# "Let's get ready to bundle!": Crowd-level Human Keypoint Tracking

*Thomas Golda*

Vision and Fusion Laboratory
Institute for Anthropomatics
Karlsruhe Institute of Technology (KIT), Germany
thomas.golda@kit.edu

## Abstract

This work examines the suitability of a state-of-the-art human pose tracking method for application within surveillance scenarios and focuses on public places in urban areas that tend to suffer from crowdedness, such as city centers. Starting with a short introduction to motivate keypoint tracking in surveillance applications, this report will present details about the adapted method, which follows an LSTM-based approach. Afterwards, different changes that had to be incorporated in order to successfully apply the given method to our target setting will be presented. Finally, various experiments will show how the chosen method performs, based on experiments with simulated data.

## 1 Introduction

Anomaly detection amongst other strongly related topics like outlier and novelty detection, plays an important role in various research fields as network traffic monitoring, time series analysis, medical image analysis, and video surveillance. However, when talking about anomalies in the context of video surveillance the understanding of what an anomaly actually looks like can differ strongly

between applications. For instance, an anomaly can be an abandoned suitcase at a public place, a vehicle driving through a pedestrian zone or suspicious or salient behaving people. With the rising interest in unconstrained activity and action recognition[1] in urban settings and application-oriented research for video surveillance, the task of detecting unusual behavior gets more and more attention. This report focuses on human-centered features in order to distinguish between usual and unusual behavior. Therefore, on a basis of person skeletons provided by human pose estimators, we try to create corresponding body joint tracklets, as proposed in [3]. In order to do so, different keypoints corresponding to a certain person have to be tracked over time to obtain the desired tracklets, which can afterwards be used for further behavioral analysis.

# 2 Tracking Keypoints in the Wild

## 2.1 Human Pose Estimation

Human Pose Estimation describes the problem of estimating a skeletal representation of a person based on information gathered using certain types of sensors. The skeletal representation is typically represented as a graph $G = (V, E)$ where $V \subset \mathbb{R}^n$ is a set of keypoints and $E \subset V \times V$ is a set of edges connecting various keypoints. Depending on the chosen skeletal model the graph can be seen as a tree. In general, Human Pose Estimation considers sensors used in classical video cameras or depth cameras delivering RGB or RGB-D information respectively. However, in the field of video surveillance RGB-D cameras are rarely applied, which is due to price and often large distances of subjects to the mounted camera. Therefore, this work focuses on the case of 2D skeletons obtained using classical cameras and RGB data. With this constraint, the resulting skeletons produced by human pose estimation algorithms consequently consist of keypoints in a two-dimensional space with $V \subset \mathbb{R}^2$.

[1] https://actev.nist.gov/

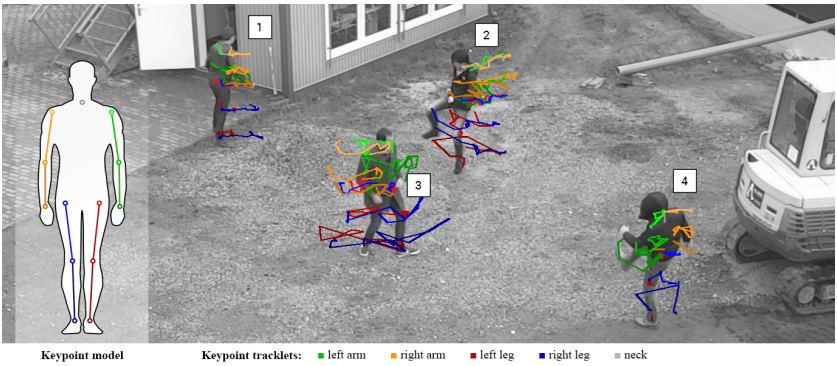


**Figure 2.1**: Crop of exemplary network camera footage. This is an example of keypoint bundles extracted using a simple tracking-by-detection approach. Based on detections in $t$ consecutive frames, for every person keypoint tracklets were extracted. The different resulting body parts of the keypoint bundles are highlighted in different colors for visualization purposes. In addition to that, the lightness of those colors encodes the proximity of the corresponding keypoint to the pose model center, i.e. shoulders and hips have lighter colors than wrists and ankles. For a better understanding, the underlying human pose model with the corresponding colors is shown on the left.

