Unsupervised Learning Motion Models Using Dynamic Time Warping

Marek Kulbacki[1], Jakub Segen[2], Artur Bak[3]

[1] Systems Research Institute, Polish Academy of Sciences, PL 01-447 Warsaw, 6 Newelska St., e-mail: kulbacki@ibspan.waw.pl
[2] Bell Labs, Murray Hill, USA, e-mail: segen@ieee.org
[3] e-mail: abak@traf.ict.pwr.wroc.pl

Abstract. This paper concerns essential, practical problem in automatic animation human-like figures with the support of informatics technologies connected with motion capture domain. The main problem we want to solve is partition set of primitive motions into appropriate groups according to similarity between motions. Up to now, experiments in systems of this kind, appeared be not too adequate to needs. In this situation, we had been faced with the necessity of creating new methods for supporting process of managing motion data. We construct motion models to easier extract features of given motions. Using these models we propose measure of discrepancy between motions. It shows how two motions are similar to each other, normalizes length of motions and decreases high dimension of considered motion data, so clustering may take place in dimensionally reduced space.

Keywords: dynamic time warping, motion capture, computer animation, motion grouping, classification, probabilistic motion models.

1 Introduction

Currently a motion capture technique [7] is very willingly used for creation of realistic human-like figures animations. There are two most often used types of this technique. In the first case reflective markers are fixed on joints of alive actor and the motion of markers is tracked. In the latter case magnetic sensors are fixed on actor joints. These sensors are tracking disturbances of magnetic field during motion. In order to achieve realistic animation there is recorded motion of each human joint. This causes that it is necessary to describe motion with a large set of data. Such data are hard to process in some fields of applications. This problem is especially visible in use of multimedia databases. Managing the tremendous amounts of data is often supported by clustering and classification methods. It is not easy to find such methods for motion sequences.

In our approach we try to solve this problem. In consecutive sections we describe problems and propose solutions that make up the method of motions clustering and classification. At the beginning of article, we describe motion representation that is most appropriate to methods used by us. Next we show the method of motions standardization and definition of distance measure. We base on classic DTW and extend DDTW [2] method with specific for our purposes discrepancy measure. Using distance measure and motion standardization, we describe clustering based on classical Agglomerative Clustering algorithm [4,6,5]. We also describe motion classification relying on probabilistic generic motion models defined by us. At the end we indicate proposed application of our solutions.

2 Motion Representations

We utilize several motion representations, for different levels of abstraction. A motion is a time-varying function which provides the configuration of an articulated figure at a time. Input representation is an original motion capture sequence; it is represented as Raw Data Model (RDM). We denote a RDM by $m(t) = (p_0(t), q_0(t), q_1(t), \ldots, q_L(t))^T$, where $p(t) \in \mathbb{R}^3$ and $q_1(t) \in \mathbb{R}^3$ describe the translational and rotational motion of the root segment[1], and $q_i(t) \in \mathbb{R}^3$ gives the rotational motion of the i^{th} joint for $1 \leq i \leq L$. From RDM we extract shorter Primitive Motions (PM). Their main feature is that they are uniform. For each Primitive Motion, we generate Specific Model (SM) as a Timmer splines parameters calculated according to RDM data. We denote SM as $s(m) = (s_1(m), s_2(m), \ldots, s_M(m))^T$, where M is a number of SM parameters, and $s_i \in \mathbb{R}^3$. Specific Model is used by clustering algorithm. For every motion group a probabilistic Generic Model (GM) is evaluated. GM is a set of parameters described by Gaussian distributions over parameters of Specific Models of particular group. From these distributions new PM's can be generated (which haven't been provided as motion capture files). Fig. 1 shows the process of determining various motion representations. More detailed description of each model is described in [13].

