Human Motion Prediction With Graph Neural Networks

Irmak Guzey[1], Ahmet E. Tekden[1], Evren Samur[2] and Emre Ugur[1]

I. Introduction

An articulated human body consists of coupled multiple links whose motions affect each other. The effects generated on body parts by the muscles propagate through the body and influence other parts depending on the kinematic and dynamic relations. In this work, we propose to use graph neural networks, propagation networks [1] in particular, to investigate the problem of modelling full-body motion. Given the real movement of one or multiple parts of the body, our work aims to predict the movement of the rest of the body exploiting the underlying graph structure that encodes the task. The body parts and the relations between them are encoded as the nodes of a graph and edges between these nodes. How the nodes are related to each other is learned, and how the effects of multiple nodes on each node should be accumulated is computed in graph structure. A publicly available whole body motion data set is used to train our network. Preliminary results showed that the system could predict motion of body parts given ground truth motion of a subset of hand, foot and head.

Previously, [3] and [4] developed deep recurrent neural networks to model human motion, and learned time-dependent short-term representations of human body. [5] and [6] developed vision dependent models which are not time-dependant and they used the model to synthesize new trajectories. Our propagation network model is not time-dependent, the predictions are made through time by chaining predictions back to back. We further explicitly represent each link and let the system learn to propagate the effects of chain of links in each single step.

II. Proposed Model

Figure 1 gives a graphical illustration of the framework that is used, in this section this illustration will be clarified.

In this environment we represent articulated human bodies with a graph structure $G =< O, R >$ where nodes are represented by objects (links) $O = \{o_i\}_{i=1...N_o}$ where o_i represents the attributes of the object i (N_o is the number of objects in the scene) and edges are represented by relations (connections between the links) $R = \{r_j\}_{j=1...N_r}$ where r_j represents the attributes of relation j (N_r is the number of relations in the scene).

[1]Irmak Guzey, Ahmet E. Tekden and Emre Ugur are with Department of Computer Engineering, Bogazici University, Istanbul, Turkey. [2]Evren Samur is with Department of Mechanical Engineering, Bogazici University, Istanbul, Turkey.

Object attribute extraction differs, according to whether the corresponding link is a reference point or not. If it is, then only its' actual position is given as the object attribute, more formally $o_{i,t} =< x_{i,t}, \_ >$, on the other hand if a link's position is aimed to be predicted then its' object attribute $o_{i,t} =< \dot{x}_{i,t}, \dot{v}_{i,t} >$ consists of its' predicted position $\dot{x}_{i,t}$ and predicted velocity $\dot{v}_{i,t}$ in time t. Each relation attribute $r_j =< \delta x_j >$ consists of the relative positions difference of the objects that the relation j connects, more formally $\delta x_j = x_i - x_k$ where object k and i has the relation j between them.

In propagation networks the main aim is to calculate the propagating effects of object and relation attributes. This is done by having two subsequent MLPs that are connected with each other, which evaluate these effects and use them at the same time in a loop. The attributes are first encoded by encoders for objects and relations, f_O^{enc} and f_R^{enc}, relatively, in order to be processed more efficiently (effects of the encoders are explained in [1]).

$$c_{k,t}^r = f_R^{enc}(r_{k,t}), k = 1...N_r \tag{1}$$

$$c_{i,t}^o = f_O^{enc}(o_{i,t}), i = 1...N_o \tag{2}$$

where $o_{i,t}$ and $r_{k,t}$ represent object and relation attributes in given time t. Object encoder f_O^{enc} is an MLP with 3 hidden layers of 150 neurons and relation encoder f_R^{enc} is an MLP with 1 hidden layer of100 neurons.