## 2.2 Human Keypoint Tracking

### 2.2.1 Introduction

In general, the process of observing a single entity over time is referred to as *tracking*. Classical tasks are object and pedestrian tracking, where for a given timestep detections of single entities are associated with corresponding detections in earlier timesteps. This kind of tracking is also known as tracking-by-detection. Typically, such entities are represented by bounding boxes that enclose the subject. The idea of tracking bounding boxes can be extended to the tracking of single points over time, which is subject of this report.
Given a video or image sequence showing pedestrians our goal is to obtain a set of tracklets $K_i$ for every single person $i$ describing its movement over a short period of time. The set of tracklets is defined as

$$K_i = \{(\mathbf{k}_1^1, ..., \mathbf{k}_t^1), (\mathbf{k}_1^2, ..., \mathbf{k}_t^2), ..., (\mathbf{k}_1^n, ..., \mathbf{k}_t^n)\} \tag{2.1}$$

where $\mathbf{k}_t^n \in \mathbb{R}^2$ is the $n$-th two-dimensional keypoint at timestep $t$. Hence, each item in $K_i$ is a single corresponding keypoint tracklet. We refer to $K_i$ as a *keypoint bundle*. Such keypoint bundles can be obtained using various approaches. In this work, we assemble these bundles by tracking keypoints directly in order to create less noisy tracklets than those acquired by following an approach just based on person detection. Figure 2.1 shows an example for a scene and corresponding keypoint bundles.

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

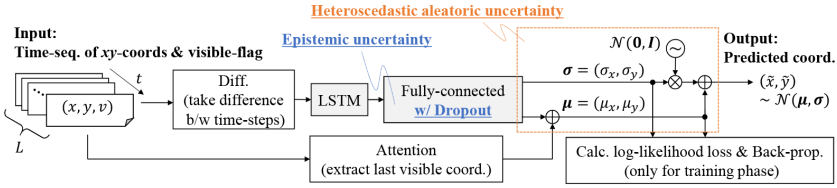


**Figure 2.2**: The figure shows the structure of the probabilistic Neural Pose Predictor by Okada et al. [6]. Whereas the epistemic uncertainty is modelled using dropout in the final fully-connected layer of the LSTM-architecture, the aleatoric uncertainty is incorporated using an sampling process from a normal distribution.

Okada et al. [6] followed the approach of tracking keypoints by using an LSTM-based network architecture called *(probabilistic) Neural Pose Predictor* (NPP). In general, the NPP is part of a classical particle filter where it is responsible for modeling the motion model of humans and hence is an essential part of the prediction step within the update process of the bayesian filter. The main contribution of Okada et al. is the incorporation of different uncertainties within the NPP, namely heteroscedastic aleatoric uncertainty (inherent system stochasticity) and epistemic uncertainty (model uncertainty due to limited data). Based on a fixed time horizon of ten timesteps the position at the next time step is predicted, which is done for every single keypoint and every detected pedestrian. To be exact, input to the NPP is a sequence of differences between consequent timesteps in a single track. The LSTM then predicts values for the

mean and standard deviation, which are used to define a normal distribution that is afterwards used to sample an actual estimate for the predicted difference. The explained procedure is also shown in Figure 2.2.

# 3 From One to Many

We adapted the approach of Okada et al. [6] in order to evaluate its performance on crowd-level applications, which is a topic rarely covered in the computer vision community. It is obvious that this is a very challenging task, due to various problems like the ambiguity in appearance and lots of dynamic occlusions. Figure 3.1 shows how such a crowded surveillance situation can look like. Although pedestrians in this image are at a quite large scale, already the proximity to other pedestrians can lead pose estimators to estimate false poses. This is supported by many people wearing dark clothes, which due to less

**Figure 3.1**: Example of a crowded scene. People in such monitored areas are typically even of smaller size, however independent of the scale the same problems occur: ambiguity in appearance and strongly dynamic occlusions.

contrast is very challenging for pose estimation methods. However, tackling it in this way offers more detailed information about the behavior of single subjects and still keeps a high level of data privacy at the same time. This is especially useful when it is used in assistance systems for video surveillance, where the system just creates hints for a human operator. In our experiments we therefore are mainly interested in the overall keypoint tracking performance.