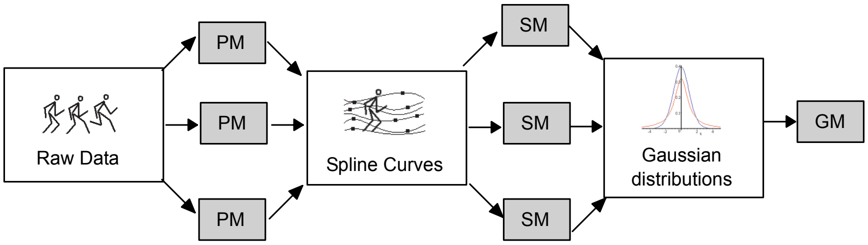


Fig. 1. Motion representations transformation process

[1] root segment - the most important joint in the human skeleton (base joint)

3 Motion Comparison and Standardization

Our motion sequences are represented as sets of time series. Several pattern matching techniques, able to deal with time sequential data, have been applied to match movement patterns: Dynamic Time Warping [3,8], Hidden Markov Models [9,12], Artificial Neural Networks [10].

We have chosen classic DTW approach as a tool to match motion sequences; it is conceptually simple and effective, allowing sufficient flexibility in time-alignment between test and reference motion sequence. Articulated objects such as human figures are usually represented as rotation hierarchies parameterized by a whole-body translation, a whole-body rotation, and a set of joint angles. Here motion is described by a set of motion curves, each giving the values of one of the model parameters as function of time. Using DTW we are able to solve two problems:

- find measure of discrepancy between two motion sequences,
- normalize motion sequences in the number of frames regarded.

In our case, time warping is applied in the discrete time domain to register the corresponding motion parameter signals such as joint angles. We warp each motion curve independently, so we can consider just a single curve $Q_{l,d}(t)$.[2] It represents movement of one joint for specified degree of freedom. Hereafter we call it *time series*. This definition goes for our motion sequences, because each of them is a set of motion curves at the specified period of time. The number of frames constraints include a set of $(Q_{l,d}[i], t[i])$ pairs each giving the value $Q_{l,d}$ at the specified time i. Thanks to it each motion curve may be represented as an *identical length* time series. For two motion sequences comparison we must warp independently each corresponding time series $Q_{1,x}$ and $C_{1,x}$.

In the description concerning DTW we use some text and definitions from Keogh and Pazzani [1,2] whom we gratefully acknowledge. To match two motion sequences we use an n-by-m matrix, where the element (i, j) of the matrix contains the distance $d(q_{l,d}[i], c_{l,d}[j])$ between two points $q_{l,d}[i]$ and $c_{l,d}[j]$. Each warping path W is given by mapping between $Q_{l,d}$ and $C_{l,d}$:

$$W = w_1, w_2, \ldots, w_K \qquad max(m, n) \leq K < m + n - 1 \tag{1}$$

Optimal solution is specified by :

$$DTW(Q_{l,d}, C_{l,d}) = min\left\{ \sum_{m=0}^{K} d\big(q_{l,d}[i_m], c_{l,d}[j_m]\big) \right\} = min\left\{ \sum_{k=1}^{K} w_k \right\}. \tag{2}$$

[2] where: $l \in \{1 \ldots L\}$, L - number of joints, $d \in \{x, y, z\}$ - degree of freedom

3.1 Improved Distance Measurement

Each joint in motion sequence is defined as a set of time functions for specified degrees of freedom. To determine number of elements of all functions to be equal, we carry out normalization to the number of considered motion frames. Discrete value sequences obtained in that way, are useful as elements to compare motion sequences together. Moreover there are computed derivatives for each frame.

From previous sections we know that motion recognition is based upon the comparison of corresponding joints in two motion sequences. To do it well we have to find specified distance measure, making use of this in motion sequences comparison process. We use $d\big(q_{l,d}[i], c_{l,d}[j]\big)$ to denote the distance between i^{th} and j^{th} frame of two corresponding joints to be compared. Any function that meets the above properties is a legitimate metric on the elements space. Standard Euclidean metric is good to compare single points but not appropriate here, where time series are compared. To find measure, that gives consideration to adjacent values of time series, and is sensitive on the local changes among time series elements, we have extended Keogh and Pazzani's [2] measure. It is now composed of two components:

1. Euclidean distance between two points $q_{l,d}[i]$ and $c_{l,d}[j]$,
2. square of the difference of the estimated derivatives of $q_{l,d}[i]$ and $c_{l,d}[j]$.

