EasyChair Preprint

№ 2580

Human Motion Prediction by 2D Human Pose Estimation using OpenPose

Andi Prademon Yunus, Nobu C. Shirai, Kento Morita and Tetsushi Wakabayashi

February 5, 2020

Human Motion Prediction by 2D Human Pose Estimation using OpenPose

Andi Prademon Yunus[1], Nobu C. Shirai[1], Kento Morita[1], and Tetsushi Wakabayashi[1]

Mie University, 1577 Kurima-machiya, Tsu, 514 Japan
[andi, morita, waka]@hi.info.mie-u.ac.jp
shirai@cc.mie-u.ac.jp

Abstract. The prediction of human motion became an important issue, considering that it can be utilized to solve plenty of problems forautonomous systems. In the case of human-machine interaction such asan autonomous car or robot that works in a human living environmentneed to predict the human's future movement for its moving trajectories.Some of the previous researches used Kinect camera which has a depthsensor that the camera used to detect the pose of a human body. However, in this researchwe start with using the RGB camera as the otheroption that we can rely on. We set a goal to predict 1 second ahead of themotion which includes simple motions such as hand gesture and walkingmovement. We used OpenPose library from OpenCV to extract featuresof a human body pose including 14 points. YOLOv3 is used to crop themain feature in the frames before OpenPose process the frame. We inputdistance and direction which are calculated from the features by comparing two consecutive frames into the Recurrent Neural Network Long Short-Term Memory (RNN-LSTM) model. As the result, the human movementwas predicted with 98% of accuracy. The evaluation criteria for acceptable distance was within 1.8% of diagonal frame length. We confirmedthe validity of the RGB based method in the simple human motion casefrom the result, and we conclude that this is an important step to realizethe prediction of more complex human motion.

Keywords: Human motion prediction · RNN-LSTM · OpenPose · YOLOv3 · Deep learning

1 Introduction

People constantly interact with everything around them, such as human to human interaction as well as human to machine interaction. We have been dreaming about robots that could coexist with humans, and now many people are competing to create the most reliable autonomous machines such as auto-driving car and auto-moving industrial robot. In the future, robots will work alongside humans in many applications including logistics, health-care, agriculture, disaster response, and others [1]. However, creating such a reliable autonomous machine is not an easy accomplishment, there are plenty of problems to solve, such as

preventing an auto-driving car collide with another car, or even preventing it to run over a human being. If the auto-driving car cannot predict the human movement, the problem persists. With this in mind, we develop a system that can predict the human motion to try to solve the problem. On the other hand, human motion as the object of this research is difficult to predict due to the countless motion in human behaviors, as well as the differences of the individual behaviors. For these reasons, we decided to cover the scopes of the human motion for the simple first step to memorize the human motion. We created the videos as the dataset for this research that contain simple human motion such as hand gestures and walking movement. As well as the CMU dataset is used in this research since it has similarity of the actual movement that we need.

Some researches develop their systems with data from the RGB-D camera since it has depth parameter for human pose estimation [2, 3]. RGB-D camera such as Kinect camera can estimate precisely of human body parts. However, in this research we start with using the RGB camera as the other option that we can rely on. Our main task of interest is human motion prediction with focus on data obtained from human pose estimation by OpenPose. Even though, the obtained data from OpenPose does not always give pose precisely of human body parts as shown in Fig. 1. These data give us another problem to solve. In this paper, we proposed a method to examine the unstable data can be used for human motion prediction.

A research has been performed for human motion prediction with RGB camera [4]. They focused on human motion forecasting of sports activity especially for safe martial arts such as boxing, karate or taekwondo. As a result, they obtained 0.5 second of human motion prediction by forecasting 15 frame step in a 30fps video. Nonetheless, this paper does not show the accuracy of the prediction.

In recent years, RNN has been used in many cases for prediction including human motion prediction with high accuracy [2, 9, 10]. Since the human behaviors are individually unique and varied, we need a long term prediction algorithm. RNN-LSTM provides the long term prediction and short term prediction based on its memory to save the values of behavior with high accuracy prediction [6, 8].

2 Proposed Method

Fig. 1 shows a block diagram of the proposed method for human motion prediction system in this research. The system receives input from 2 videos as training and testing samples, these videos will later be processed to provide the feature of human body pose as a coordinate data divided by human body parts. We set goal to predict 1 second ahead of the motion, and we prepared 30 fps videos which include simple motions such as hand gesture and walking movement. Nodes are defined by human body parts which cover head, neck, shoulders, elbows, wrists, hip, knees, and ankles.

