Robust Human Action Recognition Using Dynamic Movement Features

Huiwen Zhang[1,2(✉)], Mingliang Fu[1,2], Haitao Luo[1], and Weijia Zhou[1]

[1] State Key Laboratory of Robotics, Shenyang Institute of Automation, Shenyang 110016, China
zhanghuiwen@sia.cn
[2] University of Chinese Academy of Science, Beijing, China

Abstract. Action recognition has been widely researched in video surveillance, auxiliary medical care and robotics. In the context of robotics, in order to program robots by demonstration (PbD), we not only need our algorithms to be capable of identifying different actions, but also to be able to encode and reproduce them. Dynamic movement primitives (DMPs), as a trajectory encoding method, are widely used in motion synthesize and generation. But at the same time it can also be applied to action recognition. With this idea, this paper extracts a kind of dynamic features from the original trajectory within DMP framework. The feature is temporal-spatial invariant. Based on the feature, FastDTW-KNN algorithm is proposed to solve the recognition task. Experiments tested on HAR dataset and handwritten letters dataset achieved an excellent recognition performance under a large data noise, which has verified the effectiveness of our method. In addition, comparative recognition experiments based on the original feature and our extracted dynamic feature are conducted. Results show that the dynamic feature is robust under temporal and spatial noise. As for classifiers, we compared our method with KNN, SVM and DTW-KNN followed with a detailed analysis of their advantages and disadvantages.

Keywords: Action recognition · DMP · DTW

1 Introduction

With the development of robot technology, we are increasingly expecting robots to help us in our daily lives. To achieve this goal, we need a simple and intuitive way to program robots. Learning from demonstration (LfD) [1] provides such a mechanism. But we met a lot of problems and challenges in LfD. First, in order to imitate human actions, the robot needs to "understand" them, which involves in action recognition. This problem has been widely researched in many different fields, such as image processing, computer vision, pattern recognition and machine learning.

If we solve action recognition using image inputs captured by vision sensor, which happens a lot in computer vision domain, we call them vision-based

Y. Huang et al. (Eds.): ICIRA 2017, Part I, LNAI 10462, pp. 474–484, 2017.
DOI: 10.1007/978-3-319-65289-4_45

methods. Generally, this kind of methods include the following three steps: (1) detecting the motion information from the image frame and extracting the underlying features; (2) modeling the behavior pattern or the action; (3) mapping low level visual features into high level semantic information like action categories. Usually quality of feature extraction is particularly important for visual-based methods. We can divide visual features into two categories: static and dynamic. Static features such as size, shape, color depth, etc., due to complex background, occlusion, lighting changes and other factors, it is difficult to ensure recognition performance. Dynamic features such as speed, trajectory, optical flow, etc., is essential for action recognition. But obtaining these features from video or image sequence is both non-trivial and difficult. Meanwhile, the quality of these features will suffer from the limitations mentioned before. Therefore, this article directly uses inertial measurement unit (IMU) or wearable sensors to obtain dynamic information, then recognition task is conducted based upon them, which we think is more reliable and suitable in robot behaviour learning domain.

Then, the action recognition problem becomes a pattern recognition problem of time series signal. For time series data analysis, HMM has obtained a large number of successful applications, especially in the field of speech recognition [2,3]. DTW [4], act as a signal preprocessing method, can also be used to action recognition when combined with other distance-based classification method. But all of these methods suffer from a high computational complexity. So in order to use a simple classifier while still maintains a high recognition performance, we need a kind of robust features. This paper constructs the robust features based on DMP, then uses FastDTW-KNN as our classifiers, which obtained a high and reliable recognition accuracy. The whole framework of our method is shown in Fig. 1.

The main contributions of this paper are as follows: (1) Constructing a kind of robust dynamic feature with DMP. The feature is easy to calculate and very suitable for recognition task; (2) The FastDTW-KNN method is proposed to classify the feature, which can improve the recognition accuracy, but also keep a low computation complexity; (3) A large number of comparative experiments are conducted to compare different classification algorithms and features.

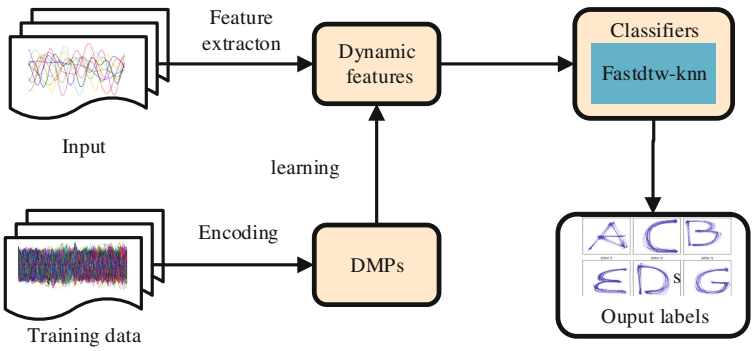

Fig. 1. The proposed recognition framework.