## 3.1 SyMPose

Especially designed for being a dataset consisting of many people and providing detailed keypoint and tracking information for each individual, we decided to use an internal synthetic dataset for for training the NPP. This dataset was created using an own optimized version of the JTA modification published by Fabbri et al. [2] for the popular video game *Grand Theft Auto V*. It was initially designed for the task of domain adaption between synthetic and real-world domain [4]. Table 3.1 gives a short comparison of some key figures of both datasets.

**Table 3.1**: Comparison of two synthetic datasets. Although SyMPose is smaller compared to JTA, it is more comparable to a surveillance situation due to the higher viewing position of the camera and the overall higher average pedestrian density (ppf) within a frame. Furthermore, the maximum number of pedestrians per frame is more than twice as high than in JTA (130 versus 60 pedestrians).

| dataset | # scenes | # frames | # poses | ppf | setting |
|---|---|---|---|---|---|
| JTA [2] | 512 | 460,800 | ca. 10m | 21 | urban |
| SyMPose [4] | 21 | 19,900 | ca. 1.3m | 68 | urban |

In order to get a better idea of how scenes from SyMPose look like, an example taken from the dataset is displayed in Figure 3.2. Most noticeable is the much higher number of pedestrians in the scene. Furthermore, the viewing perspective and camera mounting height are more consistent throughout all recorded scenes and can be hence better compared to a real-world application scenario. These were the main reasons to use this dataset, since comparable real-world datasets labeled with keypoint information in this setting do not exist or are publicly just not available. SyMPose was used throughout this work for training and evaluation.

**Figure 3.2**: Synthetic crowded scene. SyMPose consists of around two dozen scenes showing different places filled with many people. All scenes are recorded from a higher-level camera that should simulate the target scenario in urban settings. The underlying skeleton model consists of a subset of the JTA model and was designed to mimic the model used in CrowdPose [5].

## 3.2 Whole Pose Inference

In the initial proposed method, Okada et al. [6] designed the NPP to work on single keypoints. This, however, includes the assumption that every keypoint has the identical, independent motion model. Apparently, the human skeleton is restricted in its configurations and the movement of a single joint (i.e. keypoint) is strongly dependent on its adjacent joints. We therefore extended the single-keypoint approach to a multi-keyoint approach and compare it with the former. This is done by providing a feature vector to the NPP consisting of information for all keypoints instead just for a single one.

## 3.3 Expansion of Inter-Frame Distances

Since recorded data in such surveillance scenarios is dominated by people at small scale and slow velocities, the relative movement between two frames is

also quite small compared to a close-up shot of only few people. With a typical recording frame rate of 25 or 30 frames per second this results in distances of only few pixels between two poses or frames. This is a challenging problem for the NPP, which was designed to work with multi-person scenarios yet at larger scale.

### 3.3.1 Affine Transformation

A first way to tackle this problem is to rescale tracks by a certain factor $\alpha \in \mathbb{R}^+$. Typically, such a value is determined by normalizing the data using its mean and standard deviation. Doing so results in an affine transformation and will affect the size of the poses and hence the position of the keypoints as well as the displacement between timesteps. The benefit of this approach is, that still the full frame rate and therefore the full range of information can be used. However, this transformation comes with additional lightweight computations and has to be reverted afterwards to extract the real coordinates of the predicted position.

### 3.3.2 Reduction of Frame Rate

Another orthogonal way to tackle this problem is to skip some intermediate frames, and hence sample a given sequence. This will have a similar effect as the first way, however it is associated with a loss of information since fast movements could be overlooked. Furthermore, the prediction frame rate is also affected by the new resulting frame rate. The benefit of this approach is that no transformation is needed and the predicted pose can be used directly.

# 4 Evaluation

In the following, we will examine the performance of the changes and extensions presented in Section 3. First the used evaluation metrics will shortly be presented and analyzed. These metrics were chosen, since they were also used by Okada et al. [6].

