g., head to head or tail to tail) can be corresponded, except for a fake hypothesis, i.e., (c) for \(\forall m \in M^*_t, \forall n \in N^*, a^n_m = 0\) if \(\hat{m} \neq \hat{n}\). To ensure that feasible solutions \(\theta_t \in \Theta_t\) and \(\gamma_t \in \Gamma_t\) result in a valid target assignment and pair-wise part association, we need to apply an additional constraint that two detected parts are connected if and only if both the detections are assigned to different targets: (d) for \(\forall m, m' \in M^*_t, s^m_m \wedge s^{m'}_{m'} \leq \sum_{n=1}^{N} a^n_m \wedge s^{m'}_{m'} \leq \sum_{n=1}^{N} a^n_{m'}\).

B. Bayesian-Inference Integer Linear Program (BILP)

We jointly resolve the targets’ assignment and part association problems by minimizing the following cost function:

\[
\mathcal{L}_t = \min_{\theta_t \in \Theta_t, \gamma_t \in \Gamma_t} \left( -\log(p(\theta_t)) + \log(p(\gamma_t)) \right) \quad (6)
\]

where the values of \(\theta_t\) and \(\gamma_t\) which attain the minimum value of (6) are the maximum likelihood of target assignment and part association at time \(t\).

1) Bayesian Inference: With regard to the target tracking problem, the assumption of Markov property in the target state sequence is frequently used in extended Kalman filter [60], particle filtering [61], [62], and MCFPJD [63] or MDP [39]. These approaches are appropriate for the task of online tracking due to their concern on region neighborhood. We also use this assumption to resolve the target assignment problem. Let \(V_t = \{v^1_t, v^2_t, \ldots, v^N_t\}\) denote the states of all the \(N\) targets. State vector \(v^k_t\) contains dynamic information of the position and velocity of the \(j\)th target at time \(t\). \(D_t = \{d^1_t, d^2_t, \ldots, d^M_t\}\) represents the positions of all the \(M\) part detection candidates at time \(t\). The assumption of Markov property is twofold.

First, the kinematic state dynamics follow a first-order Markov...
where $A$ and $C$ are our state transition (or motion) and observation models, respectively. $\hat{v}_{t-1}$ and $\Sigma_{t-1}$ are the mean value and the error covariance matrix at state $v_{t-1}$, respectively. $\mathcal{N}$ is the normal distribution. $\Omega$ and $\gamma$ represent the covariances of the process and observation noise with the mean value of zero, respectively. $p(\hat{d}_t^n)$ is calculated by marginalizing $p(\hat{d}_t^n|v_{t-1}^n)$ and $p(o_t^n|v_{t-1}^n)$.

$$p(\hat{d}_t^n) = \int p(\hat{d}_t^n|v_t^n) \int p(o_t^n|v_{t-1}^n) \, dv_{t-1} \, dv_t^n. \quad (8)$$

However, in a realistic case, Markovian assumption-based methods carry the associated danger of drifting away from the correct target, as they treat the targets as conditionally independent of one another. This risk can be mitigated by optimizing data assignment and using prior knowledge, as shown in [64]–[66]. In our approach, when dealing with the data association problem, we use a geometric model as the prior knowledge. Given $K$ training samples centered at a single mouse, let $X_k = \{x_1^k, x_2^k, \ldots, x_K^k\}$ (where $k = 1, \ldots, K$) denote the locations of the mouse parts in the $k$th training sample. $x_i^k$ is the location of the $i$th part in the $k$th training sample. Let $X_k$ refer to a geometric template and suppose that the location of the mouse parts in a feasible solution $\theta$ is generated by one of our geometric templates $X_k$. We then expand $p(\theta_t)$ as follows:

$$p(\theta_t) = \sum_{k=1}^{K} p(\theta_t|X_k) p(X_k) \quad (9)$$

where our collections of $K$ training samples have been introduced into the calculation of $p(\theta_t)$ before it can be marginalized out.

By conditioning on the geometric template $X_k$, the locations of the mouse parts can be treated as conditionally independent of one another. As the association probability of fake detection is not constrained by our geometric template, we rewrite the first term of (9) as follows:

$$p(\theta_t|X_k) = \prod_{n \in N^*} p_t(a_{t}^{n}) \prod_{m \in M^*, n \in N^*} p_t(X_k|a_{t}^{m}) p_t(a_{t}^{m}) p_t(X_k) \quad (10)$$

where

$$p_t(a_{t}^{m}) \propto \begin{cases} ((1 - \hat{p}(d_m^n)) \beta)^{a_{t}^{m}}, & \text{if } m = 0 \\ \hat{p}(d_m^n) * p_t(a_{t}^{m} = d_m^n)^{a_{t}^{m}}, & \text{otherwise} \end{cases} \quad (11)$$

where $\hat{p}(d_m^n)$ is the detection probability of the $m$th part detection candidate at time $t$, $p(\hat{d}_t^n)$ is a probability distribution obtained from our motion model, and $p_t(a_{t}^{m})$ is an assignment probability representing that the detection candidate index $m \in M^*$ is generated by target $n \in N^*$ at time $t$. If $m = 0$, which means the target $n$ is missing, $p_t(a_{t}^{m})$ can be estimated using an empirical parameter $\beta$ of false detection density.