The subsequent chapters of this paper are organized as follows: The Sect. 2 introduces related work and the background of DMP. The Sect. 3 elaborates the method of using DMP to obtain dynamic features. Section 4 introduces our classifier followed with a lot of contrast experiments in Sect. 5. Finally, we give a brief conclusion and outlook of the future work.

2 Related Work

Most researches on action recognition focus on using computer vision methods [5,6]. Generally, they include two steps. First, extracting features from image or video data, and then a classifier is used to recognize actions in feature space. Static features is one kind of important features which we can obtain from the source data. They are mainly used for describing the size, color, edge, contour, shape and depth information of the human body. With these information, a good classifier can be constructed. Carsson et al. [7] implemented an action recognition algorithm by shape matching. Another kind of features are dynamic features, and they are considered to be a very important information in computer vision, such as optical flow, space-time features. Danafar et al. [8] used optical flow and SVM to recognize actions in the context of video surveillance. However, these dynamic feature extraction methods are complex in their own right. Besides, they are prone to be affected by environment. More importantly, in the context of robot behavior learning, we are not only seeking a mechanism for action recognition, but also hoping that it can be used for motion encoding and reproduction. DMP provides such a mechanism.

DMP was first proposed by Ijspeert, Schaal et al. [9]. Its main idea is to use our well-known second-order differential equations to represent movements, and then adjust it through a nonlinear term to achieve our desired attractor landscapes. Given the good characteristics of DMP, it is used in various fields such as control, planning, and learning [10,11]. Many improvements and extensions of DMP are also presented within a few decades, such as rhythmic DMP for gait research [12], DMP with obstacle avoidance [13], coupling multiple DMPs in Movement coordination [14] and probabilistic DMP [15]. As a powerful tool, DMP has also been used in imitation learning [16]. But all of these work are concentrated in the motion encoding, representation and generalization, instead, this paper will systematically study the DMP in the context of action recognition. To demonstrate its advantages, this paper compared a variety of commonly used classifiers, and proposed a FastDTW-KNN classification method.

3 Dynamic Feature Extraction

3.1 Modeling Movements with DMP

A DMP model can be represented by the differential equations shown in Eq. (1), which can be understood as the effect of a linear spring damping system subjected to an external force disturbance.

$$\begin{aligned} \tau \dot{v} &= \alpha_v \left[\beta_v \left(g - x \right) - v \right] + f \\ \tau \dot{x} &= v \end{aligned} \tag{1}$$

Where x and v represent position and velocity respectively; x_0 and g stand for the start and target position; τ is time scaling factor; α_v and β_v are constants used for control the damping characteristics of the system. f is a nonlinear function which can generate complex movements by modifying the weights parameters in f. It is defined as follows:

$$f(s) = \frac{\sum_i w_i \Psi_i(s)}{\sum_i \Psi_i(s)} s(g - x_0) \tag{2}$$

Where, $\Psi_i(s) = exp\left(-h_i(s - c_i)^2\right)$. h_i, c_i represent the precision and center of Gaussian function.f doesn't depend on time directly, but rather depend on a phase variable s , which changes from 1 to 0 as defined in Eq. (3).

$$\tau \dot{s} = -\alpha_s s \tag{3}$$

$$f_{target}(s) = \tau^2 \ddot{x} - \alpha_v \left[\beta_v (g - x) - \tau \dot{x}\right] \tag{4}$$

The formula in Eq. (3) is called the canonical system, which is equivalent to an internal clock. The system defined in Eq. (1) is called a transformation system. If we know f, we will be able to get our desired output via the transformation system. From the Eq. (2) we can see, f is decided by weights w. So in order to encode the movement, we need to find w. The interesting thing is that we found w is invariant in temporal-spatial space, which means whether you extend or shorten a given path or scale it by multiplying a random factor, as long as its contour is similar, then w basically keeps the same. In other words, w encodes a class of trajectories with similar topologies. This is also the main reason that feature w can be used for classification. The next section we will introduce the method used to calculate w.