## 4.1 Evaluation Metrics

### 4.1.1 Mean Squared Error

The Mean Squared Error (MSE) is a widely used metric to measure the deviation from a certain target. In the context of human poses it is defined as

$$\text{MSE} = \frac{1}{2N}\sum_{i=1}^{N}(\hat{k}_{i,x} - k_{i,x})^2 + (\hat{k}_{i,y} - k_{i,y})^2 \tag{4.1}$$

where $N \in \mathbb{N}$ is the number of keypoints defined by the underlying pose skeleton model, $\hat{\mathbf{k}}_i = (\hat{k}_{i,x}, \hat{k}_{i,y})$ is the ground truth position for the i-th keypoint and $\mathbf{k}_i = (k_{i,x}, k_{i,y})$ is the position of the prediction of the corresponding keypoint. It is obvious, that the value of the MSE is strongly dependent on the scale of two compared poses.
To tackle this scale dependency, we us a normalized version of the MSE. By dividing the MSE by the squared scale factor $\alpha$ we obtain the following equation

$$\alpha\text{MSE} = \frac{\text{MSE}}{\alpha^2} = \frac{1}{2N\alpha^2}\sum_{i=1}^{N}(\hat{k}_{i,x} - k_{i,x})^2 + (\hat{k}_{i,y} - k_{i,y})^2 \tag{4.2}$$

### 4.1.2 Object Keypoint Similarity

The Object Keypoint Similarity (OKS) [1] is a metric introduced by the COCO consortium for the purpose of evaluating the pose detection performance. It is an attempt to create a metric that can be compared to the Intersection of Union, which is especially known for its application in bounding box detection scenarios.

$$\text{OKS} = \frac{\sum_i [exp(\frac{-d_i^2}{2 \cdot s^2 \kappa_i^2}) \cdot \delta(v_i > 0)]}{\sum_i [\delta(v_i > 0)]} \tag{4.3}$$

where $N$ is the number of keypoints in the underlying skeleton model, $s$ is the square root of the area of the smallest bounding box enclosing all corresponding keypoints and $d$ is the actual distance between two poses. Furthermore, $v_i$ is the

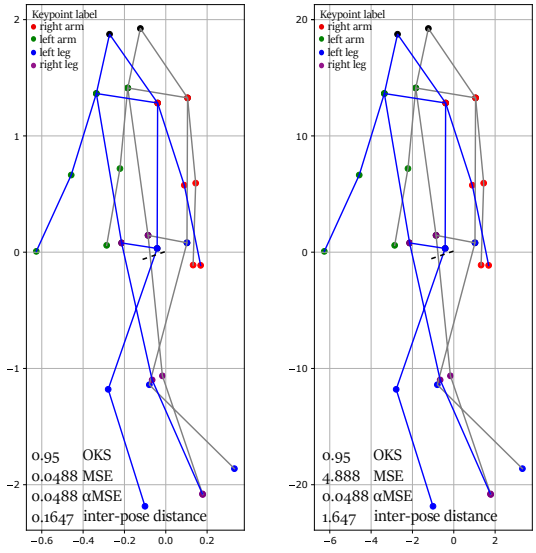


**Figure 4.1**: Visual and quantitative comparison for poses at different scales and the effects on the evaluation metrics OKS and MSE. Inter-pose distance and MSE are directly effected by the rescaling of poses, whereas OKS and $\alpha$MSE stay the same due to their scale invariance.

visibility of the i-th keypoint and $\kappa_i$ are manually obtained keypoint constants that model the annotation uncertainty of human annotators in the actual COCO dataset. Due to the way the OKS is defined, the metric is scale invariant. This comes from the fact that the actual keypoint positions are combined with the occupied area, which can be seen as a certain kind of normalization.

Figure 4.1 shows two examples of pose comparisons. Both examples look very similar, however they are at different scale. While the OKS stays the same even at larger scale, the MSE is higher the bigger the scale is.

## 4.2 Evaluation Results

First, we take a look on the presented changes and extensions. Table 4.2 summarizes the results for single- and multi-keypoint approach, as well as both inter-frame distance expansion ways. In order to compare MSE results between different scales, we report the $\alpha$MSE introduced in Section 4.1.

**Single- vs. Multi-Keypoint** Throughout all examined configurations the single-keypoint approach achieves similar or even slightly better results compared to the multi-keypoint approach. With increasing scaling factor $\alpha$ both the single-keypoint and multi-keypoint approach improve with regard to the OKS and $\alpha$MSE metric. Furthermore, with the the expanding absolute gap between timesteps, the single-keypoint approach builds up a lead over the multi-keypoint method. This, however underlines that using a NPP just trained on single keypoints might be sufficient and does not benefit from the additional information that is available when predicting whole poses.

