Study on Improvement of Estimation Accuracy in Pose Estimation Model Using Time Series Correlation

Atsuya Yamakawa
Graduate School of Fundamental Science and Engineering Department of Communications and Computer Engineering
Waseda University
Tokyo, Japan
0505ya.soc.jack@suou.waseda.jp

Takaaki Ishikawa
Global Information and Telecommunication Institute
Waseda University
Tokyo, Japan
takaxp@ieee.org

Hiroshi Watanabe
Graduate School of Fundamental Science and Engineering Department of Communications and Computer Engineering
Waseda University
Tokyo, Japan
hiroshi.watanabe@waseda.jp

***Abstract*—Detecting human pose in a video is a difficult task. Although many high-performed human pose estimation models have been proposed in the last few years, the estimation accuracy has always been a major concern. In this study we present a method to improve the accuracy of human pose estimation for videos. Technically, predicted human pose is a set of time series data. Thus, by using time series correlation, human pose estimation can be performed in a better accuracy. We combine a CNN based human pose estimation model with a multiple object tracking framework to achieve this. Undetected/mis-detected body joints will be interpolated using the information from previous and following frames. As a result, our proposed method improved the accuracy of an existing CNN based human pose estimation model by reducing the number of undetected and mis-detected frames by 6.30% and 0.98% respectively.**

Keywords—Computer Vision, Human Pose Estimation, Multiple Object Tracking

I. Introduction

Human pose estimation has been a hot topic in the field of computer vision. It has a wide range of applications such as action recognitions, video games, animations, etc. Recently, many machine learning based human pose estimation models have outperformed traditional approaches in the accuracy of localizing human keypoints in images and videos. The approach using Convolutional Neural Network (CNN) enables flexible pose estimation, without special equipment.

However, when we apply these models to a video taken by a general camera, keypoints may not be detected, or detection errors may occur depending on the photographic conditions and methods. Such incorrect keypoint detection has an adverse effect on motion detection and motion analysis.

The purpose of this study is to improve the general pose estimation accuracy of existing CNN based pose estimation models. In this paper, we propose a method of utilizing time series correlation between same body joints among different frames. Also, we investigate the impact of our proposed method in terms of pose estimation accuracy, when CNN based pose estimation models are applied to videos.

II. Related Research

A. OpenPose

OpenPose [1] is a CNN based pose estimation model proposed by Zhe et al. in 2017. OpenPose can detect and localize human's keypoints, such as eyes and body joints. Keypoint locations are obtained as x and y coordinate values of the image. One of the advantages of OpenPose is the ability to estimate human pose easily, and in real-time. As it can be applied to images taken by general cameras, it can be applied to various fields [2]. In this paper, we use OpenPose as the human pose estimator.

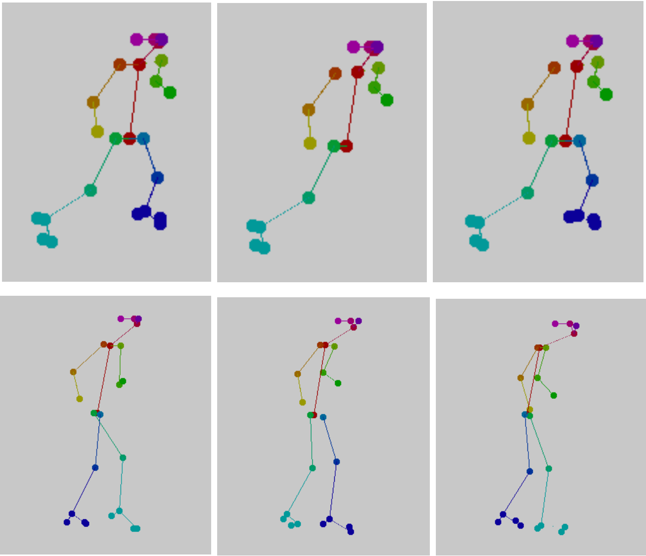

Fig. 1. **Top:** Example of 3 consective frames where keypoints are undetected. **Bottom:** Example of 3 consective frames where keypoints are mis-detected.

OpenPose consists of two networks. The image is analyzed by a CNN to generate feature maps. This CNN is initialized by the first 10 layers of VGG-19 [3] and fine-tuned. After extracting the feature maps, that will then be passed to the respective network. One is a network to generate confidence maps for part detection. The other is to generate part affinity fileds for part association. Combining these two outputs, the pose estimation for each person will be generated.