3.2 Feature Calculation

The process of feature learning is the process of solving weights, which can be summarized as follows: (1) Calculate from demonstration trajectories; (2) Integrate formula (3) to get the phase variable; (3) Use Eq. (1) to get Eq. (4). Comparing formulas (2) and (4), we can find it is a function approximation problem. To achieve a better performance, we need to estimate a set of weights which can minimize $J = \sum_i \left(f_{target}(s) - f(s)\right)^2$. This problem is equivalent to minimizing the weighted minimum quadratic error of the following equation:

$$J_i = \sum_{t=1}^{T} \Psi_i(t) \left(f_{target}(t) - w_i \xi(t)\right)^2 \tag{5}$$

Where $\xi(t) = s(t)(g - x_0)$, this is a weighted linear regression problem. Its solution is:

$$w_i = \frac{\xi^T \mathbf{\Gamma}_i \mathbf{f}_{t\,\mathrm{arg}\,et}}{\xi^T \mathbf{\Gamma}_i \xi} \tag{6}$$

Where,

$$\xi = \begin{bmatrix} \xi(1) \\ \xi(2) \\ \cdots \\ \xi(T) \end{bmatrix}, \mathbf{\Gamma}_i = \begin{bmatrix} \Psi_i(1) & & & \\ & \Psi_i(2) & & \\ & & \cdots & \\ & & & \Psi_i(T) \end{bmatrix}, \mathbf{f}_{t\arg et} = \begin{bmatrix} f_{t\arg et}(1) \\ f_{t\arg et}(2) \\ \cdots \\ f_{t\arg et}(T) \end{bmatrix}$$

According to Eq. (6), we can get the weight of each base function. The feature vector is constructed by concatenating all weights together and then fed to our classifier.

4 Classifier

In the previous section, we introduced how to extract our dynamic features. Once the feature is acquired, a classifier is required to complete the classification task. In principle, as long as the feature is good enough, we can achieve a good classification performance even that a simple classifier is used. So in this paper we use the most simple KNN classifier, and compared with the SVM. In addition, we find that the extracted dynamic features have a certain drift in the time dimension, which may reduce the classification effect. We know that DTW is a classic way of dealing with time drift, but it is computationally expensive. In order to maintain a good classification effect, as well as minimize the computational complexity, we use a method called Fast DTW [17] to align our dynamic features, and then fed into the KNN classifier. DTW is used to measure the similarity of two sequential signals with different length, and is widely used in time series signal analysis. The basic principle is calculating distance metric with dynamic programming method. Distance between two sequences with length is defined as:

$$D(i,j) = Dist(i,j) + \mathtt{min}\left[D(i-1,j), D(i,j-1), D(i-1,j-1)\right] \tag{7}$$

DTW is used to measure the similarity of two sequential signals with different length, and is widely used in time series signal analysis. The basic principle is calculating distance metric with dynamic programming method. Distance between two sequences with length i, j is defined as:

For traditional DTW, in order to obtain $D(i,j)$, we must traverse the entire D matrix and the time complexity of the algorithm is $o(N^2)$. To speed up the calculation, there are usually two ways: (1) reduce the search space in D matrix; (2) data abstraction, reduce the length of the data. FastDTW combines both of them. It consists of three steps: coarsening, projection and refinement. Coarsening devotes to shrinking the original signal length without a big loss of accuracy. Projection finds a minimum-distance warp path with the coarse curve. Refinement refines the warp path projected from a lower resolution through local adjustments of the warp path. Through all of these operations, we can reduce the time complexity to $o(N)$. Using FastDTW we can get the distance metric accurately and quickly, then KNN gives us the estimated labels. This constitutes the FastDTW-KNN algorithm.

5 Experiments

This section will conduct a series of comparative experiments and compare our methods from multiple perspectives. We have two purposes. The first one is to verify the time advantage of FastDTW-KNN. The second is to show the good recognition performance when using the proposed dynamic features introduced in Sect. 3. In order to verify the temporal-spatial invariance, the test data will be randomly extended or cut off 10% length to introduce the temporal noise. Mean while, the test data is scaled by multiply a random factor to introduce spatial noise. Therefore, our test data is divided into four classes: the original non-polluting test data, data with temporal noise, data with spatial noise and data with temporal-spatial noise. As shown in Fig. 2, which takes letter "C" as an example. For simplicity, in later sections, graphs and tables, we use words"clean, temporal, spatial, T-S" represent these four different noises, respectively.