**Affine Transformation** In order to evaluate the impact of the affine transformation, different values for $\alpha$ were chosen, namely $\alpha = 10^0$ (i.e. no scaling), $\alpha = 10^1$, $\alpha = 10^2$, $\alpha = 10^3$, and $\alpha = 10^4$. The data distribution is characterized by its mean $\mu$ and its standard deviation $\sigma$ which are given in Table 4.1 for every examined frame rate at scale $\alpha = 1$. The values show that the normalization would yield a scaling factor $\alpha$ between roughly around 2 and 12, and hence would fall into the range of our experiments with $\alpha = 1$ and $\alpha = 10$. As mentioned above, independent from the chosen NPP-approach (single- vs. multi-keypoint), with increasing values of $\alpha$ the prediction performance first improves and drops for larger values of $\alpha$. The results show, that choosing a larger rescaling factor than the factor induced by normalization is beneficial to the performance of the NPP and leads to significant improvement. The same observation can not only be made for the full, but also for the reduced frame rate experiments. One conceivable reason for this behaviour might be that the LSTM architecture of the NPP struggles with very small values close to zero. This is likely to happen since input and output of the LSTM are differences

**Table 4.1**: Mean and standard deviation for the distance between consequent poses. Since intermediate frames were dropped for the reduced experiments, the actual distance between consequent frames was increased throughout the dataset.

| **Framerate [fps]** | **30** | **6** | **3** |
|---|---|---|---|
| $\mu$ | 0.1036 | 0.3644 | 0.6367 |
| $\sigma$ | 0.0802 | 0.2685 | 0.4892 |

**Table 4.2**: Evaluation results on the SyMPose test set. With decreasing number of frames the performance of the pose prediction drops. This is logical since less information about the actual motion is available. Furthermore, with an increasing factor $\alpha$ the prediction performance improves up to a value of around $10^2$ to $10^3$ and begins to drop afterwards. This result shows the weakness of LSTMs when used with small values.

| **Framerate [fps]** | | **30** | | **6** | | **3** | |
|---|---|---|---|---|---|---|---|
| | $\alpha$ | **OKS** | **$\alpha$MSE** | **OKS** | **$\alpha$MSE** | **OKS** | **$\alpha$MSE** |
| single | $10^0$ | 0.989 | 0.02 | 0.75 | 0.23 | 0.58 | 0.73 |
| multi | | 0.989 | 0.02 | 0.75 | 0.23 | 0.52 | 0.78 |
| single | $10^1$ | 0.990 | 0.02 | 0.87 | 0.13 | 0.78 | 0.26 |
| multi | | 0.990 | 0.02 | 0.86 | 0.13 | 0.79 | 0.28 |
| single | $10^2$ | 0.996 | **0.01** | **0.93** | 0.07 | **0.83** | 0.21 |
| multi | | 0.994 | **0.01** | 0.91 | 0.09 | 0.80 | 0.23 |
| single | $10^3$ | **0.998** | **0.01** | **0.93** | **0.06** | **0.83** | **0.15** |
| multi | | 0.997 | **0.01** | 0.92 | 0.07 | 0.77 | 0.20 |
| single | $10^4$ | 0.996 | **0.01** | 0.89 | 0.10 | 0.57 | 0.54 |
| multi | | 0.996 | **0.01** | 0.88 | 0.10 | 0.56 | 0.57 |

between consecutive poses. For people at small scale and slow motion velocity this is often the case.

**Reduction of Frame Rate** Finally, in contrast to the scaling approach, we also evaluated the impact of reduced frame rates on the performance of the keypoint tracker. Although the spatial distance between two consecutive poses

**Table 4.3**: Prediction based on last difference. This table shows the results following a naive approach that takes the last observed difference between timestep $t-1$ and $t$ as a prediction for the next pose at timestep $t+1$. For the reduced frame rate scenarios, the naive approach is inferior to both single- and multi-keypoint approach if $\alpha$ is chosen between $10^1$ and $10^3$.