OpenPose has the capability to process videos. However, each frame of the video is considered as a single image and processed independently. As most CNN based pose estimations are, OpenPose is designed to estimate human poses in images. Since OpenPose estimates poses from two-dimensional images, keypoints may be undetected or mis-detected depending on the spatial and temporal resolution of the input sequence. For example, if one's motion is relatively too fast to the camera's shutter speed, incorrect detections are likely to occur. Fig. 1 shows an example of keypoint undetection and mis-detection.

Also, OpenPose has a random order of which person's pose to be estimated. Due to this characteristic, it is hard to

978-1-7281-9802-6/20/$31.00 ©2020 IEEE

determine same person's keypoint coordinates among different frames in the video of multiple people.

B. DeepSORT

DeepSORT [4] is an object tracking framework that incorporates deep learning metric into Simple Online Realtime Tracking (SORT) [5]. SORT is an object tracking framework based on Kalman filter. When processing each frame of the video, the following eight parameters are added to the detected object to indicate its state.

$$(u, v, a, h, u', v', a', h') \tag{1}$$

(u, v) is the central coordinate value of the bounding box, a is the aspect ratio of the bounding box, and h is the height of the image. u', v', a', and h' are the speeds of the respective parameters. Kalman filter uses these parameters as a "track" in subsequent frames to enable robust object detection. Thus, SORT uses the absolute coordinates and velocity coefficients of the object to achieve object tracking that assumes a linear velocity model. However, SORT returns a relatively high number of identity switches since the association metric is accurate only when state estimation uncertainty is low.

Therefore, SORT fails to track objects in real world scenarios where occlusions appear occasionally. DeepSORT introduced a new feature vector called "appearance" as a distance function to overcome this issue for multi-viewpoint videos and made it robust to object occlusions.

III. Proposed Method

A. Keypoint undetected/mis-detected frames

Whenever keypoints are undetected in OpenPose, both x and y coordinate values of those keypoints will be 0. In this paper, for each person j's keypoint k_j^i in frame i, while both k_j^{i-1} and k_j^{i+1} are detected, but k_j^i is undetected, we define frame i as "keypoint undetected frame" i'.

$$\begin{gathered} i' \leftarrow i \\ where \;\; k_j^i = (0,0) \;\; and \;\; k_j^{i-1} \neq (0,0) \;\; and \;\; k_j^{i+1} \neq (0,0) \end{gathered} \tag{2}$$

Likewise, for each person j's keypoint k_j^i in frame i, while both k_j^{i-1} and k_j^{i+1} are detected, but k_j^i is mis-detected, we define the frame i as "keypoint mis-detected frame" i''. In fact, this indicates frames which contain keypoint outliers. We focus on the difference δ_k^i which is given as keypoint k_j's spatial distance among 2 consecutive frames $i-1$ and i.

$$\begin{gathered} i'' \leftarrow i \\ where \;\; \delta_{j,k}^i > \theta \cdot \delta_{j,k}^{i-1} \\ and \;\; k_j^{i-1} \neq (0,0) \;\; and \;\; k_j^{i+1} \neq (0,0) \end{gathered} \tag{3}$$

Value θ is the threshold which was determined in a prelaminary experiment. In this way, we define keypoint mis-detected frames based on the relative amount of changes for each keypoint.

Both keypoint undetected/mis-detected frames will be the target for being interpolated using the keypoint coordinate information of the previous and following frames.

B. Person ID assignment and interpolation

In order to interpolate coordinate values, it is necessary to have the ability to extract coordinate values of a selected person across different frames. In this study, we combine OpenPose Python API and DeepSORT to assign identifiers to each person in the image, and also output them as "person ID" along with the keypoint coordinate values. This makes it possible to selectively retrieve the keypoint coordinate values of the same person across multiple frames, by specifying the person's ID. Fig. 2 shows an example of the output image when combining OpenPose Python API and DeepSORT.

For keypoint undetected/mis-detected frames, we apply linear interpolation to interpolate keypoints. This is based on the fact that human motions do not change drastically in a small amount of time. It is likely that the undetected or mis-detected keypoint k_j^i exists near the midpoint of keypoint k_j^{i-1} and keypoint k_j^{i+1}.