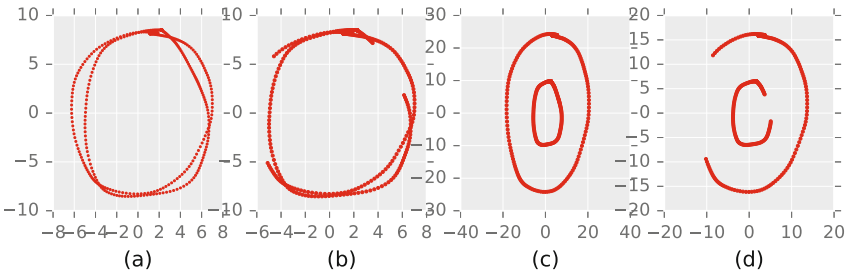

Fig. 2. Different kinds of noise added to sample character "C". (a) Stands for the original data, two samples; (b) Temporal noise is introduced by deleting some points randomly; (c) Add spatial noise by scaling the original data; (d) Both temporal noise and spatial noise are introduced

Experimental tasks can be summarized into three aspects. First, based on the HAR dataset, the recognition performance of KNN, DTW-KNN and FastDTW-KNN is investigated under the four test noise. Second, based on the hand written letters dataset, we investigate the recognition accuracy without any feature extraction. Thirdly, on the HW dataset, the recognition accuracy of different algorithms on different test noises is calculated based on the dynamic features introduced in our paper.

5.1 Datasets

This article uses two datasets. The first one is the UCI Human Activity Recognition (HAR) dataset [18]. The data is collected from 30 volunteers within an age bracket of 19–48 years. Each person performed six activities (walking, walking upstairs, walking downstairs, sitting, standing, laying). Using the embedded

accelerometer and gyroscope in smartphone carried by volunteers, we captured 3-axial linear acceleration and 3-axial angular velocity signals. The data has been video recorded to label the data manually. In our experiments, we randomly choose 70% of data as training set and the left 30% is used for test.

Another dataset is a handwritten letter dataset provided by Calinon [19]. This dataset is located in a package that they implemented for LfD. The dataset includes 26 Latin characters contributing to a total of 340 sample trajectories in two-dimensional coordinates. Velocity and acceleration information are also recorded. As before, 70% of the data is used for training and the rest is used for testing.

5.2 Experiments Results

Using the HAR data set, we compare the KNN, DTW-KNN and FastDTW-KNN algorithms respectively. The metrics of performance include accuracy, precision and recall. Table 1 shows in detail the recognition results of the three algorithms. It can be seen that the three methods can achieve 87% accuracy on the original dataset. The precision of the DTW method is slightly higher.

Table 1. Recognition performance of KNN, DTW-KNN and FastDTW-KNN on HAR dataset

Noise	KNN			DTW-KNN			FastDTW-KNN		
	acc	pre	rec	acc	pre	rec	acc	pre	rec
clean	0.87	0.88	0.87	0.87	0.89	0.87	0.87	0.89	0.87
temporal	0.70	0.73	0.71	0.87	0.89	0.87	0.87	0.89	0.87
spatial	0.80	0.84	0.80	0.68	0.77	0.67	0.68	0.77	0.67
T-S	0.61	0.68	0.61	0.71	0.81	0.71	0.70	0.80	0.71

In order to observe the performance of the three algorithms on the noise data, we drew the bar graph, as shown in Fig. 3, from which we can see the advantages of various methods clearly. KNN performs well in the original data. However, once the noise is introduced, whether it is temporal noise or spatial noise, the recognition accuracy drops rapidly. For DTW and FastDTW-KNN, the recognition accuracy keeps the same when adding temporal noise, which shows that the two methods have good temporal robustness. This is also consistent with the facts. We know that the DTW method can find the best match points between two sequences automatically. Therefore, even if the sequence is noisy in time, it does not affect the calculation of distance, and thus the recognition accuracy is also maintained. However, the DTW method and the Fast-DTW method are poor when faced with spatial noisy data. The recognition accuracy has dropped to 68% under this condition.