Combining (7)–(10) yields

$$p(\theta_t) = \left( \sum_{k=1}^{K} \prod_{m \in M^*, n \in N^*} p_t(X_k|a_{t}^{m}) \right) \left( \prod_{m \in M^*, n \in N^*} \left( \hat{p}(d_m^n) * p_t(a_{t}^{m} = d_m^n)^{a_{t}^{m}} \right) \right) \quad (12)$$

where $p(X_k|a_{t}^{m} = 1)$ represents that if candidate $m$ is assigned to target $n$, i.e., the location of target $n$ is known, how well the part location in the $k$th geometric template fits the location of this target. Computing the sum as shown in (12) is challenging due to a large amount of training samples. However, we note that if $p(X_k|a_{t}^{m} = 1)$ is very small, it will be unlikely to contribute much to the final outcome. Thus, we mainly consider geometric templates with large $p(X_k|a_{t}^{m} = 1)$. To achieve this, we first find the $O$ nearest neighbors of the body in the training samples to match $b_m \in B = \{b_1^1, b_2^1, \ldots, b^n^k\}$ in the convolutional feature space, where $B$ is a set of body candidates with $Y$ elements which are generated by the proposed detector at time $t$ and $b_m$ is the top-scoring body candidates which contain the candidate $m$. Then the probability distribution $p(X_k|a_{t}^{m} = 1)$ of the $m$th part can be estimated by fitting a Gaussian model to the location of this part in the O body neighborhoods. Different from most top-down methods [7], [27], [52] that first track the mouse and then estimate the locations of mouse parts, our approach establishes geometric relationships between the mouse parts and the mouse body within our probabilistic model and tracks the mouse parts directly based on our probabilistic model, and therefore our approach tolerates partial occlusion of the mouse body. The selection of the body targets can be formulated as follows:

$$b_m = \begin{cases} \arg\max_{b_t \in B} \sigma_b(b_t), & \text{if } \max_{b_t \in B} \sigma_b(b_t) \neq 0 \\ \text{null}, & \text{otherwise} \end{cases} \quad (13)$$

where

$$\sigma_b(b_t) = \begin{cases} \hat{p}(b_t^m), & \text{if region of candidate } m \text{ falls} \\ \text{outside region of } b_t^m \text{ by at most } v \text{ pixels} \\ 0, & \text{otherwise} \end{cases} \quad (14)$$

where $\hat{p}(b_t^m)$ is the detection probability and null is an empty value. In our experiments, we set $v = 10$. With these in hand, we approximate the first term in (12) as:

$$\sum_{k=1}^{K} \prod_{m \in M^*, n \in N^*} p_t(X_k|a_{t}^{m}) \approx \sum_{k \in O} \prod_{m \in M^*, n \in N^*} p_t(X_k|a_{t}^{m}) \quad (15)$$

where the sum is now only taken over those $k \in O$. The term $p_t(X_k|a_{t}^{m})$ is a 2-D Gaussian distribution centered at the
location \( x_{m,n}^k \) with a user-defined variance. According to the proof of Supplementary C, we obtain

\[
\sum_{k \in O} \prod_{m \in M^O, n \in N^O} p_t(X_k | a_n^m) = \prod_{m \in M^O, n \in N^O} \sum_{k \in O} p_t(X_k | a_n^m). 
\]

(16)

If \( O \neq \emptyset \), we alternatively fit a Gaussian model to the location of each part in the \( O \) body neighbors and use this equation to estimate the sum as shown in (16), given the type (e.g., head or tail base) of the candidate \( m \). Otherwise, we treat the sum in (16) as a very small variable value of \( \varepsilon = 0.0001 \).

We reformulate (12) as follows:

\[
p(\theta_i) \approx \prod_{n \in N^*} (p')^{a_n^m} \times \prod_{m \in M^*, n \in N^*} \left( \Delta(a_n^m) * \hat{p}(d_m^m) * N(d_m^m; C * A * \delta_n^m), \right) \times \left( A * \Sigma_{i=1}^n A^T + \Omega \right) \left( C^T + \gamma \right) \]

(17)

where \( \Delta(a_n^m) \) defines a scoring function, estimated by our proposed geometric model given the location of the candidate \( m \) in the body target.

2) Parts’ Association Model: To improve the multitarget tracking accuracy, several methods ([67]–[69]) have also presented group models, in which each object is considered as having the relationship with other objects and surroundings. These models can alleviate performance deterioration in crowded scenes. Similarly, here we propose a new part association model to establish the pair relationship between detection candidates. In our approach, the pair-wise term \( p(\gamma) \) shown in (6) is defined as follows:

\[
p(\gamma_i) = \prod_{m,m' \in M^*} \eta(s_{m'}^m) \]

(18)

where \( \eta(s_{m'}^m) = p_t(d_m^m, d_{m'}^m) / (1 - p_t(d_m^m, d_{m'}^m)) \). \( p_t(d_m^m, d_{m'}^m) \in (0, 1) \) corresponds to the probability that detections \( d_m^m \) and \( d_{m'}^m \) in a frame \( t \) belong to the same mouse. Note that \( \log(\eta(s_{m'}^m)) \) is negative if \( p_t(d_m^m, d_{m'}^m) > 0.5 \). Since we aim to minimize the cost function shown as (6), a smaller \( \log(\eta(s_{m'}^m)) \), i.e., a pair detection with a higher parts’ association probability, is preferred. The parts’ association probability \( p_t(d_m^m, d_{m'}^m) \) depends on the types \( m \) and \( m' \) of the part detections \( d_m^m \) and \( d_{m'}^m \). If \( m \neq m' \), we define \( p_t(d_m^m, d_{m'}^m) = IoU(d_m^m, d_{m'}^m) \). This means that two close detections denoting the same part belong to the same mouse. If the connection between the two detections of the same type exists, the detections are merged with the weighted mean of the detections, where the weights are equal to \( \hat{p}(d_m^m) \). If \( m \neq m' \), we estimate the posterior probability \( p_t(d_m^m, d_{m'}^m) = p(s_{m'}^m = 1 | \text{dis}(d_m^m, d_{m'}^m)) \) based on the Euclidean distance between the two detections. Assuming the prior probability \( p(s_{m'}^m = 1) = p(s_{m'}^m = 0) = 0.5 \), then we have

\[
p(s_{m'}^m = 1 | \text{dis}(d_m^m, d_{m'}^m)) = \frac{p(\text{dis}(d_m^m, d_{m'}^m) | s_{m'}^m = 1)}{p(\text{dis}(d_m^m, d_{m'}^m) | s_{m'}^m = 1) + p(\text{dis}(d_m^m, d_{m'}^m) | s_{m'}^m = 0)} 
\]

(19)

where \( p(\text{dis}(d_m^m, d_{m'}^m) | s_{m'}^m = 1) \) denotes the positive likelihood where two detections come from the same mouse. We estimate it by creating a 1-D Gaussian distribution of \( \text{dis}(d_m^m, d_{m'}^m) \) from positive training examples. The negative likelihood \( p(\text{dis}(d_m^m, d_{m'}^m) | s_{m'}^m = 0) \) is estimated in a similar way using negative training examples. With the above definition, \( \hat{m} = m' \) pair-wise terms facilitate merging the mouse part candidates with the same type candidates. \( \hat{m} \neq m' \) pair-wise terms allow us to connect the mouse part candidates within a valid mouse pose.

