Evaluating the Use of Interpolation Methods for Human Body Motion Modelling

Egemen Halici and Erkan Bostanci

Abstract—**This paper proposes the use of interpolation methods rather that conventional learning algorithms such as Support Vector Machines (SVM) or Policy Learning by Weight Exploration with Return (POWER) for modelling human motion. The main aim was using a simpler model with less time and space complexity for later use in the recognition of certain actions. Three different polynomial interpolation methods, namely Lagrange, spline and cubic spline have been investigated. Parts of the dataset were used instead of the complete dataset using grouping techniques to reduce the training time. A non-parametric test known as Mc-Nemar's test was used to identify statistically significant performance differences between these methods. It was found that the cubic spline resulted in better accuracy.**

Index Terms—**Interpolations, dynamic movement primitives, learning algorithms, human motion.**

I. Introduction

Learning and recognition of human motion is attracting much attention by the literature since these tasks are required in many different application areas including industrial robotics automation and motion capture for the film industry. Dynamic Movement Primitives (DMP) have been developed to model human motion. These primitives actually store the positions of different joints such as wrist or head along with the timing information allowing the complete motion to be captured. Such algorithms store the complete motion sequence with the positions in small time intervals. A general problem with such approaches is the inefficient use of memory.

From the fact that the changes in the joint locations are continuous in terms of time, this paper presents the use of polynomial interpolation in order to achieve better memory use and relatively easier functional representation for each joint. A further attempt to increase memory efficiency proposed with this study is the use of subsets of the complete motion sequence in the modelling process, still keeping the accuracy of the motion model [1], [2].

Various types of interpolation methods [3], [4] were examined here. A public motion capture dataset was employed. The dataset includes recordings of different motions captured in 120 frames per second. Using polynomial interpolation, the joint location is expressed as a polynomial of the frame number. The redundancy existing in the dataset was also removed using selected subsets of the complete sequence.

The rest of paper is structured as follows: Section II describes the dataset followed by brief descriptions of existing learning algorithms in Section III. Interpolation methods are described in Section IV and then the approach followed for selecting the subset of the complete dataset is given in Section V. Experimental results are presented in Section VI and finally the paper is concluded in Section VII.

II. Data Set

The datasets were obtained from Carnegie Mellon University Graphics Laboratory Motion Capture Database and include human motion sequences captured with 120 frames per second along with timing information as shown in Fig. 1 and Fig. 2.

Fig. 1. Images from the dataset.

Fig. 2. Images from the dataset.

In dataset, each frame has information about 29 different joint locations on the human body as depicted in Fig. 2. Some of the joints on human body can have more than one positional data. For instance, root joint has 6 different positional data.

Manuscript received November 23, 2017; revised January 24, 2017.

Egemen Halici and Erkan Bostanci are with SAAT Lab., Computer Engineering Department, Ankara University, Golbasi, Ankara, Turkey (e-mail: egemenhalici@gmail.com, ebostanci@ankara.edu.tr).

