2020 IEEE International Conference on Robotics and Automation (ICRA)
31 May - 31 August, 2020. Paris, France

Multi-person Pose Tracking using Sequential Monte Carlo with Probabilistic Neural Pose Predictor

Masashi Okada[†,⋆], Shinji Takenaka[‡] and Tadahiro Taniguchi[†,*]

***Abstract*— It is an effective strategy for the multi-person pose tracking task in videos to employ prediction and pose matching in a frame-by-frame manner. For this type of approach, uncertainty-aware modeling is essential because precise prediction is impossible. However, previous studies have relied on only a single prediction without incorporating uncertainty, which can cause critical tracking errors if the prediction is unreliable. This paper proposes an extension to this approach with Sequential Monte Carlo (SMC). This naturally reformulates the tracking scheme to handle multiple predictions (or hypotheses) of poses, thereby mitigating the negative effect of prediction errors. An important component of SMC, i.e., a proposal distribution, is designed as a *probabilistic neural pose predictor*, which can propose diverse and plausible hypotheses by incorporating epistemic uncertainty and heteroscedastic aleatoric uncertainty. In addition, a recurrent architecture is introduced to our neural modeling to utilize time-sequence information of poses to manage difficult situations, such as the frequent disappearance and reappearances of poses. Compared to existing baselines, the proposed method achieves a state-of-the-art MOTA score on the PoseTrack2018 validation dataset by reducing approximately 50% of tracking errors from a state-of-the art baseline method.**

I. Introduction

Object detection and tracking are important tasks for various robotics applications, such as autonomous vehicle control [1]–[4], visual SLAM [5], [6], and robotics manipulator control [7]. Multi-person pose estimation and tracking is a critical component in various applications, such as video surveillance and sports video analytics. In the past few years, pose estimation has progressed significantly [8] due to deep convolutional learning assisted by large-scale image corpora, such as COCO [9] and MPPI [10]. The PoseTrack dataset [11] is a video corpus for pose estimation and tracking that is annotated with multiple people in scenes and this dataset has encouraged the community to develop a diverse range of pose estimation and tracking models [12]–[17].

Most of these pose tracking models employ a two-stage scheme, i.e., *1)* poses are estimated using a deep convolutional neural network (CNN), and then *2)* poses are tracked by employing greedy bipartite matching in a frame-by-frame manner. For example, *Simple-Baseline* [15] (the Pose Track Challenge ECCV'18 Winner) introduces matching utilizing *flow-based pose similarity*, which is defined as the *object keypoint similarity* (OKS) [9] between a pose estimated from a current frame and a pose predicted from previous results using optical-flows. This optical-flow-based prediction can compensate the pose differences of the same person across multiple frames, which making the matching scores robust against fast movements of people and cameras. We believe that this type of prediction-based matching is general and strong approach, and can be applied to not only for pose tracking but also various multi-object tracking tasks.

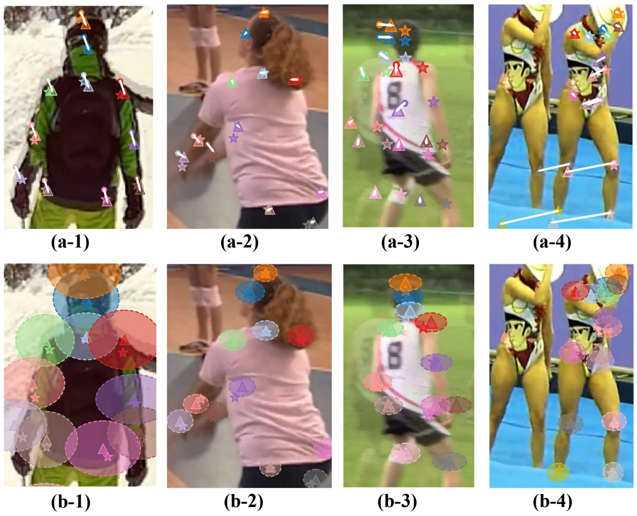

(a-1) (a-2) (a-3) (a-4)
(b-1) (b-2) (b-3) (b-4)

Fig. 1. Human poses predicted by (a) optical-flows and (b) the proposed probabilistic neural predictor. Two consecutive frames are rendered with α-blending with ground-truth current keypoints shown as '$\star$'s. In (a), '$\bullet$'s and '$\triangle$'s indicate previous and predicted keypoints, respectively. The solid lines between '$\bullet$'s and '$\triangle$'s illustrate optical-flows obtained by a state-of-the-art optical-flow estimator PWC-Net [18]. In (b), '$\triangle$'s and ovals with dashed-lines respectively indicate the average and deviation (2σ) of 100 different particles predicted by inputting the same 100 inputs to the predictor.

However, the matching process strongly relies on a single hypothesis, which could be critically vulnerable to prediction errors. The errors cause underestimation of the matching score between correct pairs, thereby resulting in mismatching. An optical-flow based prediction can give reasonable predictions, as demonstrated in Fig. 1 (a-1); however, this can also generate unreliable predictions especially when poses change quickly or there is insufficient texture information available around a given keypoint (see Figs. 1 (a-2)–(a-4) for examples). A possible solution to this is pursuing prediction accuracy; however, precise prediction appears impossible due to uncertainties [19] in human gait, imaging, pose estimation by deep CNN models, etc. Another concern is that optical-flows can be sensitive to the disappearance and reappearance

[†] Masashi Okada and Tadahiro Taniguchi are with AI Solutions Center, Business Innovation Division, Panasonic Corporation, Japan.

[‡] Shinji Takenaka is with Technology Center, Panasonic System Networks R&D Lab. Co., Ltd., Japan.

[*] Tadahiro Taniguchi is also with Ritsumeikan University, College of Information Science and Engineering, Japan.