After encoding, in order to predict the future state of the environment, encoded attributes are given as input to the MLPs. f_O^l and f_R^l represent these propagators for objects and relations relatively, and outputs of these propagators are given as input to each other in the next propagation step, as shown in figure 1.

$$e_{k,t}^l = f_R^l(c_{k,t}^r, p_{i,t}^{l-1}, p_{j,t}^{l-1}), k = 1...N_r \tag{3}$$

$$p_{i,t}^l = f_O^l\left(c_{i,t}^o, p_{i,t}^{l-1}, \sum_{k \in R_i} e_{k,t}^{l-1}\right), i = 1...N_o \tag{4}$$

where R_i represent the set of relations that the object i is a part of, and $e_{k,t}^l$ and $p_{i,t}^l$ represent the propagating effects of relation k and object i in propagation step l and given time t. Object propagator f_O^l is implemented as an MLP with 1 hidden layer of 100 neurons and relation propagator f_R^l is an MLP with 2 hidden layers of 150 neurons.

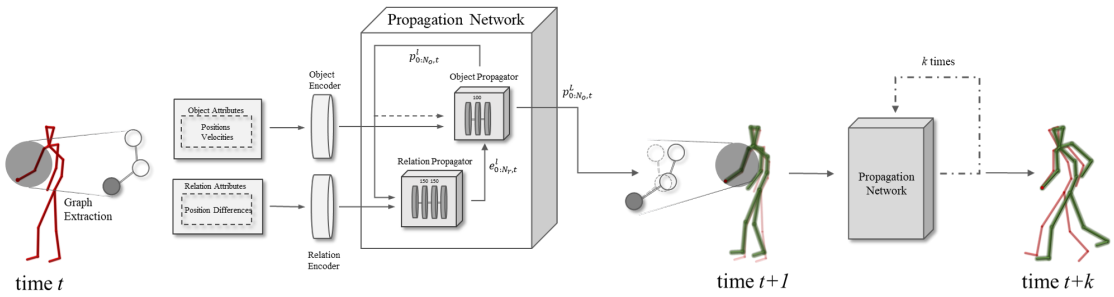

Fig. 1. The graphical illustration of the framework. Right hand is given as the reference point and other links are predicted and chained through k timesteps. One iteration of the attributes through propagation network gives the attributes in subsequent time.

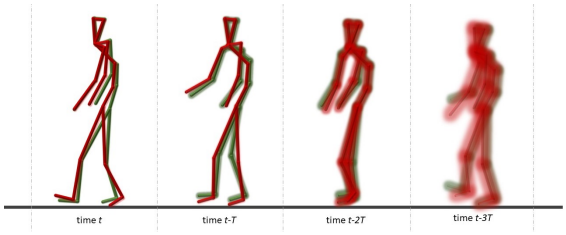

Fig. 2. Example prediction of a walking movement. Green lines represent the predictions, red lines represent the ground truth postions.

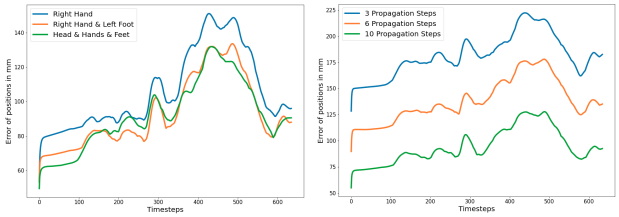

Fig. 3. Average errors of the positions of the joints with different points given and different propagation steps used.

This process goes on until number of set propagation steps L is reached, the number L is decided according to the complexity of the environment. After the number of propagation steps is satisfied the object and relation attributes in time $t+1$ are extracted from the final propagating effects.

$$o_{i,t+1} = p^L_{i,t} \tag{5}$$

Also with a similar approach by chaining the predictions, state of the environment in time $t+k$ is estimated. More about graph neural networks can be found in [2] and similar usage of propagation networks was also introduced in [7].

III. Results & Conclusion

We trained and examined the developed model with the KIT Whole Body Language dataset [8]. The dataset consists of recorded joint positions for multiple timesteps for different movement trajectories. Two trajectories with walking movement are selected and used for training and testing the model.