3) Optimization: By introducing (9) and (18), we rewrite the cost function of (6) as follows:

\[
\mathcal{L} = \min_{\Lambda} \left( \sum_{n \in N^*} a_n^0 \log \beta + \sum_{m \in M^*, n \in N^*} a_n^m \log(\Delta(a_n^m) \hat{p}(d_m^m)) \times \left( N(d_m^m; C * A * \delta_n^m, C * \Sigma_{i=1}^n A^T + \Omega \right) \left( C^T + \gamma \right) + \sum_{m,m' \in M^*} s_{m'}^m \log p_t(s_{m'}^m) \right) 
\]

(20)

s.t. (a), (b), (c), (d) in Section IV-A.

Finally, the problems of targets’ assignment and part association are jointly reformulated as

\[
\min_{\Lambda \in (0,1)^l} \Phi^T \Lambda \text{ s.t. } Ay \leq h
\]

(21)

where \( \Lambda = [\theta, \gamma]^T \) is a binary vector of length \( J = N(M+1) + M \cdot M \), and \( \Phi = [\Phi_1, \ldots, \Phi_J]^T \) is the cost vector with \( \Phi_j = -\log(\beta) \) if \( 1 \leq j \leq M \), \( \Phi_j = -\log(\Delta(a_n^m) \hat{p}(d_m^m)) \) if \( M < j \leq N(M+1) \) and \( \Phi_j = -\log(p_t(s_{m'}^m)) \) if \( N(M+1) < j \leq N(M+1) + M \cdot M \). \( A \) and \( h \) stand for a constraint matrix and linear equality constraints, respectively, resulting in constraints (a)–(d).

To solve (21), we first relax the constraint \( s_{m'}^m \in \{0, 1\}, \) \( a_n^m | a_n^m = 0 \) to \( 0 \leq s_{m'}^m \leq 1, 0 \leq a_n^m \leq 1 \). Then, we use a branch-and-bound [70]-based global optimization method which is summarized and shown in Supplementary E. With the convergence analysis of the proposed algorithm shown in Supplementary E, it is proven that the algorithm converges to the global optimum.

The time complexity to solve (21) is \( O(A^{1.5} B^2) \) [71], where \( B = N \cdot (M+1) + M \cdot M \) denotes the number of the variables and \( A = N + M + 1 + M \cdot M \) denotes the number of the constraints.

C. Targets’ State Update

After having obtained an optimal solution of (21), we associate each target with its corresponding detection candidate. If a target is assigned to a fake detection candidate, we assume that the target is occluded or lost. To guarantee tracking identification of each mouse part, we do not discard these targets like JPDAm [45] and MDP [39] (see their tracking results in Fig. 88 and 99). Instead we predict the target state using the motion model \( \delta_n^m = A * \delta_{n-1}^m \) till the target is detected again. If a target is assigned to a true detection candidate, our goal is to maximize the posterior probability of each target.
Algorithm 1 Algorithm for Online Mouse Part Tracking

**Input:** a video sequence $F$, the proposed mouse part and body detector, $C$ training samples of a single mouse.

**Output:** bounding boxes of all the targets.

1. for $t = 1$ to $T$ do
2. Generate a set of part and body detection candidates using the proposed mouse part and body detector on the frame $F_t$.
3. Fit a geometric model with appearance features to $O$ neighbors of the top-scoring body candidate, which contain candidate $m$.
4. Compute $\Delta(a^n_m), N(d^n_m)$ and $p_t(s^n_m)$ in Eq. (17);
5. Construct constraint matrix $A$ in Eq. (21) using constraint formulas (a), (b), (c) and (d);
6. Construct cost vector $\Psi$ in Eq. (21);
7. Obtain best solution of $\theta$ and $\gamma$ by solving Eq. (21) in Alg. S2;
8. Update the state of each target using Eq. (25);
9. Obtain bounding boxes of all the targets dependent on the best solution of $\theta$ and $\gamma$;
10. return bounding boxes of all the targets.

state $\hat{v}_t^n$ given the detection $d_t^n$. The updated formula can be represented as follows:

$$\hat{v}_t^n = \arg\max_{\hat{v}_t^n} p(v_t^n|d_t^n).$$

To compute the posterior probability $p(v_t^n|d_t^n)$, we first define

$$P = A * \Sigma_{n-1} * A^\top + \Omega.$$  

Following Bayes’ rule, we derive the posterior probability as follows:

$$p(v_t^n|d_t^n) \sim N\left((P^{-1} + C^T\chi^{-1}C)^{-1}(C^T\chi^{-1}d_t^n + P^{-1}\hat{v}_t^n), (P^{-1} + C^T\chi^{-1}C)^{-1}\right).$$

where parameters $N$, $C$ and $\chi$ have been defined as those of (7).

Afterward, the estimated state $\hat{v}_t^n$ is the mean of Gaussian distribution

$$\hat{v}_t^n = (P^{-1} + C^T\chi^{-1}C)^{-1}(C^T\chi^{-1}d_t^n + P^{-1}\hat{v}_t^n).$$

Note that we solve the part tracking problem of multi-mice in an online manner, which is illustrated in Supplementary F and is summarized in Algorithm 1.

V. EXPERIMENTAL SETUP

A. Our Multi-Mice PartsTrack Dataset

In this article, we introduce our new dataset for multi-mice part tracking in videos. The dataset was collected in collaboration with biologists of Queen’s University Belfast, U.K., for a study of neurophysiological mechanisms involved in Parkinson’s disease. In our dataset, two or three mice are interacting freely in a 50*110*30-cm home cage and are recorded from the top view using a Sony Action camera (HDR-AS15) with a frame rate of 30 fps and 640 by 480 pixels VGA video resolution. All the experiments are conducted in an environment-controlled room with constant temperature (27 °C) and light condition (long fluorescent lamp 40 W).