[⋆] Corresponding author: okada.masashi001@jp.panasonic.com

978-1-7281-7395-5/20/$31.00 ©2020 IEEE

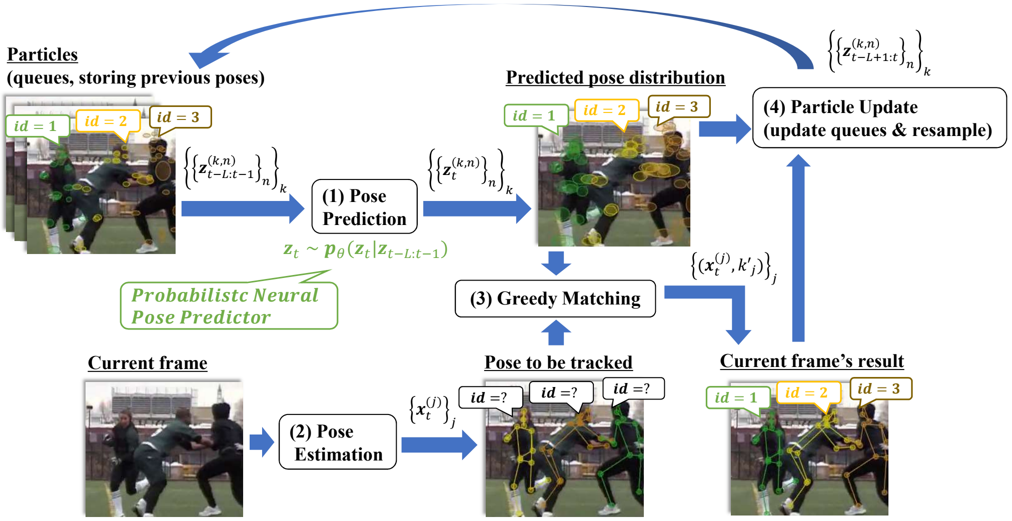

Particles
(queues, storing previous poses)
id = 1
id = 2
id = 3
$\left\{\left\{z^{(k,n)}_{t-L:t-1}\right\}_n\right\}_k$
(1) Pose Prediction
$z_t \sim p_\theta(z_t|z_{t-L:t-1})$
Probabilistc Neural Pose Predictor
$\left\{\left\{z^{(k,n)}_{t}\right\}_n\right\}_k$
Predicted pose distribution
id = 1
id = 2
id = 3
(4) Particle Update (update queues & resample)
$\left\{\left\{z^{(k,n)}_{t-L+1:t}\right\}_n\right\}_k$
(3) Greedy Matching
$\left\{(x^{(j)}_t, k'_j)\right\}_j$
Current frame
(2) Pose Estimation
$\left\{x^{(j)}_t\right\}_j$
Pose to be tracked
id =?
id =?
id =?
Current frame's result
id = 1
id = 2
id = 3

Fig. 2. Proposed SMC-based pose tracking method. The details of this method will be described in Sec. II–IV.

of poses caused by occlusions. The baseline introduces multiple optical-flows of previous multi-frames to mitigate this issue; however, estimating accurate multi-frame optical-flows requires complicated processes [20], [21].

Motivated by this, we have designed an extenstion to this tracking strategy. Note that this paper primarily focuses on the pose tracking methodology and does not discuss about CNN modeling for pose estimation. In the tracking system we implemented, the CNN pose estimator from the *Simple-Baseline* is adopted without modification. Our primary contributions are summarized as follows.

- We propose a multi-person tracking method that exploits the newly-devised *probabilistic neural pose predictor* and the well-known Sequential Monte Carlo (SMC; or *particle filter*), the diagram of which is illustrated in Fig. 2.
- The proposed tracking method achieves a state-of-the-art Multiple Object Tracking Accuracy (MOTA) [22] score of 66.2 on the PoseTrack2018 validation dataset. Our method outperforms both the ECCV'18 Winner *Simple-Baseline* [15] (MOTA: 65.4) and a more recent baseline method [16] (MOTA: 65.7).

Our probabilistic neural pose predictor, as a principal component of SMC, probabilistically predicts the next poses considering two kinds of epistemic uncertainty (model uncertainty due to limited data) and heteroscedastic aleatoric uncertainty (inherent system stochasticity). This stochasticity of the predictor allows us to prepare diverse predictions (or hypotheses). To consider long context information, a recurrent neural modeling with Long Short-Term Memory (LSTM) [23] is introduced to manage difficult situations, such as frequent occlusions. Figure 1 (b) illustrates how the predictor infers plausible pose distributions.

The remainder of this paper is organized as follows. In Sec. II, we briefly review SMC and formulate a single pose tracking problem with SMC. In Sec. III, we summarize the concept and architecture of the pose predictor. Sec. IV proposes a tracking method that exploits this predictor. In Sec. V, the effectiveness of the proposed method is demonstrated in an evaluation using the PoseTrack2018 dataset.

II. SMC for Single Object Tracking

This section briefly describes an SMC formulation for a single object tracking task. Fig. 3 shows the graphical model for this formulation, which takes a more general form subsuming common state-space models, such as Hidden Markov Models. This model comprises latent states $\boldsymbol{z}_{1:T}$ and observed states $\boldsymbol{x}_{1:T}$. Latent state $\boldsymbol{z}_t$ can be predicted by a transition model $\boldsymbol{z}_t \sim p(\boldsymbol{z}_t|\boldsymbol{z}_{1:t-1}, \theta)$ parameterized with θ, the posterior $p(\theta|\mathcal{D})$ of which is inferred from given training dataset $\mathcal{D} = \{(\boldsymbol{z}_{1:t-1}, \boldsymbol{z}_t)\}$. The joint distribution of this model takes the factorized form $p(\boldsymbol{z}_{1:T}, \boldsymbol{x}_{1:T}, \theta) = p(\boldsymbol{z}_1)p(\boldsymbol{x}_1|\boldsymbol{z}_1)\prod_{t=2}^{T} p_\theta(\boldsymbol{z}_t|\boldsymbol{z}_{1:t-1})p(\boldsymbol{x}_t|\boldsymbol{z}_{1:t}, \boldsymbol{x}_{1:t-1})$, where $p_\theta(\boldsymbol{z}_t|\boldsymbol{z}_{1:t-1}) := p(\boldsymbol{z}_t|\boldsymbol{z}_{1:t-1}, \theta)p(\theta|\mathcal{D})$. The objective of SMC is to approximately infer the posterior over the latent state sequence with a set of N *particles* as