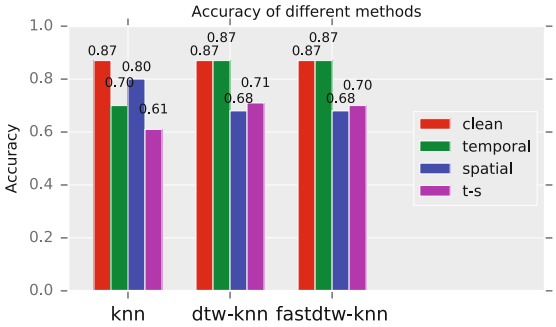

Fig. 3. Comparison of recognition accuracy on HAR dataset. "clean, temporal, spatial, T-S" stand for recognition results on different test datasets which is introduces in Fig. 1

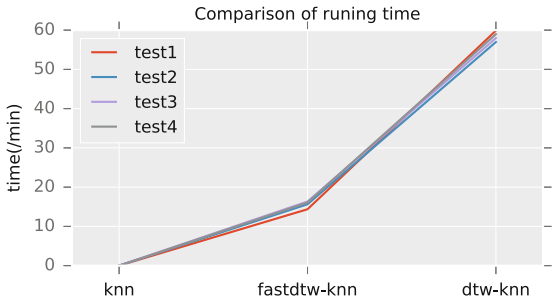

Fig. 4. Comparison of running time with four tests

In order to verify the time advantage of the Fast-DTW method compared to the DTW method, we conducted four-round tests on a computer with i5-4210 CPU, 1.7 GHz, 4 core. The test results are shown in Fig. 4. It can be seen that KNN is the fastest and FastDTW-KNN is between KNN and DTW-KNN. This shows that FastDTW-KNN not only has good temporal robustness. It's computational complexity is much lower than the traditional DTW algorithm. The larger the dataset is, the more obvious the advantage shows. But all three methods do not have a good spatial robustness. Therefore, the dynamic features are proposed to cure this drawback. The proposed feature is temporal-spatial invariant, so it can still maintain a high recognition accuracy for the signal with time and space noise. The following experimental results will prove this fact.

We compared the recognition effect with and without feature extraction. In order to prove the versatility of the extracted features, we compared different recognition methods, namely KNN, SVM and FastDTW-KNN. The experimental results are shown in the table. Table 2 shows the recognition accuracy of the three methods with extracted features. Table 3 shows the recognition accuracy of the three methods without feature extraction. Generally speaking, the accuracy is up to 98% when tested with clean data. But the volatility is very large

when noise is introduced, which drops to 34%. Recognition accuracy with the proposed dynamic feature is slightly lower, which is about 90%. But it keeps steady. Even in the worst case, we obtained 88% accuracy.

Table 2. Hand-written letters recognition based on extracted dynamic features

Algorithm	Accuracy			
	original	temporal	spatial	T-S
KNN	0.91	0.90	0.90	0.90
SVM	0.90	0.89	0.88	0.89
FastDTW-KNN	0.91	0.91	0.91	0.91

Table 3. Hand-written letters recognition without feature extraction

Algorithm	Accuracy			
	original	temporal	spatial	T-S
KNN	0.98	0.92	0.90	0.77
SVM	0.97	0.87	0.35	0.34
FastDTW-KNN	0.98	0.97	0.60	0.62

Figure 5 clearly shows the advantages of our approach. The polyline with circle markers shows the performance of the three methods in the absence of noise, which acted as a standard reference. The polyline with polygon markers and square markers represents the recognition result with and without feature extraction under the condition of temporal and spatial noise respectively. It can be seen that the recognition accuracy of our method is slightly lower than that of methods without any noise. However, the performance of our method is significantly better than others in case of noisy data. Besides, our method

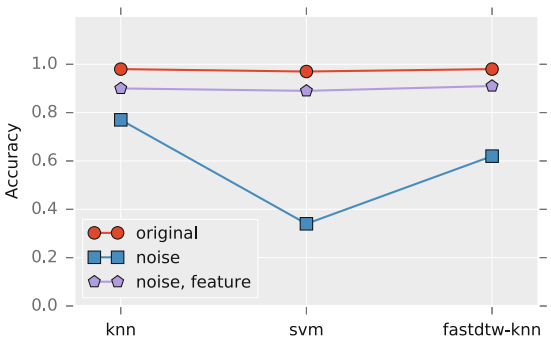

Fig. 5. Recognition performance with and without feature extraction

maintains a good consistency for the three methods. FastDTW-KNN performs slightly better, which can be attributed to the alignment of the extracted weights.

6 Conclusion and Future Work

This paper deals with the action recognition problem in the context of robot learning. We compared KNN method and DTW method. Experiments showed that the DTW method is robust to temporal noise, but computational expensive. To overcome this drawback, this paper proposed a FastDTW method to reduce the time complexity while still maintain the temporal robustness. Considering the recognition problem under spatial noise, this paper proposed a dynamic feature obtained in DMP framework. This feature is temporal-spatial invariant, so that we can maintain a high recognition accuracy even with noisy data. Besides, we found the extracted dynamic feature is prone to drift in time dimension. To fix this problem, we proposed the FastDTW-KNN algorithm, which contributes to a further improvement on the recognition accuracy. Finally, experiments were conducted on HAR dataset and handwritten letters dataset. Recognition results verified the proposed method.