The dataset provides the detailed annotations for multiple mice in each video, as shown in Supplementary G. The mice used throughout this study were housed under constant climatic conditions with free access to food and water. All the experimental procedures were performed in accordance with the Guidance on the Operation of the Animals (Scientific Procedures) Act, 1986 (U.K.) and approved by the Queen’s University Belfast Animal Welfare and Ethical Review Body. Our database covers a wide range of activities such as contacting, following, and crossing. Moreover, our database contains a large amount of mouse pose and mouse part occlusion. After proper training, six professionals were invited to annotate mouse heads and tail bases and localize each mouse body in the videos. We assign a unique identity to every mouse part appearing in the images. If a mouse part was in the field-of-view but became invisible due to occlusion, it is marked “occluded.” Those mouse parts outside the image border limits are not annotated. In total, our dataset yields five videos of two mice and five videos of three mice, each video lasts 3 min, and 400 frames randomly sampled from each video are annotated (4000 frames are annotated in total). Since the problem of multi-mice parts has not been quantitatively evaluated in the literature, we follow the evaluation metrics defined in multitarget tracking [72], [73] for this problem and compare the results with several baseline methods. We split the dataset into training and testing sets with an equal duration of time and train our network based on transfer learning with pretrained models. The code and the dataset are published on github: https://github.com/ZhehengJiang/BILP.

B. Evaluation Metrics

To evaluate the proposed mouse part detector, we use the widely adopted precision, recall, average precision (AP), and mean average precision (mAP) [72]. For a specific class, the precision value corresponds to the ratio of the positive object detections against the total number of objects that the classifier predicts, while the recall value is defined as the percentage of the positive object detections against the total number of objects labeled as ground truth. The precision–recall curves are obtained by varying the model score threshold in the range of 0 and 1, which determines what is counted as a positive detection of the class. Note that with our metrics, only the object detections that have an IoU ratio over 0.5 with the ground truth are counted as positive detections while the rest are negatives. The AP score is defined as the average of precision at the set of 11 equally spaced recall values. mAP is just the average over all the classes. To provide a fair comparison for both types of occlusion handling, we consider an occluded mouse part correctly estimated either if (a) it is predicted at the correct location despite being occluded or (b) it is not predicted at all. Otherwise, the prediction is considered as a false positive (FP).
To evaluate the part tracking performance, we consider each mouse part trajectory as one individual target and compute multiple object tracking precision (MOTP) and multiple object tracking accuracy (MOTA) [74]. The former is derived from three types of error ratios: FP, missed targets (MT), and IDs. These errors are normalized by the number of objects appearing in the image frames and can be summed up to produce the resulting tracking accuracy, where 100% corresponds to zero errors. MOTP measures how precise each mouse part has been localized. We also report the trajectory-based measures of the number of mostly tracked (MT) and mostly lost (ML) targets. If a track hypothesis has covered at least 80% of its life span based on the ground-truth trajectories, it is considered as MT. If less than 20% are not tracked, the track hypothesis is considered as ML. IDF1 [73] measures the ratio of the correctly identified detections. OSPA computes the optimal subpattern assignment metric between a set of tracks and the ground truth.

VI. EXPERIMENTAL RESULTS

In this section, we evaluate the proposed method for part tracking on the newly introduced Multi-Mice PartsTrack dataset.

A. Evaluation of Part and Body Detector

1) Implementation: The multiple scales and aspect ratios of the root boxes in our method are different for mouse parts and body. Root boxes of inappropriate scales and aspect ratios are ineffective for mouse detection. For mouse “head” and “tail base,” we choose a single aspect ratio of 1:13 (width to height) and four scales with root box widths of 24, 29, 35, and 42 pixels based on the statistics of the object shape in the training set. For mouse “body” cases, we use multiple aspect ratios of 0.5 (landscape), 1 (square), and 2 (portrait) and three scales with root box widths of 50, 80, and 128 pixels. We label a root box as a positive example if its IoU ratio is greater than 0.7 with one ground-truthed box, and a negative example if its IoU ratio is lower than 0.3 with all the ground-truthed boxes. Root boxes that are neither positive nor negative do not contribute to the training RPNs. This experiment is conducted on the two-mice dataset.

2) Results: We first investigate the performance of the proposed part and body detector using different pretrained models. All the experiments adopt a three-stage PPN based on the trade-off between speeds and system performance, as shown in Supplementary H Fig. S3. From Table I, we observe that deeper network (VGG19) has achieved superior performance over other shallow networks (i.e., AlexNet and VGG16). It is because the deeper network can learn richer image representations. But surprisingly, the accuracy of the mouse parts is degraded after using more than nine layers of AlexNet, 11 layers of VGGNet16, or 15 layers of VGGNet19. This limitation is partially because of the low-resolution features in the feature map of higher layer. These features are not discriminative on small regions, and thus degrade the performance of the downstream classifier. In comparison, these features are cooperative enough to distinguish the mouse body from the background as the region area of the mouse body is three times larger than that of the mouse head and the tail base. This result also suggests that if reliable features can be extracted, the downstream classifier is able to improve the detection accuracy. In addition, using the network with layers less than six layers of VGGNet 16 or six layers of VGGNet 19 starts to demonstrate accuracy degradation, which can be explained by the weaker representation ability of the shallower layers. In Table I, we have also compared DNN-based detectors with two conventional methods [14], [16], which only provide methods to detect mouse body. The DNN-based detectors show better performance than the conventional method in body detection.