$$p(\boldsymbol{z}_{1:T}|\boldsymbol{x}_{1:T}) \simeq \sum_{n=1}^{N} w(\boldsymbol{z}^{(n)}_{1:T})\delta(\boldsymbol{z}_{1:T} - \boldsymbol{z}^{(n)}_{1:T}), \tag{1}$$

where δ is the Dirac delta function. $w(\boldsymbol{z}^{(n)}_{1:T})$ is the weight of a particle n, which can be recursively defined as

$$w(\boldsymbol{z}^{(n)}_{1:T}) \propto w(\boldsymbol{z}^{(n)}_{1:T-1})\frac{p_\theta(\boldsymbol{z}^{(n)}_T|\boldsymbol{z}^{(n)}_{1:T-1})p(\boldsymbol{x}_T|\boldsymbol{z}^{(n)}_{1:T}, \boldsymbol{x}_{1:T-1})}{q(\boldsymbol{z}^{(n)}_T|\boldsymbol{z}^{(n)}_{1:T-1}, \boldsymbol{x}_{1:T})}, \tag{2}$$

where $q(\boldsymbol{z}^{(n)}_T|\boldsymbol{z}^{(n)}_{1:T-1}, \boldsymbol{x}_{1:T})$ is a proposal distribution for importance sampling. Note that this proposal distribution must be defined carefully to propose plausible particles. A similar derivation of (2) can be found in the literature [24].

In this formulation, particles are managed according to the following procedures. First, at time step T, new particles $\boldsymbol{z}^{(n)}_T$ are proposed from $q(\boldsymbol{z}^{(n)}_T|\boldsymbol{z}^{(n)}_{1:T-1}, \boldsymbol{x}_{1:T})$. Second, the particles are weighted by (2), and then particles $\boldsymbol{z}^{(n)}_{1:T}$ are

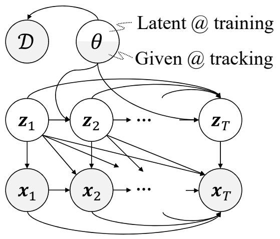


Fig. 3. Graphical model of pose track formulation.

resampled from a categorical distribution Cat as

$$\boldsymbol{z}_{1:T}^{(n)} \sim \mathrm{Cat}\left(N, \{w(\boldsymbol{z}_{1:T}^{(n)})\}_n\right). \tag{3}$$

After resampling, all weights $w(\boldsymbol{z}_{1:T}^{(n)})$ are made uniform.

In our case, $\boldsymbol{x}_t$ and $\boldsymbol{z}_t^{(n)}$ correspond to the estimated and predicted poses of a single person, respectively. Managing multiple particles allows us to have multiple pose hypotheses. The likelihood (or confidence) of each hypothesis is evaluated by the second term in the nominator of (2). This term is referred to as a *likelihood function*, and we define this function with OKS between $\boldsymbol{x}_T$ and $\boldsymbol{z}_T^{(n)}$ as

$$p(\boldsymbol{x}_T | \boldsymbol{z}_{1:T}^{(n)}, \boldsymbol{x}_{1:T-1}) \propto \mathrm{OKS}(\boldsymbol{x}_T, \boldsymbol{z}_T^{(n)}). \tag{4}$$

We assume the proposal distribution takes the form

$$\begin{aligned} &q(\boldsymbol{z}_T^{(n)} | \boldsymbol{z}_{1:T-1}^{(n)}, \boldsymbol{x}_{1:T}) \\ &:= p_\theta(\boldsymbol{z}_T^{(n)} | \boldsymbol{z}_{1:T-1}) q(\boldsymbol{z}_{1:T-1} | \boldsymbol{z}_{1:T-1}^{(n)}, \boldsymbol{x}_{1:T-1}). \end{aligned} \tag{5}$$

The second term on the right-hand side is defined as

$$\begin{aligned} &q(\boldsymbol{z}_{1:T-1} | \boldsymbol{z}_{1:T-1}^{(n)}, \boldsymbol{x}_{1:T-1}) = \\ &\alpha \delta(\boldsymbol{z}_{1:T-1} - \boldsymbol{z}_{1:T-1}^{(n)}) + (1-\alpha) \delta(\boldsymbol{z}_{1:T-1} - \boldsymbol{x}_{1:T-1}), \end{aligned} \tag{6}$$

which selects $\boldsymbol{z}_{1:T-1}^{(n)}$ or $\boldsymbol{x}_{1:T-1}$ probabilistically as the input to prediction model $p_\theta(\boldsymbol{z}_T^{(n)} | \boldsymbol{z}_{1:T-1})$. The parameter $\alpha \in [0, 1]$ controls how we weight the prediction and observation. We found that $\alpha = 0.45$ achieves the best performance on multi-person pose track formulation. In summary, (2) can be described as

$$w(\boldsymbol{z}_{1:T}^{(n)}) \propto w(\boldsymbol{z}_{1:T-1}^{(n)}) \frac{\mathrm{OKS}(\boldsymbol{x}_T, \boldsymbol{z}_T^{(n)})}{q(\boldsymbol{z}_{1:T-1} | \boldsymbol{z}_{1:T-1}^{(n)}, \boldsymbol{x}_{1:T-1})}. \tag{7}$$

In Sec. III, we discuss how we design predictor $p_\theta(\cdot)$ to propose plausible hypotheses, and, in Sec. IV, we extend this SMC scheme to multi-person pose tracking.