Considering HMM is so popular in time sequential data analysis. Subsequent work plans to use HMM or its variants as a classifier, and the results will be compared to this paper. In addition, with the development of deep learning in recent years, especially the recursive neural network, which is very successful in dealing with sequential problems, it is also very meaningful and challenging to explore the application of RNN on action recognition.

Acknowledgments. This work is supported by National Natural Science Foundation of China (Grant Nos. 51505470).

References

1. Argall, B.D., Chernova, S., Veloso, M., Browning, B.: A survey of robot learning from demonstration. Robot. Auton. Syst. **57**(5), 469–483 (2009)
2. Huang, X.D., Ariki, Y., Jack, M.A.: Hidden Markov Models for Speech Recognition, vol. 2004. Edinburgh University Press, Edinburgh (1990)
3. Rabiner, L.R.: A tutorial on hidden Markov models and selected applications in speech recognition. Proc. IEEE **77**(2), 257–286 (1989)
4. Müller, M.: Information Retrieval for Music and Motion, vol. 2. Springer, Heidelberg (2007)
5. Poppe, R.: A survey on vision-based human action recognition. Image Vis. Comput. **28**(6), 976–990 (2010)
6. Weinland, D., Ronfard, R., Boyer, E.: A survey of vision-based methods for action representation, segmentation and recognition. Comput. Vis. Image Underst. **115**(2), 224–241 (2011)
7. Carlsson, S., Sullivan, J.: Action recognition by shape matching to key frames. In: Workshop on Models Versus Exemplars in Computer Vision, vol. 1 (2001)

8. Danafar, S., Gheissari, N.: Action recognition for surveillance applications using optic flow and SVM. In: Yagi, Y., Kang, S.B., Kweon, I.S., Zha, H. (eds.) ACCV 2007. LNCS, vol. 4844, pp. 457–466. Springer, Heidelberg (2007). doi:10.1007/978-3-540-76390-1_45
9. Ijspeert, A.J., Nakanishi, J., Schaal, S.: Learning attractor landscapes for learning motor primitives. In: Advances in Neural Information Processing Systems, pp. 1547–1554 (2003)
10. Schaal, S., Peters, J., Nakanishi, J., Ijspeert, A.: Learning movement primitives. In: Robotics Research, pp. 561–572 (2005)
11. Ijspeert, A.J., Nakanishi, J., Hoffmann, H., Pastor, P., Schaal, S.: Dynamical movement primitives: learning attractor models for motor behaviors. Neural Comput. **25**(2), 328–373 (2013)
12. Nakanishi, J., Morimoto, J., Endo, G., Cheng, G., Schaal, S., Kawato, M.: Learning from demonstration and adaptation of biped locomotion. Robot. Auton. Syst. **47**(2), 79–91 (2004)
13. Park, D.H., Hoffmann, H., Pastor, P., Schaal, S.: Movement reproduction and obstacle avoidance with dynamic movement primitives and potential fields. In: 8th IEEE-RAS International Conference on Humanoid Robots, Humanoids 2008, pp. 91–98. IEEE (2008)
14. Gams, A., Nemec, B., Ijspeert, A.J., Ude, A.: Coupling movement primitives: interaction with the environment and bimanual tasks. IEEE Trans. Rob. **30**(4), 816–830 (2014)
15. Paraschos, A., Daniel, C., Peters, J.R., Neumann, G.: Probabilistic movement primitives. In: Advances in Neural Information Processing Systems, pp. 2616–2624 (2013)
16. Schaal, S., Ijspeert, A., Billard, A.: Computational approaches to motor learning by imitation. Philos. Trans. Roy. Soc. Lond. B Biol. Sci. **358**(1431), 537–547 (2003)
17. Salvador, S., Chan, P.: Toward accurate dynamic time warping in linear time and space. Intell. Data Anal. **11**(5), 561–580 (2007)
18. Reyes-Ortiz, J.L., Anguita, D., Ghio, A., Parra, X.: Human activity recognition using smartphones data set. UCI Machine Learning Repository (2013)
19. Calinon, S.: A tutorial on task-parameterized movement learning and retrieval. Intel. Serv. Robot. **9**(1), 1–29 (2016)