DOI: 10.7763/IJCTE.2018.V10.1194

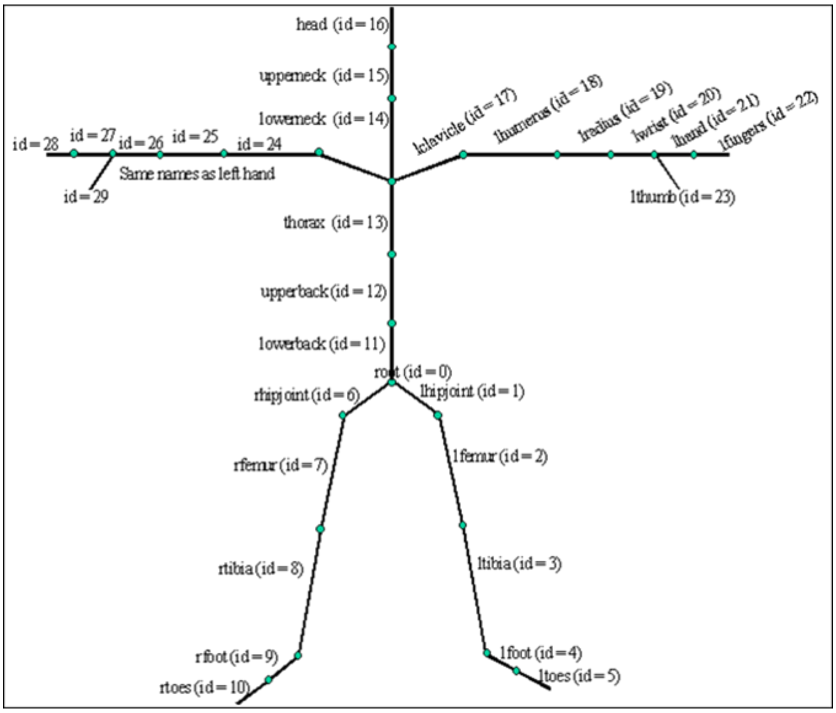


Fig. 3. Joint locations on the skeleton system.

Datasets come from stable sources and the joint tracking zones do not change over time.

Therefore, the arrays which are used for calculating polynomial equations, were prepared with exact joint information.

III. Dynamic Movement Primitives

When the human body in the sequences perform a motion, the joint locations change in due course. Once these motions can be replicated using DMP [5], it can be applied to robotics making it easier and safer to perform dangerous tasks such as carrying heavy or hot parts. The same motion can be performed by robotic limbs in such real world scenarios [6], [7]. The literature presents algorithms for learning such human body movements. Many of these algorithms make use of the complete dataset in order to obtain accurate motion results [8].

A. Support Vector Machine

Support Vector Machine (SVM) is common algorithm for modelling of human body movements. It is really strong algorithm for binary clustering problems [9].

$$f(q) = \sum_{i=1}^{N} yi\alpha ik(q, xi) + b \qquad (1)$$

xi, are vectors which comes from dataset between *i* 2 [1;*N*]. q denotes a classification coefficient. *k*(.,.) is a kernel function. SVM classification can be used for deciding the border between movement transitions. Although, SVM is efficient, easy-to-use method but it requires the complete dataset, making it inefficient in terms of memory [10].

B. Policy Learning by Weighting Exploration with the Return (POWER)

POWER algorithm is a function based learning approach [11]. This algorithm uses 2 variables, namely *y* and *z*. The first one is used for the movement phase while the second one is for the external connections. The process of learning motions such as rotation and translation is an iterative process taking about 20 iterations to yield optimal solutions [11], [12].

IV. Type of Interpolations

This section will brief describe various interpolation methods tested in this paper for training to recognize human motion. Note that all these interpolation methods consider the frame number as the input to the polynomial to be approximated and the corresponding position for that time instance as the result of the polynomial.

A. Lagrange Interpolation

The Lagrange interpolation creates a curve for known x_i for $i \in [1;N]$ values for $f(xi)$

$$P_n(x)=\sum_{i=0}^{n}\left(\prod_{\substack{j=0\\ j\neq i}}^{n}\frac{(x-xj)}{xi-xj}\right)f(x) \qquad (2)$$

B. Spline Interpolation

Originally, spline was a term for elastic rulers that were bent to pass through a number of predefined points ("knots"). There are limited values which are using for creating equations. Spline interpolation focuses on one particular interval (x_k, x_{k+1}). In this way, straight lines can be created between the specified intervals using the following equations.

$$f = Af_k + Bf_k+1$$

and

$$A \equiv (x_{k+1} - x) / (x_{k+1} - x_k), \quad B \equiv 1 - A \qquad (3)$$

C. Cubic Spline Interpolation

This interpolation is slightly different from the previous one in that it interpolates using a curve rather than a straight line. In order to compute this curved line, cubic spline interpolations are using derivatives.

$$K = \frac{y''}{(1+y'^2)^{\frac{3}{2}}} \tag{4}$$

The spline attempts to minimize the bending, as it must pass through all data points, to produce a continuous interpolation.

$$\begin{cases} q'i(x_i) = q'_{i+1}(x_i) \\ q''i(x_i) = q''_{i+1}(x_i) \end{cases} \quad 1 \leq i \ \leq n-1 \tag{5}$$