In Table II, we report the capability of different features to distinguish mouse parts and the background. Since CNN features perform better than the other hand-crafted features for body detection, we do not exploit any hand-crafted feature to maintain the system accuracy. For fair comparisons, all the proposals are generated from the same PPN which is initialized by the first 15 layers of VGG-19. We produce part
TABLE III
QUANTITATIVE EVALUATION OF MULTI-MICE PARTS TRACKING ON THE THREE-MICE DATASET

| Method (speed) | MOTA | MOTP | MT | ML | FP | FN | ID$ | IDF1 | OSPA |
|----------------|------|------|----|----|----|----|----|------|------|
| All (30.8 fps) | 81.1 | 65.5 | 25 | 0  | 566| 546| 12 | 91.2 | 6.55 |
| All(b,c,d)     | 80.2 | 65.6 | 24 | 0  | 581| 571| 26 | 89.7 |      |
| All(a,c,d)     | 79.5 | 65.6 | 25 | 0  | 615| 595| 12 | 90.75|      |
| All(a,b,d)     | 75.4 | 66.0 | 22 | 0  | 738| 718| 12 | 90.2 |      |
| All(a,b,c)     | 74.8 | 65.8 | 24 | 0  | 764| 735| 12 | 89.2 |      |

Impact of the geometric model
All without geometric model (34.7 fps) | 70.3 | 66.6 | 21 | 2  | 892| 872| 12 | 88.8 |      |

Impact of the motion model
All without motion model (31.6 fps) | 61.2 | 65.5 | 22 | 2  | 1101|1061|152|76.14|      |

Impact of the parts association model
All without parts association model (31.2 fps) | 76.3 | 65.4 | 24 | 1  | 688| 693|30 | 88.5 |      |

Comparison with the state-of-the-art
MOTDT [78] (20.6 fps) | 20.5 | 31.4 | 21 | 0  | 1592|1152|3103|20.2 |26.47 |
MDP [39] (2.5 fps) | 64.6 | 70.2 | 21 | 0  | 1114|994 | 18 | /    | 21.26 |
SORT [40] (260 fps) | 59.9 | 70.5 | 19 | 1  | 971 |1331| 106|56.4 |15.42 |
JPDAm [45] (32.6 fps) | 67.0 | 65.1 | 26 | 0  | 1264|548 | 155|44.6 |20.14 |
RNN-LSTM [46] (50 fps) | 50.6 | 67.3 | 20 | 0  | 1688|1176|102|54.6 |22.16 |
CEM [41] (7.7 fps) | 34.1 | 69.5 | 21 | 1  | 2637|1095|224|/    |24.75 |
DeepSORT [79] (20 fps) | 42.7 | 32.4 | 19 | 0  | 1796|1419|212|44.4 |18.63 |

(proposals from images and extract low-level features such as LBP, HOG, and SIFT. Interestingly, training our fc layers with LBP features on the same set of the PPN proposals actually has led to better average accuracy 95.1% (versus Conv’s 93.9% and HoG’s 88.3%), as shown in Table II. Combining LBP and Conv features, we can achieve the best result 97.37%. Similar results are also obtained using a linear SVM classifier. But the fc layers are more flexible and can be trained with other layers in an end-to-end style. Since too few interest points are detected from the mouse head and the tail base, SIFT features encoded by Fisher vector [80] present a much worse result on this task. Fig. 4 clearly shows that our combined LBP and conv features lead to the best performance for classifying mouse “head” and “tail base” proposals.

We also compare our detection network against Faster RCNN [55], SDD [75] and YOLO [76] on our dataset. These methods are also trained based on transfer learning with pretrained models. We train Faster RCNN based on the two pretrained models of Residual Network 50 and 101 [81]. SSD [75] has two input sizes (300 × 300 versus 512 × 512). The experiments show that the model with a larger input size leads to better results. YOLO uses its own pretrained model named Darknet. As shown in Fig. 5, most methods have high APs on the detection of head and body. However, the performance of Faster RCNN, SSD, and YOLO falls considerably when they are applied to detecting the tail base. Our detection method has a better performance to detect the tail base and its AP degrades less than the other methods. We also observe that if the threshold of detection scores was previously set high, lowering the threshold in Faster RCNN, SSD, and YOLO will introduce more false positives than our methods, resulting in a sharp decrease in precision as shown in Fig. 5.

B. Evaluation of Multi-Mice Parts Tracking

1) Implementation: In our tracking algorithm, each part target’s state contains position, velocity, and shape of bounding box $x_t = (x_p, y_p, w_p, h_p)$, $y_t = (x_v, y_v)$, and $(w_p, h_p)$, respectively, denote the position, the velocity, and the shape of each part target. For the shape of each target, we use the width and height of the detected bounding box. We model the motion of each part target based on the linear dynamical system discretized in the time domain and predict the state of each part target in the next image frame as $x_t = A x_{t-1} + \mu$, where $A = \text{diag}(A_x, A_y)$ is a constant transition model, and $A_x = A_y = [1, \tau; 0, 1]$. $\mu$ is a system noise and subject to a Gaussian distribution with covariance $\Omega = \text{diag}(\Omega_x, \Omega_y)$. $\Omega_x = \Omega_y = \Omega_\mu = \mu_\mu [\tau^3/3, \tau^2/2; \tau^2/2, \tau]$. Here, $\tau = 1$ and $\mu_\mu = 0.5$ refer to the sampling period and the process noise parameter, respectively. The uniform clutter density $\beta$ is estimated as $\beta = f_{\text{false}} / (w_{im} * h_{im})$, $w_{im} = 480$ and $h_{im} = 640$ are the width and height of the image, respectively, and $f_{\text{false}} = 0.1$ is the average number of the false detections per image frame. This experiment is conducted on both our two-mice and three-mice datasets. We compare the proposed approach against six baselines following their default settings.

2) Results: The results of multi-mice part tracking are reported in Table III and Supplementary Table S2. We quantitatively evaluate the proposed system using the