For undetected frame i', let the keypoint of person $j_{i'}$ in frame $i'-1$ and $i'+1$ be $k_j^{i'-1}$ and $k_j^{i'+1}$ respectively. We apply linear interpolation to keypoint $k_j^{i'}$ which both x-coordinate and y-coordinate value of person $j_{i'}$ is 0.

$$k_j^{i'} \leftarrow \frac{k_j^{i'-1} + k_j^{i'+1}}{2} \tag{4}$$

For mis-detected frame i'', let the keypoint of person $j_{i''}$ in frame $i''-1$ and $i''+1$ be $k_j^{i''-1}$ and $k_j^{i''+1}$ respectively. We apply linear interpolation to keypoint $k_j^{i''}$ which difference $\delta_{j,k}^{i''}$ is greater than the threshold $\theta \cdot \delta_{j,k}^{i''-1}$.

$$k_j^{i''} \leftarrow \frac{k_j^{i''-1} + k_j^{i''+1}}{2} \tag{5}$$

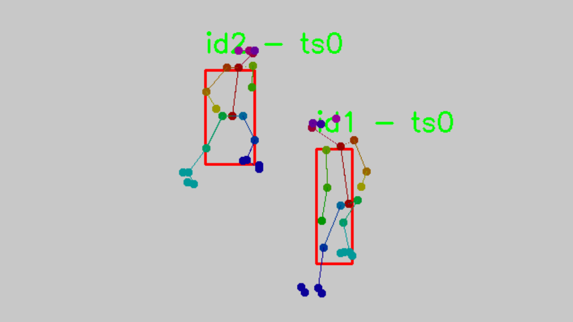

Fig. 2. Output example of pose estimation result with person ID

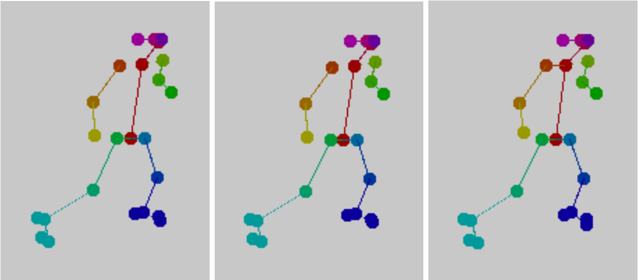

Fig. 3. Output example of interpolated result in "sample2.mp4"

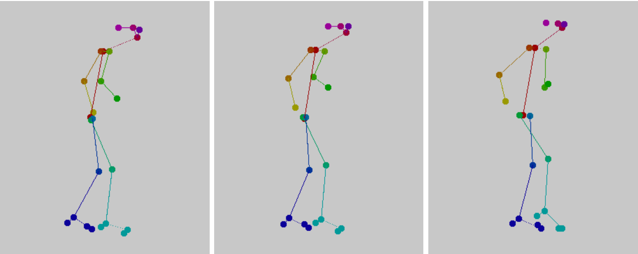

Fig. 4. Output example of interpolated result in "sample5.mp4"

TABLE I THE PERCENTAGE OF INTERPOLATED FRAMES

Video	Number of interpolated "undetected frames"	Percentage of interpolated "undetected frames" [%]	Number of interpolated "mis-detected frames"	Percentage of interpolated "mis-detected frames" [%]	Total number of frames
sample1.mp4	437	8.88	46	0.934	4923
sample2.mp4	329	12.8	47	1.82	2579
sample3.mp4	29	0.624	33	0.71	4650
sample4.mp4	145	4.55	55	1.73	3187
sample5.mp4	374	2.84	87	0.66	13188
sample6.mp4	18	8.14	0	0	221

TABLE II THE NUMBER OF UN-DETECTED FRAMES ($\theta = 2, \theta = 3$)

Video	Number of frames	Number of people	Number of frames detected as mis-detected frames		Number of frames which are visually judged to be mis-detected frames	
			θ=2	θ=3	θ=2	θ=3
sample7.mp4	4650	1～3	63	34	39	32
sample8.mp4	3187	2	80	58	66	55
sample9.mp4	13188	1～3	115	105	111	105

TABLE III THE PROPERTIES OF THE INPUT VIDEOS

Video	Resolution	FPS	Length	Number of people
sample1.mp4	1980×1080	30	0:02:44	5～22
sample2.mp4	1980×1080	30	0:01:26	1～3
sample3.mp4	1280×720	30	0:02:35	1
sample4.mp4	1280×720	24	0:02:13	2
sample5.mp4	1280×720	24	0:09:10	1～3
sample6.mp4	640×360	24	0:00:09	2