III. Probabilistic Neural Pose Predictor

We model predictor $p_\theta(\cdot)$ as a trainable neural network. Fig. 4 illustrates the architecture of this model, which we refer to as the *probabilistic neural pose predictor*. This predictor is designed to have probabilistic behaviors by incorporating epistemic uncertainty and heteroscedastic aleatoric uncertainty [19], which allow us to propose multiple hypotheses satisfying both diversity and plausibility. In addition, to utilize the time-sequence input $z_{1:T-1}$, we employ recurrent neural modeling by LSTM [23] with a stateless architecture. Here, the time length is constrained to L, thus $z_{T-L:T-1}$ is input to the model.

A. Epistemic Uncertainty

Epistemic uncertainty accounts for uncertainty in the model parameters θ due to a lack of sufficient data $\mathcal{D}$, i.e., $p(\theta|\mathcal{D})$, which is also referred to as model uncertainty. If nearly infinite data is available, this uncertainty should vanish. However, in a practical case where $\mathcal{D}$ is insufficient and/or a model has a deep (or overparameterized) architecture, this uncertainty remains and should be managed carefully. We exploit *dropout as inference* [25] to model this uncertainty, which approximates $p(\theta|\mathcal{D})$ as a Gaussian distribution $q(\theta)$. It has been proven that the variational inference problem, i.e., $\mathrm{argmin}_q \mathrm{KL}(q(\theta)||p(\theta|\mathcal{D}))$, is approximately equivalent to training networks with dropout, where $\mathrm{KL}(\cdot||\cdot)$ denotes Kullback-Leibler divergence. In our modeling, dropout is performed both in the training and test steps. Consequently, even if the same inputs are fed into the predictor during the test step, different models are sampled probabilistically from $q(\theta)$ and applied to each input, thereby proposing diverse hypotheses.

B. Heteroscedastic Aleatoric Uncertainty

Aleatoric uncertainty represents noise inherent in observations. In pose prediction, this uncertainty can originate from sudden changes of human gaits, fast camera panning and tilting, and pose estimation errors by deep CNN models. We model aleatoric uncertainty with heteroscedasticity. Specifically, our predictor is trained to estimate input-dependent Gaussian distributions and adaptively change the diversity (or variance) of particles according to different situations. For example, in cases in which people move very quickly, the predictor scatters particles in a wider area. In cases with slower movement, the predictor concentrates particles in a narrower area. This behavior helps us utilize the finite particles effectively.

In the training step, model parameter θ is optimized to minimize the log-likelihood loss defined as

$$\begin{aligned} &-\log p(\theta|\mathcal{D}) \propto -\log p(\mathcal{D}|\theta) p(\theta) \propto \\ &\sum_i \left\{ (\boldsymbol{\mu}_i - \boldsymbol{\mu}_i^*)^T \boldsymbol{\sigma}^{-1} (\boldsymbol{\mu}_i - \boldsymbol{\mu}_i^*) + \log |\boldsymbol{\sigma}_i| \right\} + \lambda ||\theta||^2, \end{aligned} \tag{8}$$

where i is the index of an element in dataset $\mathcal{D}$. $\boldsymbol{\mu}_i$, $\boldsymbol{\sigma}_i$ and $\boldsymbol{\mu}_i^*$ respectively denote estimated mean, deviation and ground-truth for input i. The last term in (8) is an L2 regularization term originating from the Gaussian posterior $p(\theta)$. In the test step, a single point is sampled from the estimated Gaussian distribution.

IV. Proposed Tracking Method

Algorithm 1 gives the pseudocode for the proposed tracking method. We implemented this method using TensorFlow [26], and all independent threads and for-loops except for the outermost loop ($\ell 2$) can be executed efficiently in parallel on GPUs. Our prototype implementation can simultaneously track 10 poses at about 30 fps on a single

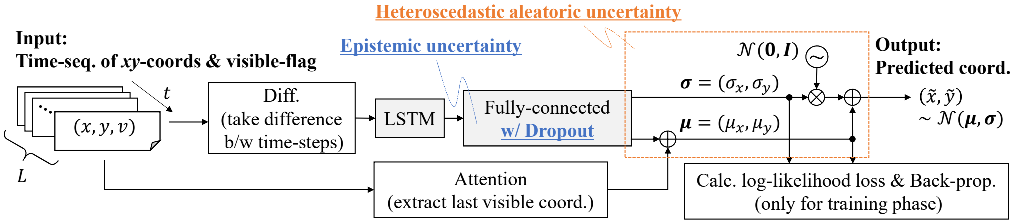


Fig. 4. Architecture of probabilistic neural pose predictor. Prediction is performed in a pointwise manner, i.e., each keypoint of a pose is processed independently. Shaded boxes indicate trainable modules parameterized with θ, which are trained to estimate the residual between the last visible coordinates and the current prediction. The number of LSTM units is 64. The fully connected module has one hidden layer with 40 hidden nodes and the leaky-ReLU activation function. The dropout probability of the hidden nodes is 0.3.

NVIDIA RTX2080 GPU. The rest of this section explains the variables and procedures of Algorithm. 1.