The accuracy of the model differs according to the number of reference points and the number of propagation steps that are used while training. As the number of reference points increases and their positions cover a larger area, it is expected the model to have better accuracy since propagation can effect more variate positions easier. Also as the number of propagation steps increases, the number of links that are accurately evaluated is likely to increase. Figure 3 shows average difference between the actual and predicted positions of the links with the model trained with differing number of reference points and propagation steps.

In figure 4 the models are trained with the right hand given as the reference point, and the other links' positions are predicted. The figure shows the predicted and actual positions of the links in left leg and right arm. It can be seen that while the average error in right arm is around 10mm the average error in left leg is around 100mm. This shows how the network's information propagates starting from the reference point and how its' affect decreases in the links that are further from the reference.

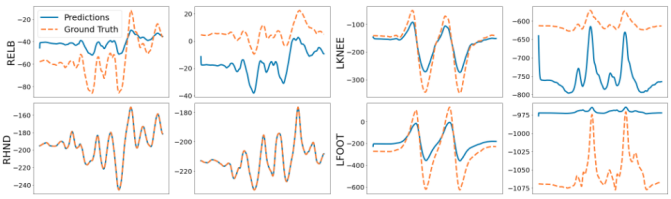

Fig. 4. Positions of the links (horizontal and vertical positions in mm) of right arm and left leg with the model trained with right hand as the reference point. RELB, RHND represent right arm elbow and hand, LKNEE, LFOOT represent left leg knee and foot links relatively.

In this study, we investigated use of graph neural networks in encoding whole body motion, representing links with nodes and relations with the edges in the graph. Our preliminary experiments show that after learning, our system can predict motions of various body parts given motion of a single hand or leg, with different accuracies. We additionally showed that number of propagation becomes important in relating body parts that are far away in the kinematics chain.

ACKNOWLEDGMENT

This work was supported by the Scientific and Technological Research Council of Turkey (TUBITAK, 118E923).

REFERENCES

[1] Y. Li, J. Wu, J. Zhu, J. B. Tenenbaum, A. Torralba, R. Tedrake, "Propagation Networks for Model-Based Control Under Partial Observation" in *ICRA 2019*.

[2] P. Battaglia, J.B.C. Hamrick, V. Bapst, A. Sanches, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, C. Gulcehre, F. Song, A. Ballard, J. Gilmer, G. E. Dahl, A. Vaswani, K. Allen, C. Nash, V. J. Langston, C. Dyer, N. Heess, D. Wiersta, P. Kohli, M. Botvinick, O. Vinyals, Y. Li, and R. Pascanu, "Relational inductive biases, deep learning, and graph networks" in *arXiv preprint arXiv:1806.01261*, 2018.

[3] J. Martinez, M. J. Black and J. Romero, "On Human Motion Prediction Using Recurrent Neural Networks" in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*Honolulu, HI, 2017, pp. 4674-4683.

[4] K. Fragdiadaki, S. Levine, P. Felsen and J. Malik, "Recurrent Network Models for Human Dynamics" in *International Conference on Computer Vision*, 2015.

[5] D. Holden, T. Komura, and J. Saito. 2017. "Phase-functioned neural networks for character control" in *ACM Trans. Graph.* 36, 4, Article 42 (July 2017), 13 pages.

[6] D. Holden, J. Saito, and T. Komura. "A deep learning framework for character motion synthesis and editing" in *ACM Trans. Graph.* 35, 4, Article 138 (July 2016), 11 pages.

[7] A. E. Tekden, A, Erdem, E. Erdem and M. Imre, M. Y. Seker and E. Ugur. "Belief Regulated Dual Propagation Nets for Learning Action Effects on Groups of Articulated Objects" in *ICRA 2020*.

[8] C. Mandery, Ö. Terlemez, M. Do, N. Vahrenkamp and T. Asfour, "The KIT Whole-Body Human Motion Database", International Conference on Advanced Robotics (ICAR), pp. 329 - 336, 2015