TABLE IV THE NUMBER OF PROPERLY INTERPOLATED FRAMES

Video	Interpolated undetected frames	Correctly interpolated frames	Interpolated mis-detected frames	Correctly interpolated frames
sample1.mp4	3247	437	221	46
sample2.mp4	365	329	66	47
sample3.mp4	30	29	34	33
sample4.mp4	184	145	58	55
sample5.mp4	496	374	105	87
sample6.mp4	21	18	0	0

IV. Experimental Results

A. Experiment overview

In order to evaluate our proposed method, we used a total of six videos. This set of videos include various situations, such as sports videos, videos of multiple people taken by a fixed-point camera, etc. These videos contain at least one of each keypoint undetected/mis-detected frame. For example, in "sample1.mp4", there are many people in the video, and each person is small. In other words, the spatial resolution of the person in the video was low and many undetected frames were found.

On the other hand, videos such as "sample3.mp4" and "sample5.mp4" are videos that only few people are pictured as the subject. In terms of frame rates, we used videos that are both 30fps and 24fps. These are the typical frame rates for general cameras. The properties of the input videos are shown in Table III.

We processed the video using OpenPose and DeepSORT, and we applied our proposed method. We compared the pre-interpolated and post-interpolated frames for all interpolated frames. Table I. shows the number of frames that were visually judged to be interpolated to the area near the ground-truth location.

B. Preliminary experiment

As a preliminary experiment, we evaluate the detection accuracy of mis-detected frames by changing the value of θ in (3). The difference δ^i is given as the absolute number of pixels. We do not want to set a threshold for δ^i, since frame rates and resolutions may differ depending on the input video. Therefore, we define a threshold θ in order to focus on the ratio between difference $\delta^i$and δ^{i-1}.

The threshold θ in (3), and the number of frames that could be visually determined to be mis-detected frames are shown in Table II. For both $\theta = 2$ and $\theta = 3$, some of the frames which were detected as mis-detected frames were not mis-detected frames.

However, the percentage of that was lower, when threshold was set to $\theta = 3$. Therefore, in this study we use $\theta = 3$ as the threshold so that only frames that are obviously mis-detected frames are the ones we interpolate.

C. Keypoint undetected/mis-detected frames interpolation

In "sample1.mp4", due to the low spatial resolution of each person in the video, most of the frames were defined as undetected/mis-detected frames. Most of the time in the video, each person appeared in a very small area. Hence, it was difficult to judge whether they were interpolated near the ground-truth position or not. Fig. 5 shows an example of 3 consecutive frames of the input video "sample1.mp4". Fig. 6 shows the output example of the interpolated result.

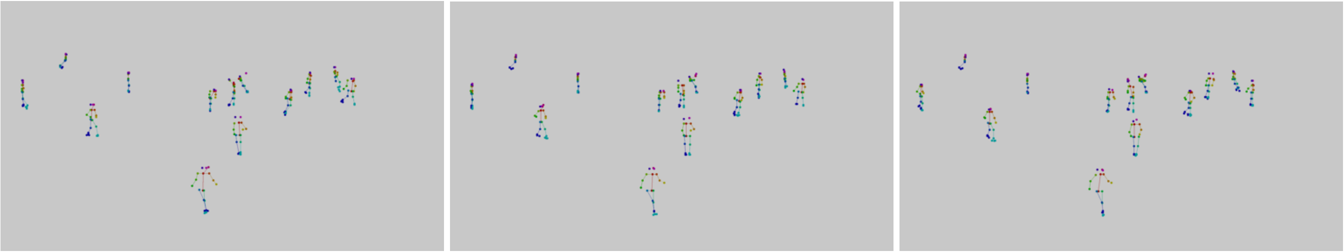

Fig. 5. Example input frame in “sample1.mp4”

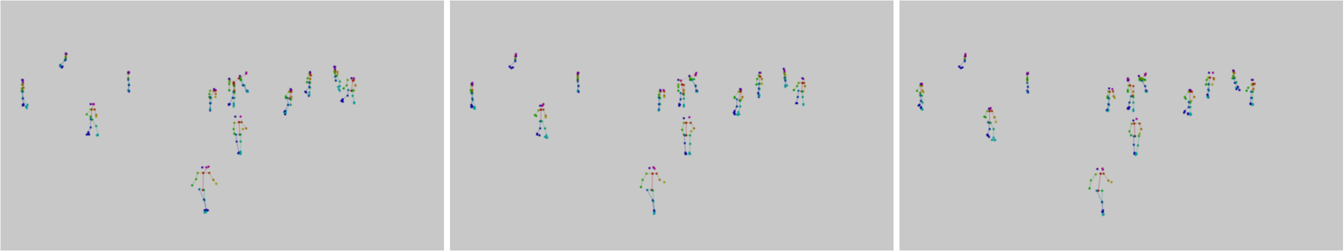

Fig. 6. Output example of interpolated result in “sample1.mp4”