| **Framerate [fps]** | | **30** | | **6** | | **3** | |
|---|---|---|---|---|---|---|---|
| | $\alpha$ | **OKS** | **$\alpha$MSE** | **OKS** | **$\alpha$MSE** | **OKS** | **$\alpha$MSE** |
| naive | $10^0$ | 0.984 | 0.01 | 0.85 | 0.14 | 0.73 | 0.38 |
| naive | $10^1$ | 0.983 | 0.02 | 0.86 | 0.14 | 0.74 | 0.38 |
| naive | $10^2$ | 0.988 | 0.01 | 0.86 | 0.14 | 0.73 | 0.38 |

increases with reduced frame rate, the results do not improve. This is most probably mainly due to the loss of information. While scaling increases the overall size of poses and distance between these, it keeps the information of the movement itself. This makes it easier to anticipate the pose for the next timestep. By reducing the frame rate more complex movements between consecutive time steps are conceivable, which makes it more difficult to produce the correct prediction.

**Naive Prediction Comparison** Since the comparison of the learning-based approaches for single- and multi-keypoint setup showed that the former achieves as good results as the latter, we were also interested whether an even simpler approach could be sufficient for a crowd setup. Therefore, a naive approach was examined that takes its prediction for the next timestep solely based on the difference between the last two time steps. Table 4.3 reports the obtained results for this naive approach and for values of $\alpha$ up to 100. The first thing that stands out are the consistent results over all scales $\alpha$. This is logical, since we just take the existing pose difference between the last two timesteps and add it on the current pose to obtain a prediction. Smaller deviations are due to the evaluation process which takes random snippets from each sequence in the test set to evaluate the performance. Since the approach is of linear nature, it behaves independent from the scale identical. Concerning the reduction of the frame rate, the same observations could be made as in the single- and multi-keypoint

experiments. This naive approach performs as both examined NPP-approaches at $\alpha = 1$, but starts to get outperformed for bigger values of $\alpha$.

# 5 Conclusion

This work examines the suitability of a state-of-the-art human pose tracking method proposed by Okada et al. [6] for application within video surveillance scenarios. We re-implemented the initial algorithm proposed in the paper and applied it to synthetically generated crowd data for training and evaluation purposes. In addition to that, we extended and adapted the method by investigating its performance on whole-body predictions and expanded the inter-timestep distances in two ways. With the single-keypoint approach performing almost all the time best, the assumption came up that simpler predictions perform better in settings with persons at small scale. We therefore finally compared the single-keypoint approach to a naive prediction approach solely based on the last measured offset.
In future work, we will examine the impact further ways to create such keypoint bundles, e.g. on tracking-by-detection using bounding boxes or a combined way to smooth out the resulting tracklets. We furthermore will go from synthetic data to real world data, which to this point has only been evaluated on a qualitative way which was not part of this report.

# References

[1] COCO Consortium. *COCO - Keypoint Evaluation*. Accessed: 2020-11-01. URL: `http://cocodataset.org/#keypoints-eval`.

[2] Matteo Fabbri et al. “Learning to Detect and Track Visible and Occluded Body Joints in a Virtual World”. In: *European Conference on Computer Vision (ECCV)*. 2018.

[3] Thomas Golda. “Image-based Anomaly Detection within Crowds”. In: *Proceedings of the 2018 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory. Ed.: J. Beyerer, M. Taphanel*. Vol. 40. Karlsruher Schriften zur Anthropomatik / Lehrstuhl für Interaktive Echtzeitsysteme, Karlsruher Institut für Technologie ; Fraunhofer-Inst. für Optronik, Systemtechnik und Bildauswertung IOSB Karlsruhe. KIT Scientific Publishing, Karlsruhe, 2019, pp. 11–24. ISBN: 978-3-7315-0936-3.

[4] Thomas Golda et al. “Image domain adaption of simulated data for human pose estimation”. In: *Artificial Intelligence and Machine Learning in Defense Applications II*. Ed. by Judith Dijk. Vol. 11543. International Society for Optics and Photonics. SPIE, 2020, pp. 112–127. DOI: 10.1117/12.2573888. URL: https://doi.org/10.1117/12.2573888.

[5] J. Li et al. “CrowdPose: Efficient Crowded Scenes Pose Estimation and a New Benchmark”. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 10855–10864. DOI: 10.1109/CVPR.2019.01112.

[6] Masashi Okada, Shinji Takenaka, and Tadahiro Taniguchi. *Multi-person Pose Tracking using Sequential Monte Carlo with Probabilistic Neural Pose Predictor*. 2020. arXiv: 1909.07031 [cs.CV].