A. Notations and Internal Variables

The number of estimated poses from current frame t and their (tentative) index are denoted as C_t and j, respectively. The tracking system manages at most F_{max} *filters* to track multi-person poses, each of which has a unique track ID k and handles P particles. Particle n of filter k comprises an L-sized queue that stores previous poses $\boldsymbol{z}_{t-L:t-1}^{(k,n)}$. During the tracking process at t, only *active* filters are executed. The activation and deactivation of filters are controlled by *lifetime counts* l_k which manages the appearance and disappearance of people. A set of active filters at t are denoted as $\mathcal{K}_t$ ($|\mathcal{K}_t| := F_t \leq F_{max}$). Note that all of the filters are inactive when the algorithm just starts (ℓ1). The experiments in Sec. V used $P = 300$, $L = 10$, and $F_{max} = 100$.

B. Module Procedures

1) Pose Prediction: (ℓ3) $F_t \times P$ sequences of poses $\boldsymbol{z}_{t-L:t-1}^{(k,n)}$ are collected from the active filters and then (ℓ4) input to the predictor to output $F_t \times P$ predicted poses $\boldsymbol{z}_t^{(k,n)}$. Note that these procedures are not executed in case where there are no active filters (e.g., $t = 1$).

2) Pose Estimation: (ℓ5) C_t poses $\boldsymbol{x}_t^{(j)}$ are estimated by inputting a current RGB image to the deep CNN model of *Simple-Baseline* [15]. Here we used an opensource codebase[1] to realize the CNN model and completely followed the experimental settings described in the literature [15] to train the model.

3) Greedy Matching: (ℓ6) In total, $C_t \times F_t \times P$ OKS values $d_{OKS}^{(j,k,n)}$ are calculated from $F_t \times P$ predicted poses $\boldsymbol{z}_t^{(k,n)}$ and C_t estimated poses $\boldsymbol{x}_t^{(j)}$. (ℓ7) The shape of this OKS tensor is transformed to $C_t \times F_t$ by taking the weight average along the n-axis to calculate a matching score matrix[2]. Then, (ℓ8) this matrix is input to the bipartite matching procedures. Formed pairs whose matching score is less than a given threshold (i.e., two poses are distant to each other) are removed to prevent inappropriate matching. Here, j'_k and k'_j denote the indices of counterparts of filter k and pose j, respectively. Note that these variables take negative values when no counterparts are assigned due to shortage and overage of the active filters, and the thresholding. (ℓ10–11) If k'_j has a valid value for pose j, a tuple of $(\boldsymbol{x}_t^{(j)}, k'_j)$ is output as the tracking result. (ℓ12) If k'_j has an invalid value, (13) a new filter k_{new} is activated and (17) this new index is output with $\boldsymbol{x}_t^{(j)}$. (ℓ15–16) The newly activated queue states are initialized with $\boldsymbol{x}_t^{(j)}$ and zeros as invisible keypoints.

4) Particle Updates: (ℓ19) The newest states $\boldsymbol{z}_t^{(k,n)}$ are pushed to the queues (thereby removing the oldest ones). (ℓ20–21) If j'_k has a valid value, the queues of filter k are updated by probabilistic resampling and selection according to (3) and (6), respectively. This process is not executed when $j'_k < 0$ because no information is available to update the confidences of the hypotheses. (ℓ22, ℓ24) The lifetime count l_k is incremented or decremented according the existence of filter k's counterpart. (ℓ25–26) If ℓ_k obtains zeros, a person tracked by filter k tracks is considered to have disappeared completely; thus, the filter is deactivated.

V. Experiments

A. Comparison to State-of-the-art Method

The main objective of this experiment was to demonstrate that the proposed method has advantages over the state-of-the-art pose tracking method [12]–[17].

Training and evaluation were conducted using the PoseTrack2018 dataset. The annotations include 17 body keypoint locations and unique track IDs for multiple persons in the videos. Training data $\mathcal{D}$ for were created from the training annotation data, and the probabilistic neural pose predictor (Fig. 4) was trained using the Adam optimizer [27]. Here, the learning rate was 10^{-3}, and mini-batch size was 30.

We evaluated the performance of the proposed tracking method using the validation data and official evaluation tool[3]. Table II summarizes the MOTA scores obtained by the baseline and proposed method. The best result obtained by the proposed method was a 66.2 MOTA score. The proposed method outperformed both the winner of the ECCV18 PoseTrack Challenge [15] (MOTA: 65.4) and the recent state-of-the-art method [17] (MOTA: 65.7).

[1] `https://github.com/mks0601/TF-SimpleHumanPose`

[2] We determine particle weights as followings: the weights of the top e% particles with higher OKS are set to 1; the remaining weights are set to 0. The *eliteness* ratio e is 15%.