V. Classification of Polynomials

It was mentioned that this study attempts to use a subset of the complete dataset in order to achieve memory efficiency. These subsets are parts from the dataset that were sampled using optimized interval values. In this way, the polynomials can be accurately modelled with very low levels of errors. To illustrate this, it is known from the dataset that for a motion that lasts second 120 frames are stored. For 6 seconds this makes a total of 720 frames. Instead of using all these frames to construct the polynomials, one can choose a step value of 10 in order to employ 72 frames for the polynomial creation. In order to reconstruct the motion from the polynomial, the following function is defined to match the class of the data with the sample:

$$S(f) = round\left(\frac{g*a}{f}\right) \tag{6}$$

where f denotes the frame, g is the group and a is the step value.

VI. Experimental Results

This section presents the results of using the above mentioned polynomial interpolation methods for learning human movements. Tables I and II show the Mean Absolute Error (MAE) and Mean Squared Error (MSE) values for the reconstructed human movements using the interpolation methods. It can be seen that the error rates increase, as the step and group variables take larger values.

TABLE I: Error Values by Interpolation Calculations for DS1

Group & Step	Type Of Interpolation	Lagrange	Spline	Cubic Spline
	Joint Name			

Group=6
Step = 2
	Head_1	MSE= 0.0820
		MAE= 0.1050
			MSE= 0.0256
			MAE= 0.0616
				MSE= 0.0070
				MAE= 0.04

Group =6
Step =10
	Head_1	MSE= 0.4940
		MAE= 0.3170
			MSE=0.2810
			MAE=0.2630
				MSE= 0,099
				MAE= 0,194

Group =4
Step =6
	Root_1	MSE= 0.0004
		MAE= 0.01
			MSE= 0.0007
			MAE= 0.0108
				MSE= 0.0002
				MAE= 0.0083

Group =6
Step =8
	Root_1	MSE= 0.0316
		MAE= 0.0353
			MSE=0 .0005
			MAE=0 .0088
				MSE= 0.0001
				MAE= 0.007

TABLE II: Error Values by Interpolation Calculations for DS2

Group &Step	Type Of Interpolation	Lagrange	Spline	Cubic Spline
	Joint Name			

Group =6
Step =2
	Head_1	MSE=0,1522
		MAE=0,103
			MSE=0,0313
			MAE=0,0557
				MSE=0,0223
				MAE=0,042

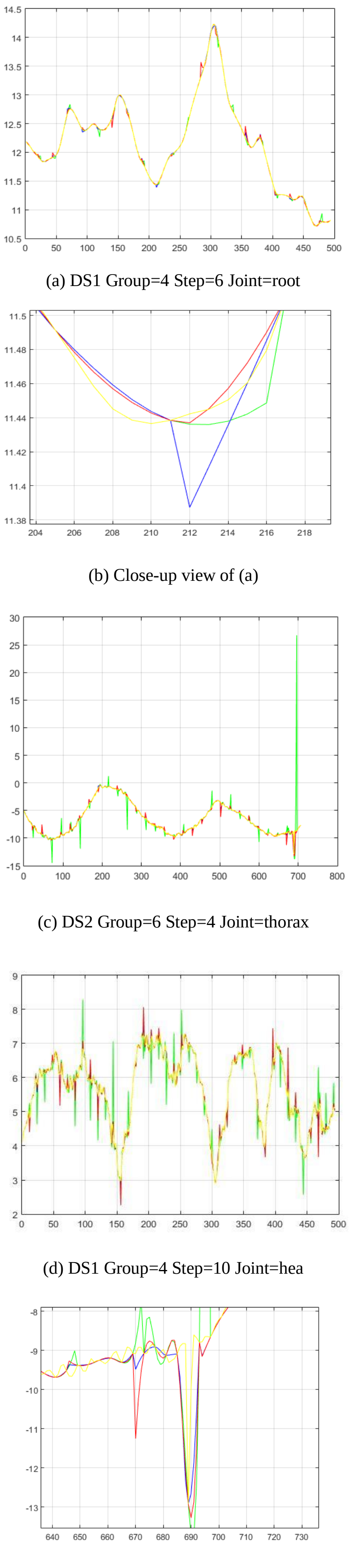
Group =6
Step =10
	Head_1	MSE=0,7351
		MAE=0,3379
			MSE=0,129
			MAE=0,187
				MSE=0,059
				MAE=0,145