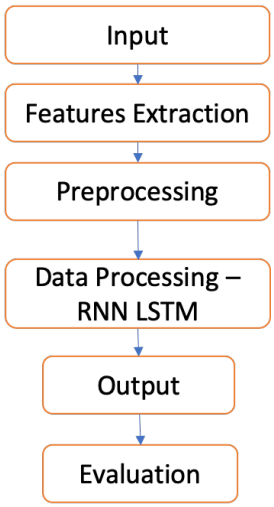


Fig. 1: Block diagram of the proposed method for human motion prediction.

Before we proceed to obtain the prediction, we need to convert the coordinate data into the movement data which contains distance and direction. These movement data obtained from the change of the coordinate from the frame F_i to the frame F_{i-30} with Euclidean formula. After the movement data has been obtained, we proceed to the processing method that includes RNN-LSTM for the prediction.

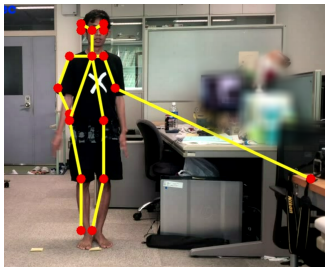


Fig. 2: Frame with human body part miscalculation.

Feature Extraction OpenCV provides the pose estimation for human body that we can use for this research. Feature extraction is one of the main part of this experiment. This experiment uses the dataset from the COCO dataset for the human body pose which contains 18 points consists of human body (e.g., nose, neck and shoulders) [7, 13]. This process obtains the coordinate of each body

point [x,y]. However, the pose estimation that has given by the OpenPose is not always as reliable as expected. Fig. 2 shows the mistake to estimate the left wrist position. The problem of this estimation mistake should be solved since it will become interference for the sequence data prediction. In case of that, we will not get the prediction correctly. Therefore, we need to limit the window frame of the human feature before processing it on OpenPose. We use YOLOv3 to restrict the window size of the frame. YOLO is a single neural network that predicts bounding boxes and predicted class probabilities directly from full image, and in one evaluation [12]. Since this whole detection pipeline consists of a single network, it can be optimized end-to-end directly on detection performance [12]. As shown in Fig. 3, the frame of human feature will be the only input through OpenPose, then the result from OpenPose will be set back to the original frame image. Now, we have obtained the coordinate data of human body pose without any long estimation error.

Fig. 3: YOLOv3 frame cropping result

Data Preprocessing The obtained coordinate data are not suitable to the proposed learning process. Because it contains x, y coordinate in the floating point data type. The value of this x, y coordinate is varied and includes the detailed number. These data will be a big burden for the learning process to memorize all detail and variety. This section converts the obtained coordinates data to the movement data distance d and direction θ which are calculated by the following equations:

$$d_i = \sqrt{(x_i - x_{i-fs})^2 + (y_i - y_{i-fs})^2}, \tag{1}$$

$$\theta_i = \arcsin(y_i - y_{i-fs}/d_i), \tag{2}$$

where i is the number of the frame, x_i is coordinate x in the frame i, y_i is the coordinate y and fs is the frame step which has a constant value of 30 since the video has 30fps property, the frame step would be 30 frames for human motion

prediction of 1 second ahead. In the frame i, b is the opposite side of the angle, and r is the hypotenuse of the angle.

Data Processing and Prediction Once the movement data are ready, we can proceed to obtain the prediction data. We use the movement data instead of the coordinate data because the coordinate data have a big number and the coordinate values [x,y] have large variety, which causes the RNN-LSTM has difficulties on saving the memory. Because the x and y coordinate take wide range of value. In order to restrict the range of value, the proposed method calculated the movement data by Eq. 1 and Eq. 2. Movement data is suitable for RNN-LSTM.

RNN is a class of neural network where connections between the computational units form a directed graph along a temporal sequence. Unlike feedforward networks, RNN can use their internal memory to process arbitrary sequence of inputs. Each of the computing unit in an RNN has a time varying real valued activation and modifiable weight. RNNs are created by applying the same set of weights recursively over a graph-like structure [11]. The learned model in RNN has the same input size, since it has terms of the transition from one state to the other state. LSTM is an extended version of RNN which has the extended memory to memorize not only the short term of the sequence data, but further long term of the sequence data. LSTM networks were discovered by Hochreiter and Schmidhuber in 1997 [8]. LSTM works even given long delays between significant events and can handle signals that mix low and high frequency components.