Algorithm 1: Proposed Multi-person Pose Tracking Method

```
DeactivateAllFilters()
for $t \leftarrow 1$ to $\infty$ do
    // (1) Pose Prediction
    $\{\{\boldsymbol{z}_{t-L:t-1}^{(k,n)}\}_{n=1}^{P}\}_{k \in \mathcal{K}_t} \leftarrow$ GetActiveFilterStates()
    $\{\{\boldsymbol{z}_{t}^{(k,n)}\}_{n}\}_{k} \leftarrow$ PredictPose($\{\{\boldsymbol{z}_{t-L:t-1}^{(k,n)}\}_{n}\}_{k}$)
    // (2) Pose Estimation
    $\{\boldsymbol{x}_{t}^{(j)}\}_{j=1}^{C_t} \leftarrow$ EstimatePosesFromCurrentFrame()
    // (3) Greedy Matching
    $\{\{\{d_{OKS}^{(j,k,n)}\}_{j}\}_{k}\}_{n} \leftarrow$ CalcOKS$\left(\{\{\boldsymbol{z}_{t}^{(k,n)}\}_{n}\}_{k}, \{\boldsymbol{x}_{t}^{(j)}\}_{j}\right)$
    $\{\{d_{score}^{(j,k)}\}_{j}\}_{k} \leftarrow$ CalcScore$\left(\{\{\{d_{OKS}^{(j,k,n)}\}_{j}\}_{k}\}_{n}\right)$
    $(\{j'_k\}_k, \{k'_j\}_j) \leftarrow$ BipartiteMatch$\left(\{\{d_{score}^{(j,k)}\}_{j}\}_{k}\right)$
    for $j \leftarrow 1$ to $C_t$ do
        if $k'_j > -1$ then
            Output$\left(\boldsymbol{x}_{t}^{(j)}, k'_j\right)$
        else
            $k_{new} \leftarrow$ ActivateNewFilter()
            $l_k \leftarrow 1$
            $\{\boldsymbol{z}_{t}^{(k_{new},n)}\}_{n} \leftarrow \{\boldsymbol{x}_{t}^{(j)}\}_{n}$
            $\{\boldsymbol{z}_{t-L+1:t-1}^{(k_{new},n)}\}_{n} \leftarrow \boldsymbol{0}$
            Output$\left(\boldsymbol{x}_{t}^{(j)}, k_{new}\right)$
    // (4) Particle Update
    foreach Active Filters $k \in \mathcal{K}_t$ do
        $\{\boldsymbol{z}_{t-L+1:t}^{(k,n)}\}_{n} \leftarrow$ Push$\left(\{\boldsymbol{z}_{t}^{(k,n)}\}_{n}, \{\boldsymbol{z}_{t-L:t-1}^{(k,n)}\}_{n}\right)$
        if $j'_k > -1$ then
            $\{\boldsymbol{z}_{t-L+1:t}^{(k,n)}\}_{n} \leftarrow$
              ResampleAndSelect$\left(\{\boldsymbol{z}_{t-L+1:t}^{(k,n)}\}_{n}, \boldsymbol{x}_{t-L+1:t}^{(j'_k)}\right)$
            $l_k \leftarrow \mathrm{Min}(l_k + 1, 30)$
        else
            $l_k \leftarrow l_k - 1$
            if $l_k < 0$ then
                DeactivateFilter($k$)
```

B. Ablation Study

This experiment was performed to clarify which component of the proposed method contributes to the overall improvement. Here, variants of our method were prepared: *1)* both or either types of uncertainty were invalidated, and *2)* the length of time-sequence L was varied. Note that we removed epistemic uncertainty modeling by deactivating dropout in the fully-connected layer of the predictor, and heteroscedastic aleatoric uncertainty was removed by fixing the value of $\boldsymbol{\sigma}$.

As a principle metric, we focused on num_switches (an intermediate variable used to calculate MOTA) rather than MOTA in this experiment. MOTA comprises of three variables: num_switches as tracking error counts, and num_misses and num_false_positives as pose estimation error counts (see the source code of the evaluation tool or [22]). We can distinctly compare the tracking performances by inputting shared pose estimation results to different tracking models and focusing on num_switches. Specifically, we focused on the total num_swithces of the most frequently appeared keypoint (i.e., nose).

We also included an evaluation of the *Simple-Baseline* method in this experiment to clearly demonstrate that the above state-of-the-art result was achieved by the proposed tracking approach rather than other factors (e.g., more accurate pose estimation). Since an evaluation of num_switches was not conducted in a previous study [15], we used our own implementation of the baseline method for this test.

The results of this ablation study are summarized in Table II, which demonstrates that involving both types of uncertainty contributes to performance improvement. Utilizing the time sequence input with LSTM is also effective. By referring to longer pose contexts (e.g., $L = 10, 15$), the predictor can infer more plausibile hypotheses, which results in overall performance improvement. However, parameter L should be determined carefully because it affects computational complexity (i.e., the memory size and computational time of sequential LSTM forwarding) and training stability. A comparison of results obtained by *Simple-Baseline* indicates that our best result achieves approximately 50% of the baseline's tracking errors. Fig. 5 shows some visualized results obtained by the baseline and proposed method.

VI. Related Work

Recently, uncertainty-aware modeling has been receiving increasing attention in robotics studies, including reinforcement learning [28], [29], imitation learning [30], [31], motion and path planning [32], [33], and unfamiliar situation detection [34]. For example, Refs. [28], [29] also incorporated the two types of uncertainty to forward dynamics modeling, thereby solving the major inherent problem of model-based reinforcement learning, i.e., the model-bias problem. Approaches that are similar to our method have been proposed in previous tracking and SMC studies, such as SMC based multi-object tracking [35] and the combination of SMC and neural networks [24], [36], [37]. However, to the best of our knowledge, the proposed method is the first to apply SMC to multi-person pose tracking using a probabilistic neural network that incorporates the two types of uncertainty, and the results provide a new state-of-the-art in the challenging and competitive benchmark task.

VII. Conclusion

In this paper, we have proposed an SMC-based multi-person pose tracking method that utilizes a probabilistic neural pose predictor. By incorporating epistemic uncertainty and heteroscedastic aleatoric uncertainty, our pose predictor can propose diverse and plausible hypotheses for the next frame poses, thereby solving the fragility of the sole hypothesis approach of the state-of-the-art baseline [15]. In addition, recurrent neural modeling is introduced, which exploits long context information to make prediction robust against complex situations, such as cases with significant occlusions. The experimental results demonstrate that the proposed method achieves a state-of-the art MOTA score of