The first part gives information about offset between points to be compared. The second part adds the "intelligence" to the entire measure. Thanks to this we are able to deal with situations where examinated sequences are not different enough. We use the following method for estimating derivative from joint data:

$$D_x\Big[q_{l,d}[i]\Big] = \left.\frac{q_{l,d}[i] - q_{l,d}[i-1]}{t[i] - t[i-1]}\right|_{t[i]-t[i-1]=1} = q_{l,d}[i] - q_{l,d}[i-1], \quad 1 < i \leq n \tag{3}$$

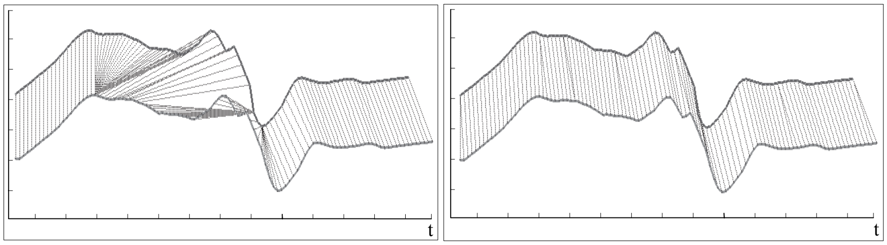
This estimate is the slope of the line through the point $q_{l,d}[i]$ and its left neighbor. Note the estimate is not defined for the first element of the sequence. Instead we use the estimate of the second element.

On the basis of above equations we have created new measure. The full definition of this measure is

$$d(q_{l,d}[i], c_{l,d}[j]) = \sqrt{|c_{l,d}[j] - q_{l,d}[i]|^2} \cdot \Big(D_x\Big[c_{l,d}[j]\Big] - D_x\Big[q_{l,d}[i]\Big]\Big)^2 \tag{4}$$

This equation doesn't meet all conditions concerning distance measure, so we called it *measure of discrepancy* between elements $q_{l,d}[i]$ and $c_{l,d}[j]$.

The weak point of standard DTW is that it only considers data points on Y-axis value. Keogh and Pazzani's DDTW algorithm [2] takes into consideration a derivative of the signal. We base on it and propose two components extension naming it Value-Derivative Dynamic Time Warping (VDDTW). VDDTW's computational cost is similar to DTW, just as is the case with DDTW: "DDTW's time complexity is $O(mn)$, which is the same as standard DTW" [2].



(a) (b)

Fig. 2. Examples of some experimental datasets: **a)** the alignment produced by classic DTW **b)** the alignment produced by VDDTW

3.2 Extention VDDTW to Entire Motion Sequence

As we know the previous discussion referred to the single joint comparison for specified degree of freedom. This is only a small element of the whole motion sequence. Our skeleton is made up of eighteen joints, so this comparison operation must be applied to each joint separately, taking into consideration existing degrees of freedom. The full motion warping algorithm is shown below:

```
REQUIRE motion sequence $A$,  motion sequence $B$
ENSURE All warping cost,  Warped Sequence
for each existing joint
    for each existing degree of freedom
        TmpCost ⇐ Least Warping Cost;  TmpMotion ⇐ Reverse Warped Path
    end for
    AllCost ⇐ AllCost+TmpCost;    update NewMotion using TmpMotion
end for
```

As an output we get whole cost used to warp motion sequence B into motion sequence A. Additionally this algorithm produces Warped Sequence of motion B (B_W) which have a length of motion A. This case requires explanation. The question is, how to get Warped Sequence of motion B? During the warping motion B into motion A three cases are distinguished:

1. substitution - 1:1 correspondence of successive samples;
2. deletion - multiple samples of B map to a sample of A;
3. insertion - a sample of B maps to multiple samples of A.