In this research, the input will be 14 nodes of human body parts and we use 3 stacked of hidden layer for the learning model in RNN-LSTM. The last output will be 14 nodes as well as the input. Some other related researches used 3 stacked layer RNN-LSTM as well [2–4]. We used the Mean Squared Error (MSE) that is defined by

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x})^2, \tag{3}$$

to estimate the loss value in the training process, where n represents n data points on all variables, x_i is the vector of observed values of the variable being predicted, and $\hat{x}$ represents the predicted value .

Evaluation Method The prediction data are approached from the prediction process. But, we are still not sure about the accuracy of these predictions. We set the ground truth prediction for the i-th frame F_i to the coordinate data in $i+30$-th frame F_{i+30}, thus, we can compare the distance between 2 nodes from different frames. The distance is calculated by the following equation which was proposed in the related research [2]:

$$E = \sqrt{(x_{i+30} - x_p)^2 + (y_{i+30} - y_p)^2}, \tag{4}$$

where i is the number of the frame, x_i is coordinate x in the frame i, y_i is the coordinate y in the frame i. By using the predicted movement data at i-th frame d_i and θ_i, the coordinate value in $(i+30)$-th frame (x_p, y_p) is calculated by

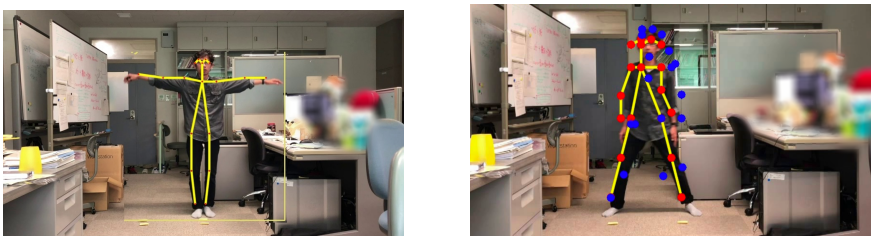
$$x_p = x_i + d_i, \tag{5}$$

$$y_p = y_i + d_i, \tag{6}$$

where x_p is the x coordinate of the prediction, x_i is the x coordinate of frame i, d_i is the value of distance movement of prediction result, y_p is the y coordinate of the prediction, y_i is the y coordinate of frame i.

Table 1: Evaluation distance by percentage of value limited to 1.8% of the the diagonal frame pixels away.

Nodes	Our Dataset	CMU Dataset
Head	84	57
Neck	88	46
Right Shoulder	93	15
Right Elbow	39	40
Right Wrist	29	36
Left Shoulder	88	31
Left Elbow	50	50
Left Wrist	30	40
Right Hip	88	58
Right Knee	95	63
Right Ankle	94	59
Left Hip	90	72
Left Knee	95	82
Left Ankle	93	83



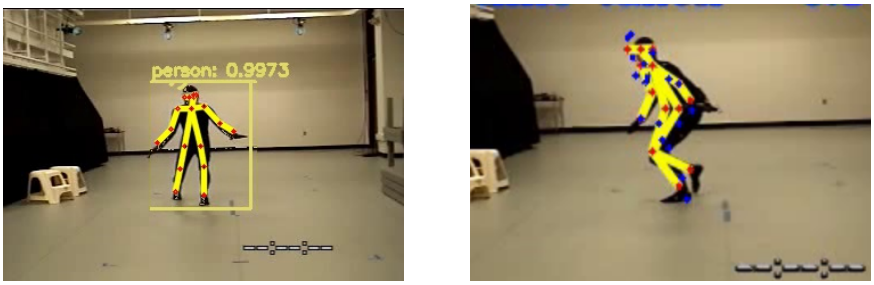
(a) Feature extraction by YOLOv3 and OpenPose.

(b) RNN-LSTM Prediction

Fig. 4: Result frame from feature extraction process and data processing on our dataset by RNN-LSTM.