[3] https://github.com/leonid-pishchulin/poseval

TABLE I

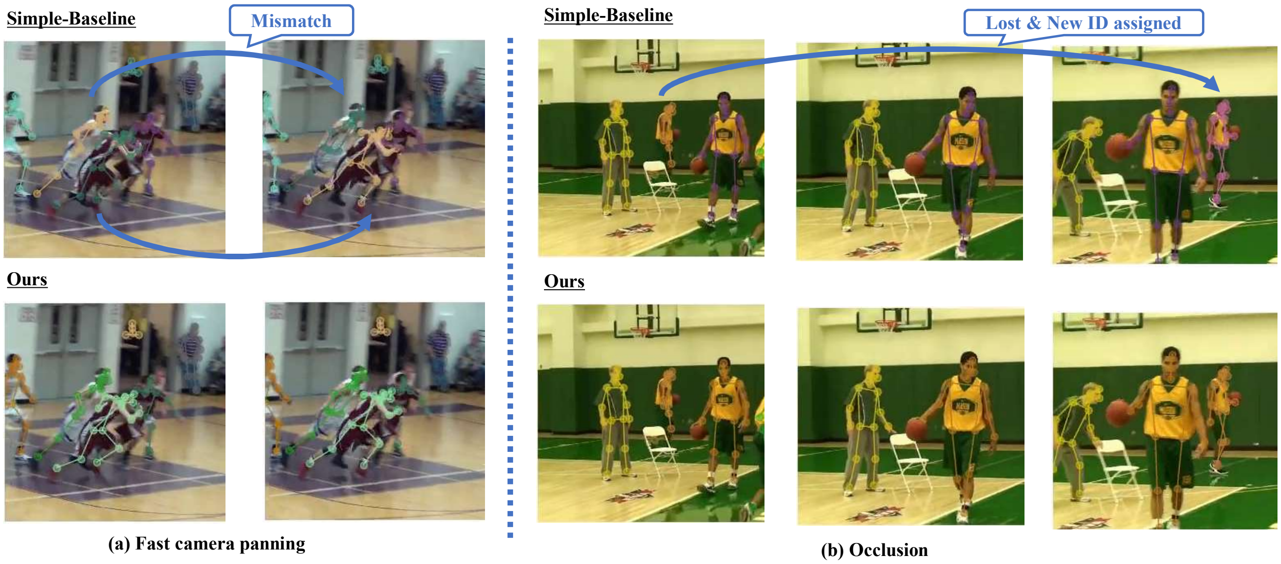
MULTI-PERSON POSE TRACKING PERFORMANCE ON POSETRACK2018 VALIDATION DATASET.

Method	MOTA Head	MOTA Sho.	MOTA Elb.	MOTA Wri.	MOTA Hip	MOTA Knee	MOTA Ank.	MOTA Total
MDPN-152-A [14]	50.9	55.5	65.0	49.0	48.7	50.5	45.1	50.6
Detect-and-Track [12]	61.7	65.5	57.3	45.7	54.3	53.1	45.7	55.2
Pose Flow [13]	59.8	67.0	59.8	51.6	60.0	58.4	50.5	58.3
STAF [17]	-	-	-	-	-	-	-	60.9
Simple-Baseline [15]	73.9	75.9	63.7	56.1	65.5	65.1	53.6	65.4
TML++ [16]	76.0	76.9	66.1	56.4	65.1	61.6	52.4	65.7
Ours	72.5	76.5	66.8	58.6	63.2	65.2	57.0	**66.2**

TABLE II

RESULTS OF ABLATION STUDY.

Method	Aleatoric Uncertainty	Epistemic Uncertainty	Time-Seq. Length L	MOTA Total	num_switches
Ours	✓	✓	15	66.1	232
Ours	✓	✓	10	66.2	**213**
Ours	✓	✓	7	66.1	242
Ours	✓	✓	3	66.0	266
Ours	✓		10	66.1	252
Ours		✓	10	66.0	285
Ours			10	65.9	303
Simple-Baseline [15]	-	-	-	65.4	407



(a) Fast camera panning

(b) Occlusion

Fig. 5. Visualized results of *Simple-Baseline* and proposed method. Estimated poses are color-coded according to track IDs. The baseline method fails to track some poses due to fast pose changes and occluions. In such difficult cases, the proposed method can track the poses successfully.

66.2 with approximately 50% reduction in tracking errors from the baseline.

Other sophisticated uncertainty modeling could improve the proposed method's overall performance, such as employing α-divergence dropout [38] and neural network ensembles [28] for epistemic uncertainty modeling, as well as replacing the output Gaussian distribution with a Gaussian Mixture Model by introducing Mixture Density Networks [24], [39] for aleatoric uncertainty modeling. In addition, inputting optical-flows to a neural predictor seems promising approach to utilize visual information, and the end-to-end supervised learning of SMC [36], [37] is appealing to automatically design an effective likelihood function $p(\boldsymbol{x}_T|\boldsymbol{z}_{1:T}^{(n)}, \boldsymbol{x}_{1:T-1})$, which could improve the validity of the matching score compared to existing hand-crafted metrics, e.g., OKS .

The uncertainty- and context-aware concept for SMC proposed in this paper is simple, general, and strong, and we exepct that this concept is applicable to a variety of SMC-based robotics tasks, such as SLAM. Other possible future work could include more challenging tracking tasks, such as 3D human pose tracking [40] and dense human pose tracking [41].

REFERENCES

[1] K. Behrendt, L. Novak, and R. Botros, "A deep learning approach to traffic lights: Detection, tracking, and classification," in *IEEE International Conference on Robotics and Automation*, 2017.

[2] S. Verma, Y. H. Eng, H. X. Kong, H. Andersen, M. Meghjani, W. K. Leong, X. Shen, C. Zhang, M. H. Ang, and D. Rus, "Vehicle detection, tracking and behavior analysis in urban driving environments using road context," in *IEEE International Conference on Robotics and Automation*, 2018.

[3] A. Buyval, A. Gabdullin, R. Mustafin, and I. Shimchik, "Realtime vehicle and pedestrian tracking for Didi udacity self-driving car challenge," in *IEEE International Conference on Robotics and Automation*, 2018.

[4] S. Sharma, J. A. Ansari, J. K. Murthy, and K. M. Krishna, "Beyond pixels: Leveraging geometry and shape cues for online multi-object tracking," in *IEEE International Conference on Robotics and Automation*, 2018.