Of course cases discussed above concern process of single joint warping in the range of the whole motion sequences. For the following explanations, we assume that signal Q represents joint from motion B and signal C represents joint from motion A. We can say that signal Q is warped into C, and the warped signal is denoted by Q_W. Then if $q_{l,d}[i]$ and $c_{l,d}[j]$ are related by substitution it follows that $q_{l,dW}[j] = q_{l,d}[i]$. In case of a deletion, where multiple samples of $Q, (q_{l,d}[i], q_{l,d}[i+1], \ldots, q_{l,d}[i+k])$, correspond to one $c_{l,d}[j]$, $q_{l,dW}[j] = mean(q_{l,d}[i], q_{l,d}[i+1], \ldots, q_{l,d}[i+k])$. Finally, an insertion implies that one sample of $Q, q_{l,d}[i]$, maps to multiple samples of $C, (c_{l,d}[j], c_{l,d}[j+1], \ldots, c_{l,d}[j+k])$. In this case, the values for $q_{l,dW}[j], q_{l,dW}[j+1], \ldots, q_{l,dW}[j+k]$ are determined by calculating a Timmer cubic B-spline distribution around the original value $q_{l,d}[i]$.

Presented algorithm is applied in this work to normalize length of motions (Warped motion) and as a measure of discrepancy (All warping cost) used for motion clustering. Measure of discrepancy between motion m_1 and m_2 (using specific models of these motions) we denote as $\delta(s(m1), s(m2))$. Total time complexity is strictly dependent on joint number (L) and length of motion sequences A and B(respectively m and n). It is about $O(|L| \cdot |m| \cdot |n|)$ or after reduction of the searching space $O(|L| \cdot |n| \cdot |K|)$, where delimiter $K \leq \frac{m}{2}$.

4 Clustering and Classification

Clustering of motions capture sequences is not simple unless the distance measures and standardization of motions are well defined. Since when we have these mechanisms based on VDDTW the clustering algorithm itself is similar to other domains clustering methods. However we define a few specific elements that are necessary for the next classification and future use of clustered motions set. It concerns especially the clustering representation relied on probabilistic generic models.

The main goal of clustering is partition of primitive motions into appropriate clusters. It should be done according to similarity between motions. This similarity is identified with distance between motions[3] defined in previous section. The less distance between two motions the more similar these motions are. We have to require clustering process to divide motions set in proper way. Motions of one cluster should be similar and motions of different clusters should be dissimilar to each other. Beside the motions partition, we also need certain description of each cluster. The set of all clusters descriptions is called clustering representation. Division of motions set into clusters and clustering representation we treat as main tasks of clustering process.

4.1 Cluster Finding

The method we use to partition set of motions is classical Agglomerative Clustering algorithm. Disadvantage of this method is high time complexity.

[3] in a sense of measure of discrepancy

It has an impact on the fact that in every step we have to check all possible partition spaces. Solution like this is not always acceptable, especially in real time animation domain. The advantage is certainty that we find global optimal solution in respect of criterion of acceptance. Suppose we have motions set - R, that contains N primitive motions $(m_1, \ldots, m_N)$. Actual set of groups from R set we denote as X. The number of motions in any group G_i is denoted as N_i. In the first step we set number of groups equal to number of motions. Initially every motion m_n from set R belongs to separate group G_i in set X, where $n, i \in [1..N], N_i = 1$. In the consecutive steps of clustering algorithm, adjacent groups are merged into new larger group. We break algorithm when *stop condition* is satisfied. In a single step two most adjacent groups are merged so we have to define distance between these groups. It is based on internal average discrepancy between primitive motions in the group. This is not pure distance measure but hereafter we call it distance for clarity. Average internal discrepancy in a group G_i is equal to average from all possible discrepancies δ as we can compute between all primitive motions in this group:

$$\overline{\delta}_i = \text{average}\big(\{\delta(s(m_a), s(m_b)) \bigm| m_a, m_b \in G_i,\ a \neq b\}\big) \tag{5}$$