Fig. 4. ROC curve shows true positive rate (TPR) against false positive rate (FPR) for different mouse parts. (a) Head. (b) Tail.
the system performance, as they ensure that every solution of all the types of constraints makes evident contributions to the system performance, as they ensure that every solution is physically feasible. We also examine the impact of the geometric, motion, and part association models. Removing one of the three components significantly decreases the performance. In particular, the motion model plays the most crucial role, which is obvious on the three-mice dataset. The part association and geometric models can effectively reduce ID switch and FP respectively. This is expected because these two models help obtain the best target assignment and part association. We observe the change of our approach’s speeds by removing different components and the geometric model has the biggest impact on speeds. We also implement several standard multitarget tracking methods on our Multi-Mice PartsTrack dataset. For fair comparison, we use the proposed PPN detector to generate part bounding boxes and perform part tracking using several state-of-the-art part trackers [39]–[41], [45], [46], [78]. As we can see, at the bottom of Table III and Supplementary I Table S2, our tracker achieves the highest MOTA scores. Our method results in the lowest number of ID switches and highest IDF1 scores. This is primarily due to our powerful geometric and pair-based part association models, which can handle part identities more robustly. We conduct ablation studies on the multistage training design shown in Fig. S3, showing the effectiveness of our training strategy. The established SORT [40] has achieved the best efficiency due to the use of a simple target assignment model, where the cost function is fully dependent on IOU distance between each detection and the predicted bounding box. Evidence shows that this method cannot maintain consistent tracking performance (see Table III and S3). Note that the speed of all the approaches reported in Table III and S3 excludes the time of the part candidate detection. Fig. S5 and Table S2 show the tracking results of the conventional methods (Motr and MiceProfiler), which track mice by fitting ellipses and hard-coded geometric models, respectively. From Fig. S5, we observe that if the mice are very close, the models used in these methods cannot be properly fit, leading to tracking targets’ swaps or incorrect part localization. Moreover, these models are hard-coded in the system, and thus limit the flexibility of their methods. Figs. S6, S8, and S9 in Supplementary F provide the qualitative comparison of the proposed tracking method against MDP [39] and JPDAm [45] using the same detection results. As shown in Fig. S6, MDP swap identities between targets 2 and 5 after occlusion is caused by “pinning” in the image of the middle column, while JPDAm assigns a new identity to target 6 after it is occluded by target 3. Similar problems can also be witnessed in Figs. S8 and S9. Compared with these two tracking methods, the proposed approach correctly integrates the detection results with the tracking practice and predicts the occlusion.

Overall, our approach is not the fastest in multi-mice tracking, but sustains the best accuracy. Fig. S4 shows exemplar tracking results of the test sequences in the proposed Multi-Mice PartsTrack dataset.

C. Generalization to Other Datasets

To demonstrate the generalization of our proposed approach, we also implement the approach on two public datasets related to other insects or animals: locusts were recorded in a laboratory setting and zebras recorded in the wild [82]. The publicly available datasets of multiple animals recorded from the top view are scarce and the two datasets appearing here are examples. They contain 700 and 900 annotated frames for training and validation, and 1300 and 1600 unlabeled frames for testing, respectively. Following [82], we split the annotated datasets into 90% training examples and 10% validation examples. Our detection approach achieves 100% accuracy of detecting animal’s heads, tails, and bodies on both the datasets. Since their testing datasets are unlabeled, we illustrate the performance of the proposed approach against these datasets in the supplementary videos. In comparison, our dataset seems more challenging, where the mouse parts are deformable with more occlusions during moving. Moreover, our annotated dataset is larger with a quantitative test bed for part detection and tracking of multiple mice.

VII. Conclusion

In this article, we have presented a novel method for markerless multi-mice part detection and tracking. We have demonstrated that the proposed multistage part and body
detector showed effective hard negative mining and achieved satisfactory detection results. We also proposed a new formulation based on target assignment with the learned geometric constraints and a pair-wise association scheme with motion consistency and restriction. Moreover, we presented a challenging annotated dataset to evaluate the proposed algorithms for multi-mice part tracking. The experimental results on the proposed datasets demonstrate that the proposed algorithm outperformed other state-of-the-art and conventional methods. We also demonstrate the generalization ability of our approach on other species. Our future work will address social behavior analysis using the proposed tracking method.

REFERENCES

[1] C. Liu, J. Wang, H. Li, C. Fietkiewicz, and K. A. Loparo, “Modeling and analysis of beta oscillations in the basal ganglia,” IEEE Trans. Neural Netw. Learn. Syst., vol. 29, no. 5, pp. 1864–1875, May 2018.

[2] A. Vogel-Ciernia et al., “The neuron-specific chromatin regulatory subunit BAF53b is necessary for synaptic plasticity and memory,” Nature Neurosci., vol. 16, no. 5, pp. 552–561, May 2013.

[3] A. V. Kalnai, A. M. Stewart, C. Song, K. C. Berriedge, A. M. Graybiel, and J. C. Fentress, “Neurobiology of rodent self-grooming and its value for translational neuroscience,” Nature Rev. Neurosci., vol. 17, no. 1, pp. 45–59, Jan. 2016.

[4] H. Wang, J. Peng, X. Zheng, and S. Yue, “A robust visual system for small target motion detection against cluttered moving backgrounds,” IEEE Trans. Neural Netw. Learn. Syst., vol. 31, no. 3, pp. 839–853, Mar. 2020.

[5] J. A. Villacorta-Atienza and V. A. Makarov, “Social network architecture for cognitive navigation in dynamic environments,” IEEE Trans. Neural Netw. Learn. Syst., vol. 24, no. 12, pp. 2075–2087, Dec. 2013.

[6] A. F. R. Araujo and O. V. Santana, “Self-organizing map with time-varying structure to plan and control artificial locomotion,” IEEE Trans. Neural Netw. Learn. Syst., vol. 26, no. 8, pp. 1594–1607, Aug. 2015.

[7] F. De Chaumont et al., “Computerized video analysis of social interactions in mice,” Nature Methods, vol. 9, no. 4, p. 410, 2012.

[8] H. Jiang et al., “Automated home-cage behavioural phenotyping of mice,” Nature Commun., vol. 1, no. 1, pp. 1–9, Dec. 2010.

[9] Z. Jiang et al., “Context-aware mouse behavior recognition using hidden Markov models,” IEEE Trans. Image Process., vol. 28, no. 3, pp. 1133–1148, Mar. 2019.

[10] X. P. Burgos-Artizzu, P. Döllar, D. Lin, D. J. Anderson, and P. Perona, “Social behaviour recognition in continuous video,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2012, pp. 1322–1329.

[11] J. Altman, “Observational study of behavior: Sampling methods,” Behaviour, vol. 49, nos. 3–4, pp. 227–266, 1974.

[12] M. B. Sokolowski, “Social interactions in ‘simple’ model systems,” Neuron, vol. 65, no. 6, pp. 780–794, Mar. 2010.

[13] A. J. Spink, R. A. J. Tegelenbosch, M. O. S. Buma, and L. P. J. J. Noldus, “The EthoVision video tracking system—A tool for studying ethology,” Neuron, vol. 49, nos. 3–4, pp. 227–266, 1974.