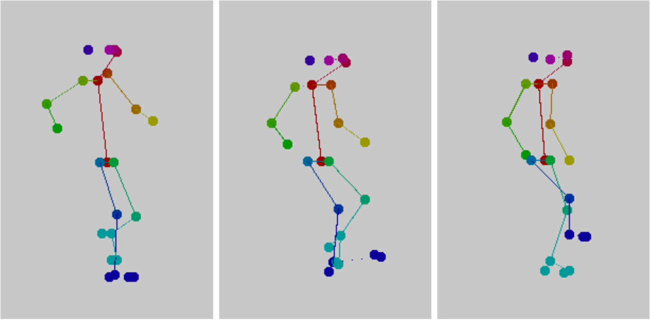

Fig. 7. Example of 3 consecutive input frames in “sample4.mp4”

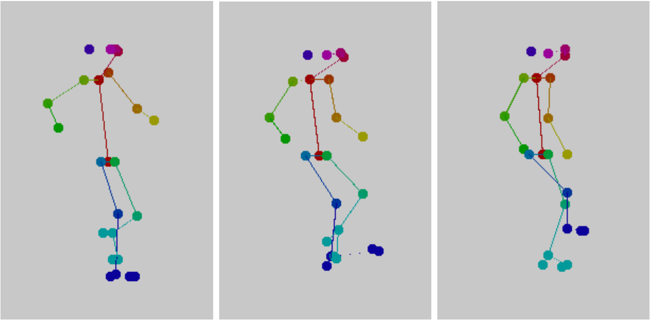

Fig. 8. Output example of interpolated result in “sample4.mp4”

In other videos, most of the interpolated keypoints were corrected to the area near the ground-truth position. In “sample3.mp4”, almost all of the undetected/mis-detected keypoints were correctly interpolated to the ground-truth position. The subject of the video was a single person, standing and facing forward for most of the time. Fig. 7 shows an example of 3 consecutive input frames in “sample4.mp4”. Fig. 8 shows the example output of incorrect interpolation of undetected/mis-detected frames. Table IV shows the number of interpolated undetected/mis-detected frames and the the number of frames that were correctly interpolated.

V. Conclusion

In this paper, we proposed a method to improve the accuracy of human pose estimation using OpenPose. By combining OpenPose's python API and DeepSORT, we notated identifiers to each person along with the x, y coordinate of the body joints. This made it possible to refer to the same person’s keypoint among multiple frames.

From the experimental results, undetected and mis-detected keypoints were interpolated by the proposed method. We confirmed that undetected keypoints were interpolated to the area near the ground truth at a high rate. As a result, the average number of undetected frames and mis-detected frames were reduced by 6.30% and 0.98%, respectively. Therefore, the pose estimation accuracy was improved among all six videos used for the experiment.

While some mis-detected keypoints were corrected to the plausible location, some were not. We will improve how we define mis-detected keypoints, and how we interpolate error keypoints.

References

[1] Z.Cao, T.Simon, S.Wei, and Y.Sheikh: “Realtime Multi-Person 2D Pose Estimation using Part Affinity Field,” In *Computer Vision and Pattern Recognition (CVPR 2017)*, pp.7291-7299, Apr. 2017.

[2] T. Golda, T. Kalb, A. Schemann, and J. Beyerer: “Human Pose Estimation for Real-World Crowded Scenarios,” In *International Conference on Advanced Video and Signal Based Surveillance (AVSS 2019)*, Jul. 2019.

[3] K. Simonyan and A. Zisserman: “Very deep convolutional networks for large-scale image recognition,” In *International Conference on Learning Representations (ICLR 2015)*, Sep. 2015.

[4] N. Wojke, A. Bewley, and D. Paulus: “Simple Online and Realtime Tracking with a Deep Association Metric,” In *International Conference on Image Processing (ICIP 2017)*, pp.3645-3649, Jul. 2017.

[5] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft: “Simple Online and Realtime Tracking,” In *International Conference on Image Processing (ICIP 2016)*, pp.3464-3468, Feb. 2016.