[5] A. Concha and J. Civera, "DPPTAM: Dense piecewise planar tracking and mapping from a monocular sequence," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2015.

[6] P. Liang, Y. Wu, H. Lu, L. Wang, C. Liao, and H. Ling, "Planar object tracking in the wild: A benchmark," in *IEEE International Conference on Robotics and Automation*, 2018.

[7] C. Rauch, T. Hospedales, J. Shotton, and M. Fallon, "Visual articulated tracking in the presence of occlusions," in *IEEE International Conference on Robotics and Automation*, 2018.

[8] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2D pose estimation using part affinity fields," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[9] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *European Conference on Computer Vision*, 2014.

[10] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, "2D human pose estimation: New benchmark and state of the art analysis," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[11] U. Iqbal, A. Milan, and J. Gall, "Posetrack: Joint multi-person pose estimation and tracking," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[12] R. Girdhar, G. Gkioxari, L. Torresani, M. Paluri, and D. Tran, "Detect-and-track: Efficient pose estimation in videos," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 350–359, 2018.

[13] Y. Xiu, J. Li, H. Wang, Y. Fang, and C. Lu, "Pose flow: Efficient online pose tracking," *arXiv preprint arXiv:1802.00977*, 2018.

[14] H. Guo, T. Tang, G. Luo, R. Chen, Y. Lu, and L. Wen, "Multi-domain pose network for multi-person pose estimation and tracking," in *European Conference on Computer Vision*, 2018.

[15] B. Xiao, H. Wu, and Y. Wei, "Simple baselines for human pose estimation and tracking," in *European Conference on Computer Vision*, 2018.

[16] J. Hwang, J. Lee, S. Park, and N. Kwak, "Pose estimator and tracker using temporal flow maps for limbs," *arXiv preprint arXiv:1905.09500*, 2019.

[17] Y. Raaj, H. Idrees, G. Hidalgo, and Y. Sheikh, "Efficient online multi-person 2D pose tracking with recurrent spatio-temporal affinity fields," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4620–4628, 2019.

[18] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "PWC-Net: Cnns for optical flow using pyramid, warping, and cost volume," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[19] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?," in *Advances in neural information processing systems*, 2017.

[20] J. Janai, F. Guney, A. Ranjan, M. Black, and A. Geiger, "Unsupervised learning of multi-frame optical flow with occlusions," in *European Conference on Computer Vision*, 2018.

[21] Z. Ren, O. Gallo, D. Sun, M.-H. Yang, E. Sudderth, and J. Kautz, "A fusion approach for multi-frame optical flow estimation," in *IEEE Winter Conference on Applications of Computer Vision*, 2019.

[22] K. Bernardin, A. Elbs, and R. Stiefelhagen, "Multiple object tracking performance metrics and evaluation in a smart room environment," in *IEEE International Workshop on Visual Surveillance, in conjunction with ECCV*, 2006.

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

[24] S. S. Gu, Z. Ghahramani, and R. E. Turner, "Neural adaptive sequential monte carlo," in *Advances in Neural Information Processing Systems*, 2015.

[25] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *international conference on machine learning*, pp. 1050–1059, 2016.

[26] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. Software available from tensorflow.org.

[27] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations*, 2014.

[28] K. Chua, R. Calandra, R. McAllister, and S. Levine, "Deep reinforcement learning in a handful of trials using probabilistic dynamics models," in *Advances in Neural Information Processing Systems*, pp. 4754–4765, 2018.

[29] M. Okada and T. Taniguchi, "Variational inference MPC for bayesian model-based reinforcement learning," in *Conference on Robot Learning*, 2019.

[30] S. Thakur, H. van Hoof, J. C. G. Higuera, D. Precup, and D. Meger, "Uncertainty aware learning from demonstrations in multiple contexts using bayesian neural networks," in *IEEE International Conference on Robotics and Automation*, 2019.

[31] J. Silvério, Y. Huang, F. J. Abu-Dakka, L. Rozo, and D. G. Caldwell, "Uncertainty-aware imitation learning using kernelized movement primitives," in *IEEE International Conference on Robotics and Automation*, 2019.

[32] M. Bowman, S. Li, and X. Zhang, "Intent-uncertainty-aware grasp planning for robust robot assistance in telemanipulation," in *IEEE International Conference on Robotics and Automation*, 2019.

[33] L. Nardi and C. Stachniss, "Uncertainty-aware path planning for navigation on road networks using augmented MDPs," in *IEEE International Conference on Robotics and Automation*, 2019.

[34] R. McAllister, G. Kahn, J. Clune, and S. Levine, "Robustness to out-of-distribution inputs via task-aware generative uncertainty," in *IEEE International Conference on Robotics and Automation*, 2019.

[35] M. Jaward, L. Mihaylova, N. Canagarajah, and D. Bull, "Multiple object tracking using particle filters," in *IEEE Aerospace Conference*, 2006.

[36] R. Jonschkowski, D. Rastogi, and O. Brock, "Differentiable particle filters: End-to-end learning with algorithmic priors," in *Robotics: Science and Systems*, 2018.

[37] P. Karkus, D. Hsu, and W. S. Lee, "Particle filter networks with application to visual localization," in *Conference on Robot Learning*, 2018.

[38] Y. Li and Y. Gal, "Dropout inference in bayesian neural networks with alpha-divergences," in *International Conference on Machine Learning*, 2017.

[39] C. M. Bishop, "Mixture density networks," *Tech. Rep. NCRG/94/004, Neural Computing Research Group, Aston University, 1994.*, 1994.

[40] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, "Human3.6M: Large scale datasets and predictive methods for 3D human sensing in natural environments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2014.

[41] R. Alp Güler, N. Neverova, and I. Kokkinos, "Densepose: Dense human pose estimation in the wild," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.