Group =4
Step =6
	Root_1	MSE=0,0005
		MAE=0,008
			MSE=0,00046
			MAE=0,008
				MSE=0,0004
				MAE=0,0068

Group =6
Step =8
	Root_1	MSE=0,0039
		MAE=0,0233
			MSE=0,0122
			MAE=0,026
				MSE=0,002
				MAE=0,014

The obtained polynomials are depicted in result graphs. Color representations are as follows: yellow line is the original data while red, blue and green lines represent the polynomial generated using spline, cubic spline and

Lagrange interpolation methods, respectively.



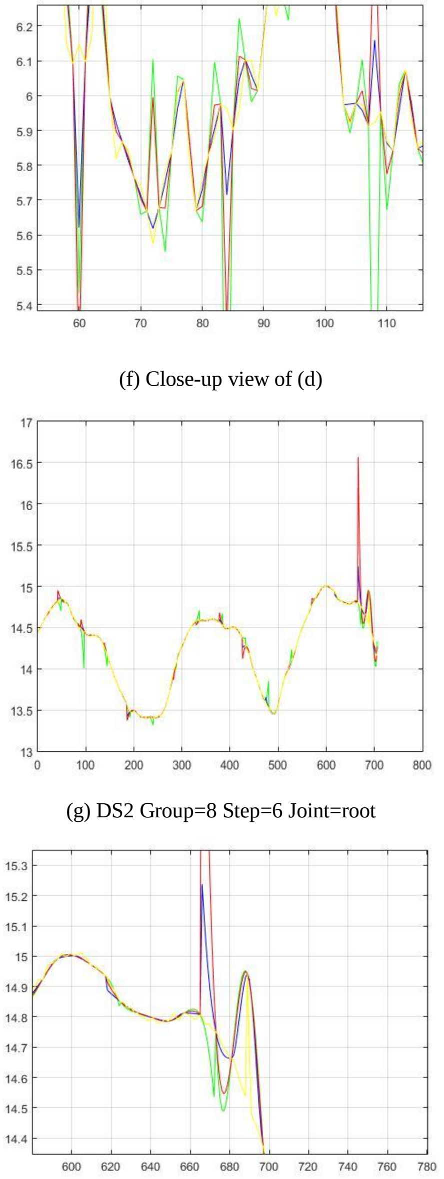
(a) DS1 Group=4 Step=6 Joint=root

(b) Close-up view of (a)

(c) DS2 Group=6 Step=4 Joint=thorax

(d) DS1 Group=4 Step=10 Joint=hea

(e) Close-up view of (c)



(f) Close-up view of (d)

(g) DS2 Group=8 Step=6 Joint=root

(h) Close-up view of (g)

Fig. 4. Graphs.

It can be seen that there are some deviations from the original data depending on the selection of the step and group variables. Moreover, each graph has a close-up plot derived from these deviations.

In addition to the visual differences in the results of the interpolation methods, a non-parametric statistical test known as Mc Nemar's test [13], [14] was applied to the error results of the interpolation methods. According to the results of this test, there are statistically significant differences between interpolation types. It is clear that the cubic spline has resulted in the most accurate reconstruction from the polynomial, while spline interpolation produced a better result than Lagrange interpolation.

TABLE III: MC NEMAR'S ANALYSIS FOR INTERPOLATION RESULTS

Interpolation Type	Lagrange	Spline	Cubic
Lagrange	**0**	**2.5997** ↑	**2.9711** ↑
Spline		**0**	**3.7139** ↑
Cubic			**0**

VII. Conclusion

This paper attempted to use various interpolation methods namely spline, cubic spline and Lagrange interpolation in order to model human movements obtained from a test sequence. It was found that the interpolation methods can achieve significant accuracy and are more memory efficient than the conventional learning algorithms with the use of step and group variables. Future work will investigate the use of a multiple model that may reduce the reconstruction errors obtained using these interpolation methods. We are also planning to use these results in an application that will work on the data obtained from sensors acquiring human motion in real time.