Distance D_{12} between two groups G_1 and G_2 is defined as discrepancy $\overline{\delta}_{12}$ between all primitive motions in new merged group $G_1 \cup G_2$. We compute distance matrix M that contains distances between all currently existing groups. Matrix M is symmetrical $(D_{12} = D_{21})$, so we have to calculate distance only $\frac{K^2-K}{2}$ times[4]. On the base of matrix M we can choose two most adjacent groups for merge. These are groups G_i and G_j for which the distance D_{ij} is the least. We break algorithm when in the given step distance D_{ij} is greater than maximal acceptable distance of merge D_{max}:

$$D_{ij} > D_{max} \quad \forall i, j \in (1, \ldots, K) \tag{6}$$

4.2 Generic Model for Group of Primitive Motions

Clustering algorithm gives partitioning, of motion set R into groups. To effective utilize the partition it is important to define appropriate clustering representation. These are appropriate descriptions of groups. In this case for every group G_i from set X we calculate exactly one probabilistic description. It is formulated as a set of gaussian distributions. These distributions are calculated over each parameter of specific model among all primitive motions in given group. All probabilistic distributions for given group are encapsulated in parametric model of this group named *generic model.* In the Fig. 3 we can see dependencies between specific motions models in the given group and generic model for this group. Parameters s_l, are description for consecutive

[4] K - actual number of groups in the set X

frames, where $l \in (1, \ldots, M)$ and M is the number of parameters of specific models s_m. For each of parameter s_l we evaluate Gaussian distribution over values of these parameters for all primitive motions in the given group. This distribution is denoted in generic model as two parameters: average av_l and variance v_l:

$$av_l = \frac{1}{N_i} \sum_{m_n \in G_i} s_l(m_n), \qquad v_l = \frac{1}{N_i - 1} \sum_{m_n \in G_i} \left[s_l(m_n) - av_l \right] \tag{7}$$

For given specific motion model s can be evaluated function of probability density for known parameter s_l according to distribution of generic model GM_i:

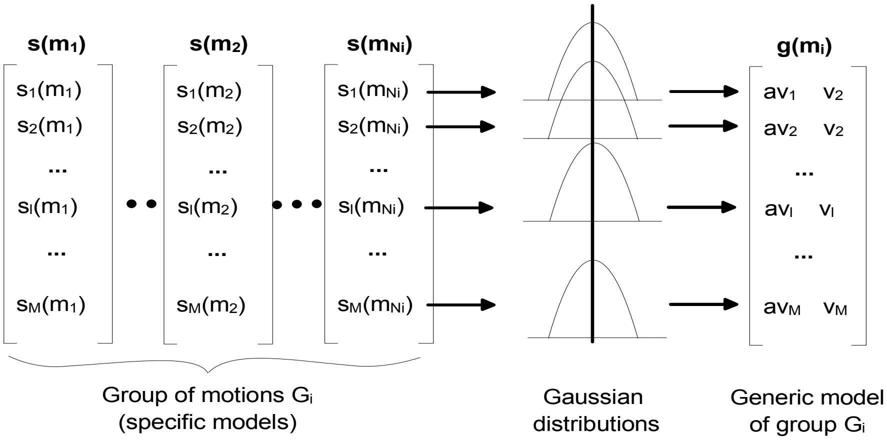


Fig. 3. Evaluation of probabilistic generic models

$$g_{i,l}(s_l(m)) = \frac{1}{\sqrt{v_l 2\pi}} \exp\left\{ - \frac{[s_l(m) - av_l]^2}{2v_l} \right\}. \tag{8}$$

So far we did not say about problem of different primitive motions length. In effect number of parameters M in specific models may differ between particular motions. In that case, it is difficult to compute the number of generic model parameters. We need the method of normalization all specific motions in a given group into the same number of parameters. We are finding prototypes for every group. It is similar to Oates method [6]. Prototype T_i is the most typical motion in the given group. This motion minimizes average of discrepancy with all the rest members of group G_i. In consecutive step, we normalize all specific motions from the group using VDDTW algorithm. Specific model of typical motion in the group is used as template signal in VDDTW algorithm. As a result of this operation, we get a set of specific motions in the same number of parameters. Thanks to this, it is possible to compute Gaussian distributions over values of primitive motions parameters. Generic model for the group is evaluated on the base of set of normalized specific motions. Specific models of the rest motions in the group are normalized into the lengths of specific model of typical motion.