3 Experiment Results

The result of the extracted feature in Fig. 4a and Fig. 5a give us the 18 nodes of human body pose estimation. Even though, because of the eyes and ears as the human body parts practically will not move from the head we decided to conclude the node of ears, eyes, and nose as a head node that will be centered to nose node. YOLOv3 works well to prevent distant pose miscalculation. Prediction result on our dataset is shown in Fig. 4b, where the red points are the current position, and the blue points are the prediction position. As well as the prediction result on CMU dataset in Fig. 5b. Table 1 shows the evaluation distances for each node defines the frequency of the value below 1.8% of of the the diagonal frame pixels away from the ground truth in percentage. Generally, the prediction results on our dataset shows the better results than the prediction results on CMU dataset where the right knee and left knee have been obtained 95% of the prediction results are reliable. Altough, elbow and wrist nodes on our dataset show that RNN-LSTM has difficulty on predicting the sequence data since elbow and wrist on our dataset move more than other nodes in the video. Whereas the prediction results on CMU dataset mostly below 60% of the prediction result that are reliable. Even though, the prediction results on left hip, left knee, and left ankle show more reliable results than the rest of the nodes which show 72% for the left hip, 82% for the left knee, and 83% for the left ankle.



(a) Feature extraction by YOLOv3 and OpenPose.

(b) RNN-LSTM Prediction

Fig. 5: Result frame from feature extraction process and data processing on CMU dataset by RNN-LSTM.

4 Conclusions and Discussion

Based on the result of this experiment, we have proposed the human movement prediction with RNN-LSTM based on RGB camera for 1 second prediction. We used samples of video that cover hand gesture, sideways moving, and walking. The result showed most of the predictions are close to the correct position that is

a prediction for 1 second of human movement on our dataset. We also performed the prediction on CMU dataset which has an actual sample of walking stealthily, even if the prediction results are not as reliable as on our dataset. These results are the beginning of more optimized prediction process in the future works.

We confirmed the validity of the RGB based method in the simple human motion case from the result, and we conclude that this is an important step to realize the prediction of more complex human motion. For the future works, we need to confirm the prediction result with Kalman Filter to predict the motions as a comparison to result from RNN-LSTM and the combination of RNN-LSTM with Kalman Filter. As well as the comparison of the human motion prediction with another research.

References

1. Erich Mielke, Eric Townsend, David Wingate, Marc D. Killpack. Human-robot comanipulation of extended objects: Data-driven models and control from analysis of human-human dyads. arXiv:2001.00991. (2020)
2. M. Julieta, B. J. Michael, R. Javier. On Human Motion Prediction Using Recurrent Neural Networks. arXiv:1705.02445v1. (2017)
3. Tang Yongyi, Ma Lin, Liu Wei, Zheng Wei-Shi. Long-Term Human Motion Prediction by Modeling Motion Context and Enhancing Motion Dynamics. In:935-941. 10.24963/ijcai.2018/130.(2018)
4. Wu, Erwin, Koike, Hideki. Real-time human motion forecasting using a RGB camera. 1-2. 10.1145/3281505.3281598. (2018)
5. C. Yujiao,Z. Weiye, L. Changliu, T. Masayoshi. Human Motion Prediction using Adaptable Neural Networks. arXiv:1810.00781. (2018)
6. R. Akita, A. Yoshihara, T. Matsubara and K. Uehara. Deep learning for stock prediction using numerical and textual information. IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS) pp. 1-6. doi: 10.1109/ICIS.2016.7550882 (2016)
7. Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, Yaser Sheikh. OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. arXiv:1812.08008 [cs.CV]. (2018)
8. Hochreiter, Sepp Schmidhuber, Jrgen. Long Short-term Memory. Neural computation. 9. 1735-80. 10.1162/neco.1997.9.8.1735. (1997)
9. Alex Graves. Generating Sequences With Recurrent Neural Networks. arXiv:1308.0850v5 [cs.NE] (2014)
10. Che Zhengping, Purushotham Sanjay, Cho Kyunghyun, Sontag David, Liu Yan. Recurrent Neural Networks for Multivariate Time Series with Missing Values. Scientific Reports 8:6085 DOI:10.1038/s41598-018-24271-9 (2018)
11. S. Selvin, R. Vinayakumar, E. A. Gopalakrishnan, V. K. Menon and K. P. Soman, "Stock price prediction using LSTM, RNN and CNN-sliding window model," 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Udupi, 2017, pp. 1643-1647. doi: 10.1109/ICACCI.2017.8126078
12. Asif Sattar, Human detection and distance estimation with monocular camera using YOLOv3 neural network, University of Tartu, Faculty of Science and Technology, Institute of Technology, Master Thesis (30 ECTS), (2019)
13. OpenCV "Openpose", https://bit.ly/2G9DphR. Last accessed 12 July 2019