References

[1] E. Halici and E. Bostanci, "Analysis of training performances of interpolation methods for modelling human body motion," in *Proc. International Conference on Analysis and Its Applications (ICAA),* Kirsehir, Turkey, 2016.

[2] E. Halici and E. Bostanci, "Modelling human body motion using Lagrange interpolation," *24 IEEE Sinyal İşleme ve İletişim Uygulamaları Kurultayı (SIU)*, Zonguldak, Turkey, 2016.

[3] S. Parveen and R. Tokas, "Faster image zooming using cubic spline interpolation method," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 3, no. 1, 2015.

[4] A. Safonova and J. K. Hodgin, "Analyzing the physical correctness of interpolated human motion," in *Proc. Eurograph-ics/ACM SIGGRAPH Symposium on Computer Animation*, 2005, pp. 171-180.

[5] A. Schaal, Dynamic movement primitives-a framework for motor control in humans and humanoid robotics.

[6] S. Li and T. Hseng, "Walking motion generation, synthesis, and control for biped robot by using PGRL, LPI," *Adaptive Motion of Animals and Machines*, Springer Tokyo, 2006, pp. 261-280.

[7] M. Prada and A. Remazeilles, and A. Koene, "Dynamic Movement primitives for human-robot interaction comparision with human behavioral observation," *Intelligent Robots and System (IROS)*, pp. 1168-1175, 2013.

[8] R. Vuga and M. Ogrinc, A. Gams, M. Petric, N. Sugimoto, A. Ude, and J. Morimoto, "Motion capture and reinforcement learning of dynamically stable humanoid movement primitives," in *Proc. International Conference on Robotics and Automation (ICRA)*, 2013, pp. 5284-5290.

[9] O. Williams, A. Blake, and R. Cipolla, "A sparse probabilistic learning algorithm for real-time tracking," in *Proc. Ninth IEEE International Conference on Computer Vision, Computer Vision Proceedings*, IEEE, 2003.

[10] Dietterich and G. Thomas, "Approximate statistical tests for comparing supervised classification learning algorithms," *Neural Computation*, vol. 10, no. 7, pp. 1895-1923, 1998.

[11] J. Kober and J. Peters, "Policy search for motor primitives in robotics," *Advances in Neural Information Processing Systems*, 2009.

[12] C. Daniel, G. Neumann, and J. Peters, "Policy search for motor primitives in robotics," *Advances in Neural Information Processing Systems*, 2009.

[13] J. V. D. Berg, D. Duckworth, S. Miller, H. Hu, A. Wan, X. Fu, K. Goldberg, and P. Abbeel, "Superhuman performance of surgical tasks by robots using iterative learning from human-guided demonstrations," in *Proc. 2010 IEEE International Conference on Robotics and Automation (ICRA),* IEEE, 2010.

[14] B. Bostanci and E. Bostanci, "An evaluation of classification algorithms using Mc Nemars test," in *Proc. Seventh International Conference on Bio-Inspired Computing: Theories and Applications*, pp. 2012, pp. 15-26.

E. Halici graduated from Statistics Department at Kirikkale University in 2013. Currently, he is pursuing an M.SC. degree in computer engineering from the Institute of Natural and Applied Sciences at Ankara University. His research interests include artificial intelligence and machine learning.

He also works for Oplog Company in Turkey as a software developer.

E. Bostanci received a BSc degree in Computer Engineering Department from Ankara University, Turkey in 2007. Consequently, he joined the same department as a research assistant and completed his MSc on real-time battlefield simulation in 2009. He received his PhD from School of Computer Science and Electronic Engineering, University of Essex, United Kingdom in 2014 with his thesis on real-time user tracking for augmented reality.

He started working as a planning officer designate since June, 2014, in the Gendarmerie Schools Command where he conducted the research for analysing crime scenes. He has been promoted to second lieutenant in January, 2015. He currently continues his post in Ankara University as an assistant professor.

His research interests include different yet closely related aspects of computer science from image processing, computer vision and graphics to artificial intelligence and fuzzy logic as well as mathematical modelling and statistical analysis. He recently developed a vision-based user tracking system for various augmented reality applications for cultural heritage in particular.

Dr. Bostanci has been involved in technical committees for several conferences and acted as a reviewer for various journals.