[14] B. R. Dennis, H. W. Cheng, and I. Estevez, “Appearance matters: Artifactual marking alters aggression and stress,” Poultry Sci., vol. 87, no. 10, pp. 1939–1946, Oct. 2008.

[15] K. Galwashworthy et al., “A comparison of wild-caught wood mice and bank voles in the intellecivate: Assessing exploration, daily activity patterns and place learning paradigms,” Behav. Brain Res., vol. 157, no. 2, pp. 211–217, Feb. 2005.

[16] S. Ohayon, O. Avni, A. L. Taylor, P. Perona, and S. E. R. Egnor, “Automated multi-day tracking of marked mice for the analysis of social behaviour,” J. Neurosci. Methods, vol. 219, no. 1, pp. 10–19, 2013.

[17] S. Ballesta, G. Reymond, M. Pozzobon, and J.-R. Duhame, “A real-time 3D video tracking system for monitoring primate groups,” J. Neurosci. Methods, vol. 234, pp. 147–152, Aug. 2014.

[18] R. L. Dennis, R. C. Newberry, H.-W. Cheng, and I. Estevez, “Appearance matters: Artificial marking alters aggression and stress,” Poultry Sci., vol. 87, no. 10, pp. 1939–1946, Oct. 2008.

[19] M. Galsworthy et al., “DeepCut: Joint subset partition and labeling for multi person pose estimation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 4929–4937.

[20] L. Pishchulina et al., “DeepCut: Joint subset partition and labeling for multi person pose estimation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 4929–4937.

[21] A. Perez-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,” Nature Methods, vol. 11, no. 7, pp. 743, 2014.

[22] L. Pishchulina et al., “DeepCut: Joint subset partition and labeling for multi person pose estimation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 4929–4937.

[23] G. Shu, A. Dehghan, O. Oreifej, E. Hand, and M. Shah, “Part-based object tracking by decision making,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2015, pp. 5033–5041.

[24] A. Weissbrod et al., “Automated long-term tracking and social behaviour analysis with multiple mice,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2015, pp. 5033–5041.

[25] S. Tang, B. Andres, M. Andriluka, and B. Schiele, “Subgraph decomposition for multi-target tracking,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2015, pp. 5033–5041.
H. Piriavash, D. Ramanan, and C. C. Fowlkes, “Globally-optimal greedy algorithms for tracking a variable number of objects,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2011, pp. 1201–1208.

H. B. Shitrit, J. Berczal, F. Fleuret, and P. Fua, “Multi-commodity network flow for tracking multiple people,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 36, no. 8, pp. 1614–1627, Aug. 2014.

S. H. Rezatofighi, A. Milan, Z. Zhang, Q. Shi, A. Dick, and I. Reid, “Joint probabilistic data association revisited,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Dec. 2015, pp. 3047–3055.

A. Milan, S. H. Rezatofighi, A. Dick, I. Reid, and K. Schindler, “Online multi-target tracking using recurrent neural networks,” in Proc. 31st AAAI Conf. Artif. Intell., 2017, pp. 4225–4232.

I. Pschucher, A. Jain, M. Andriluka, T. Sturm, and B. Schiele, “Articulated people detection and pose estimation: Reshaping the future,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2012, pp. 3178–3185.

G. Gkioxari, B. Hariharan, R. Girshick, and J. Malik, “Using k-poses for detecting people and localizing their keypoints,” in Proc. Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2014, pp. 3582–3589.

K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask R-CNN,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 2980–2988.

Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, “Realtime multi-person 2D pose estimation using part affinity fields,” 2016, arXiv:1611.08050.

S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, “Convolutional pose machines,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 4724–4732.

T. F. Coates, C. J. Taylor, D. H. Cooper, and J. Graham, “Active shape models-their training and application,” Comput. Vis. Image Understand., vol. 61, no. 1, pp. 38–59, 1995.

A. Mathis et al., “DeepLabCut: Markerless pose estimation of user-defined body parts with deep learning,” Nature Neurosci., vol. 21, pp. 1281–1289, 2018.

T. D. Pereira et al., “Fast animal pose estimation using deep neural networks,” Nature Methods, vol. 16, no. 1, p. 117, 2019.

S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” in Proc. Adv. Neural Inf. Process. Syst., 2015, pp. 91–99.

J. Uijlings, K. van de Sande, and T. Gevers, “Selective search for object recognition,” Int. J. Comput. Vis., vol. 104, no. 2, pp. 154–171, Oct. 2013.

R. Girshick, “Fast R-CNN,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Dec. 2015, pp. 1440–1448.

K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2014, arXiv:1409.1556.

O. Russakovsky et al., “ImageNet large scale visual recognition challenge,” Int. J. Comput. Vis., vol. 115, no. 3, pp. 211–252, Dec. 2015.

D. Fischler and R. Elschlager, “Matching patterns with large transformations,” in Proc. IEEE Conf. Comput. Vis. Workshops (ICCV Workshops), Nov. 2011, pp. 971–981.

Z. Khan, T. Balch, and F. Dellaert, “An MCMC-based particle filter for tracking multiple interacting targets,” in Proc. Eur. Conf. Comput. Vis. Berlin, Germany: Springer, 2004, pp. 279–290.

M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool, “Robust tracking-by-detection using a detector confidence pattern filter,” in Proc. IEEE 12th Int. Conf. Comput. Vis., Sep. 2009, pp. 1515–1522.

N. Wojke and D. Paulus, “Global data association for the probability hypothesis density filter using network flows,” in Proc. Proc. IEEE Int. Conf. Robot. Automat. (ICRA), May 2016, pp. 567–572.

N. Chenouard, I. Bloch, and J. Olive-Marin, “Multiple hypothesis tracking for cluttered biological image sequences,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, no. 11, pp. 2736–3750, Nov. 2013.

A. G. A. Perera, C. Srinivas, A. Hoogs, G. Broksby, and W. Hu, “Multi-object tracking through simultaneous long occlusions and split-merge conditions,” in Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR), vol. 1, Jun. 2016, pp. 656–673.