4.3 Motion Classification

We can treat the generic model as the probabilistic generator of specific models (specific models of primitive motions that haven't been delivered in input motions set). We assume that all primitive motions that belong to one group are generated by the same generic model. We can also assume that every motion in primitive motions domain is generated by exactly one generic model. The main application of generic model is classification of motions that are outside the input motions set, into appropriate groups. To perform classification we have to choose GM that probably generates considered motion. Likelihood that given generic model GM_i generates motion m may be treated like similarity of motion m to the group connected with GM_i:

$$\theta_i\big(s(m)\big) = P(G_i)\prod_{l=1}^{M} w_l g_{i,l}\big(s_l(m)\big) \tag{9}$$

The argument of above similarity function is specific model of motion m. Weight w_l is related with the joint that is described by parameter s_l of specific model for motion m. Component $g_{i,j}$ is probability density for parameter s_l in the group G_i. Component $P(G_i)$ describes likelihood of situation that any primitive motion belongs to the group G_i (it has been generated by GM_i). This likelihood can be given *apriori* or can take into consideration relative probability of this group in motions set. Because of limited set R in regards to all primitive motions space, in this classification algorithm each group has the same likelihood $P(G_i) = \frac{1}{K}$. Before we compute measure θ we must normalize given primitive motion m according to the specific motion of typical motion T_i. To do it we utilize VDDTW algorithm. Finally primitive motion is classified into the group G_i for which similarity measure $\theta_i\big(s(m)\big)$ reaches maximal value. Classification equation is given as follows:

$$h\big(s(m)\big) = \underset{i\in\{1,2,\ldots,K\}}{\operatorname{argmax}} \big[\theta_i\big(s(m)\big)\big] \tag{10}$$

4.4 Conclusions

This paper presented preliminary results of an experimental study of algorithm for human motions organization. In particular our method comprise full motion models definitions [13], algorithms of comparison and clustering of primitive motions. We did not say about problem of motions segmentation (extraction uniform motions in any motions sequences). This is very important, because it has an influence of the accuracy of our clustering algorithm.

Our main goal is to expand this ideas onto automatic animation domain. The main application of presented methods is automatic motion synthesis in tools for creation realistic animations of human like figures. We were testing these algorithms on a small training set of motions. It is hard to prove efficiency of this method because it is still developed.

References

1. E.J.Keogh, M.J.Pazzani: Scaling up Dynamic Time Warping to Massive Dataset, Principles of Data Mining and Knowledge Discovery, pp. 1-11, 1999
2. E.J.Keogh, M.J.Pazzani: Derivative Dynamic Time Warping, First SIAM Conference on Data Mining, 2001, Chicago, USA
3. D.J.Berndt, J.Clifford: Using Dynamic Time Warping to Find Patterns in Time Series, *KDD* Workshop, pp. 359-370, 1994
4. A.K.Jain, R.C.Dubes: Algorithms for Clustering Data, Prentice Hall, Englewood Cliffs, N.J., 1988
5. T.Oates, L.Firoiu, P.R.Cohen: Clustering Time Series with Hidden Markov Models and Dynamic Time Warping, Proc. IJCAI-99, pp. 17-21
6. T.Oates: Identifying Distinctive Subsequences in Multivariate Time Series by Clustering, Proc. 5-th International Conference on Knowledge Discovery and Data Mining, pp. 322-326, 1999
7. S.Dyer, J.Martin, J.Zulauf: Motion Capture White Paper, December 1995
8. A.Witkin, Motion Warping, Computer Graphics, Vol. 29, pp. 105-108, 1995
9. M.Brand, HertzmannA.: Style machines, In The Proc. of ACM SIGGRAPH 2000, pp. 183-192, 2000
10. Y.Guo, G.Xu, S.Tsuji: Understanding Human Motion Patterns, ICPR94, pp.325-329, 1994
11. T.Darrell, A.Pentland: Space - Time Gestures, CVPR/NYC, June 15-17, 1993
12. J.Yamato, J.Ohya, K.Ishii: Recognizing Human Action in Time-Sequential Images using Hidden Markov Model, IEEE CVPR, pp. 379-385, 1992
13. M.Kulbacki: Principal methods for motion synthesis of human-like figures, MA Thesis, Wroclaw University of Technology, 2001