J. Jiebel, D. M. Gavrila, and C. Schnörr, “A Bayesian framework for multi-cue 3D object tracking,” in Proc. Eur. Conf. Comput. Vis. Berlin, Germany: Springer, 2004, pp. 241–252.

Z. Qin and C. R. Shelton, “Improving multi-target tracking via social grouping,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2012, pp. 1972–1978.

K. Yamaguchi, A. C. Berg, L. E. Ortiz, and T. L. Berg, “Who are you with and where are you going?” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2011, pp. 1345–1352.

S. Pellegrini, A. Ess, K. Schindler, and L. van Gool, “You’ll never walk alone: Modeling social behavior for multi-target tracking,” in Proc. IEEE 12th Int. Conf. Comput. Vis., Sep. 2009, pp. 261–268.

E. L. Lawler and D. E. Wood, “Branch-and-bound methods: A survey,” Oper. Res., vol. 14, no. 4, pp. 699–719, Jul/Aug. 1966.

A. Nemirovski, “Interior point polynomial time methods in convex programming,” School Ind. Syst. Eng., Georgia Inst. Technol., Atlanta, GA, USA, Lect. Notes ISYE 8813, 2004.

M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The PASCAL visual object classes (VOC) challenge,” Int. J. Comput. Vis., vol. 88, no. 2, pp. 303–338, Jun. 2010.

E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi, “Performance measures and a data set for multi-target, multi-camera tracking,” in Proc. Eur. Conf. Comput. Vis. Cham, Switzerland: Springer, 2016, pp. 17–35.

K. Bernardin and R. Stiefelhagen, “Evaluating multiple object tracking performance: The CLEAR MOT metrics,” EURASIP J. Image Video Process., vol. 2008, pp. 1–10, Dec. 2008.

W. Liu et al., “SSD: Single shot MultiBox detector,” in Proc. Eur. Conf. Comput. Vis. Cham, Switzerland: Springer, 2016, pp. 21–37.

J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 779–788.

A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in Proc. Adv. Neural Inf. Process. Syst., 2012, pp. 1097–1105.

L. Chen, H. Ai, Z. Zhuang, and C. Shang, “Real-time multiple people tracking with deeply learned candidate selection and person re-identification,” 2018, arXiv:1809.04427.

N. Wojke, A. Bewley, and D. Paulus, “Simple online and realtime tracking with a deep association metric,” in Proc. IEEE Int. Conf. Image Process. (ICIP), Sep. 2017, pp. 3645–3649.

J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek, “Image classification with the Fisher vector: Theory and practice,” Int. J. Comput. Vis., vol. 105, no. 3, pp. 222–245, 2013.

K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 770–778.

A. M. Graving et al., “DeepPoseKit, a software toolkit for fast and robust animal pose estimation using deep learning,” eLife, vol. 8, p. e47994, Oct. 2019.
**Long Chen** is currently pursuing the Ph.D. degree with the School of Informatics, University of Leicester, Leicester, U.K. His research interests are in the areas of computer vision and machine learning.

**Lei Tong** is currently pursuing the Ph.D. degree with the School of Informatics, University of Leicester, Leicester, U.K. His research interests include computer vision, social network analysis, and data mining.

**Xiangrong Zhang** received the B.S. and M.S. degrees from the School of Computer Science, Xidian University, Xi’an, China, in 1999 and 2003, respectively, and the Ph.D. degree from the School of Electronic Engineering, Xidian University, in 2006. She was a Visiting Scientist with the Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, from January 2015 to March 2016. She is currently a Professor with the Key Laboratory of Intelligent Perception and Image Understanding of the Ministry of Education, Xidian University. Her research interests include pattern recognition, machine learning, and remote sensing image analysis and understanding.

**Xiangyuan Lan** received the B.Eng. degree in computer science and technology from the South China University of Technology, Guangzhou, China, in 2012, and the Ph.D. degree from the Department of Computer Science, Hong Kong Baptist University, Hong Kong, in 2016. He was a Visiting Scholar with the Center for Automation Research, University of Maryland Institute for Advanced Computer Studies (UMIACS), University of Maryland, College Park, MD, USA, from January 2015 to July 2015. He is currently with Shenzhen NS-Tech Company Ltd., Shenzhen, China. His current research interests include intelligent video surveillance, biometric security, and health informatics.

Dr. Lan is currently an Associate Editor of Signal, Image and Video Processing (Springer).

**Danny Crookes** (Senior Member, IEEE) received the B.Sc. and Ph.D. degrees from Queen’s University Belfast, Belfast, U.K., in 1997 and 1980, respectively. He is currently involved in the projects of medical imaging for cancer diagnosis, speech separation and enhancement, and capital markets software using graphics processing units (GPUs). In 1993, he was an appointed Professor of computer engineering with Queen’s University Belfast, and he was the Head of Computer Science from 1993 to 2002. He has published over 230 scientific papers in journals and international conferences. His research interests include the use of acceleration technologies (including FPGAs and GPUs) for high-performance image, video, and speech processing.

**Ming-Hsuan Yang** (Fellow, IEEE) is currently a Professor of electrical engineering and computer science with the University of California at Merced, Merced, CA, USA, and also an Adjunct Professor with Yonsei University, Seoul, South Korea. Prof. Yang received the NSF CAREER Award in 2012, the Senate Award for Distinguished Early Career Research at UC Merced in 2011, and the Google Faculty Award in 2009. He served as an Associate Editor for the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE from 2007 to 2011, and is an Associate Editor of the International Journal of Computer Vision, Image and Vision Computing, and Journal of Artificial Intelligence Research.

**Huiyu Zhou** received the B.Eng. degree in radio technology from the Huazhong University of Science and Technology, Wuhan, China, in 1990, the M.Sc. degree in biomedical engineering from University of Dundee, Dundee, U.K., in 2002, and the Ph.D. degree in computer vision from Heriot-Watt University, Edinburgh, U.K., in 2006. He is currently a Full Professor with the School of Computing and Mathematical Sciences, University of Leicester, Leicester, U.K. He has published over 400 peer-reviewed articles in the field. His research work has been or is being supported by the U.K., EPSRC, ESRC, AHRC, MRC, EU, Royal Society, Leverhulme Trust, Puffin Trust, Invest NI, and industry.