Cyclic and Non-Cyclic Gesture Spotting and Classification in Real-Time Applications

Luis Unzueta and Jon Goenetxea

Vicomtech. Mikeletegi Pasealekua 57, Parque Tecnológico 20009, Donostia-San Sebastián, Spain
{lunzueta,jgoenetxea}@vicomtech.org
http://www.vicomtech.es

Abstract. This paper presents a gesture recognition method for detecting and classifying both cyclic and non-cyclic human motion patterns in real-time applications. The semantic segmentation of a constantly captured human motion data stream is a key research topic, especially if both cyclic and non-cyclic gestures are considered during the human-computer interaction. The system measures the temporal coherence of the movements being captured according to its knowledge database, and once it has a sufficient level of certainty on its observation semantics the motion pattern is labeled automatically. In this way, our recognition method is also capable of handling time-varying dynamic gestures. The effectiveness of the proposed method is demonstrated via recognition experiments with a triple-axis accelerometer and a 3D tracker used by various performers.

Key words: Human-Computer Interaction, Gesture Spotting, Gesture Recognition, Motion Pattern, Motion Capture

1 Introduction

The semantic interpretation of human motion [1] is a key research topic in various fields, such as human-computer interaction, video-surveillance, robotics, biomechanics, biometric systems, or multimedia content analysis, amongst others. Thus, gesture recognition allows us to communicate with computers at a higher level of abstraction, adding more intelligence to motion capture and computer vision systems. Moreover, combining such semantic motion information with other communication channels such as voice or touch, i.e. multimodal interfaces, would lead to a more natural interaction [2]. To achieve the goal of recognizing motion patterns, three steps must be carried out: (1) the selection of meaningful motion-features, (2) potential gesture spotting and (3) gesture classification.

The first step consists of deciding which features derived from the data being tracked will be used for a semantic interpretation. Depending on the motion capture or computer vision system, these data could be obtained directly from sensors or images, but also from the reconstruction of the user's kinematic body structure (e.g, temporal joint positions, angles, velocities, etc). These are usually chosen beforehand, but there are also some sophisticated approaches that can make this selection (semi)automatically [3, 4]. Then, motor actions are represented by *templates* [5–7] or *state-space models* [8–12] using these selected data. The former are static shape patterns containing motion information, while the latter define the considered instantaneous motion-features as a *state*, and therefore a sequence is considered as a tour going through various states.

The second step consists of segmenting the continuous data stream into temporal regions that might possibly be gestures with a meaning. As stated in [1], the main difficulties come from the segmentation ambiguity and the spatio-temporal variability involved. Additionally, gesture spotting is more challenging when both cyclic and non-cyclic gestures are considered during the interaction, because cyclic gestures may be performed with a different starting direction and number of cycles keeping the same meaning (e.g., *waving*). Hence, there are methods explicitly designed for non-cyclic gestures which require start and end pauses [11] and others for cyclic [13] which focus on motion periods.

Finally, the third step consists of labeling the segmented motion with one of the categories of the knowledge database, or as an *unknown* motion pattern. The typical classification procedures found in the literature for motor action recognition are hidden Markov models (HMMs) [8], dynamic time warping [17], nearest neighbors [5], dynamic Bayesian networks [10], neural networks [14] and kernel methods such as support vector machines (SVMs) [15] and relevance vector machines [16].

In this is work we propose a method for gesture spotting and classification that can cope with both cyclic and non-cyclic time-varying human motion patterns in real-time applications. Both objectives are achieved with a semantic observation of the performance's temporal advance, as once the computer *knows* that the user is making a certain gesture it can segment the dataflow accordingly. Unlike other approaches (especially those based on HMMs), our method does not transform motion into symbols, and allows a measure of the proximity of new performances to those in the database. This can be useful for motion style learning tasks, which can lead to motor skills transfer through imitation.

2 Cyclic and Non-Cyclic Gesture Spotting

A system designed for coping with both cyclic and non-cyclic gestures should label the observed motion patterns after each period of cyclic gestures, and after each non-cyclic gesture has been performed, even if the user keeps moving, ignoring other transition movements. Ramanan and Forsyth [15] use joint trajectories per second as motion-features, in order to obtain a continuous stream of descriptive annotations (one per frame). Their experiments reveal that in this way choppy annotation streams are produced. Therefore, they need to apply a smoothing technique, once the observed bit strings are known, obtaining automatic action descriptions quite close to real (no quantitative results are provided for comparison). Kang et al. [17], whose work is focused on videogame control, segment potential gestures by detecting abnormal velocities, frames classified as static gestures, or frames in which the tracked trajectories have severe curvatures, attaining a reliability of 93.36%. However, in this method those gestures that may include one of these events during its performance cannot be considered. Stiefmeier and Rogen [18] transform the data stream and gestures into strings encoding motion vectors and apply an approximate string matching procedure for the spotting and classification task. They achieved a correct spotting rate of 82.7% with users performing bicycle maintenance tasks including cyclic and non-cyclic gestures.

These approaches transform movements into symbol sequences before spotting and classification tasks. Symbols are obtained by clustering neighboring positions and trajectories in order to define a finite set of possibilities with which motions can be modeled. This *grid* allows a higher generality in order to label different performances of the same gesture in the same way, however at the same time it may prevent the system from measuring the proximity of different performance styles.

On the contrary, we propose to measure the spatio-temporal consistency of the data stream with respect to each of the known gestures, and once a "clear" semantic match is obtained, label the period in which this observation has been made with the corresponding meaning. Thus, the core of our approach relies on the concept *Temporal Advance Counting Algorithm* presented by Mena et al. [12], but goes beyond it by analyzing the advance through a dynamic time buffer which is increased until the decision is taken, instead of observing a constant number of recent frames for labeling the most recent one at each time instant. In this paper we focus on the recognition of gestures performed by a single "rigid" body (e.g., one hand, the head, etc). The combination of semantic body part motion descriptions in a multibody structure (i.e., a full human body) is beyond this scope.

The motion of a body part is defined as a temporally ordered sequence of motion-features, i.e. vectors containing relevant information for further gesture classification (e.g., velocities, accelerations, angular variations, etc). Therefore, the knowledge database is constituted by a set of labeled motion patterns represented as connected states. The number of states will be the same for all of them in order to make a balanced computation of the *temporal advance* in all gesture candidates. Hence, even though this normalization is obtained through a post-processing step (concretely adjusting a cubic-spline), the number of states of the original gestures should not be too different from each other, so that they do not get too distorted. This may appear to be a major restriction on the kind of actions that can be modeled together (even after the cubic spline fitting), but the complexity of these can be higher than those presented in previous approaches in the field [15, 17, 18]. However, there is a restriction that must be accomplished and it is that gestures must be independent one of each other, i.e. there must not be gestures whose complete shape is similar to the part of another.

Algorithm 1 shows how the temporal advance is computed for a motion sequence of size p with respect to a gesture candidate C. This advance takes into consideration the proximity of the recent dataflow states with respect to those of the gesture. Hence, we call it a *weighted temporal advance*, where the weight comes from the inverse of the mean distance of advancing states with respect to their corresponding nearest ones in the gesture. A higher weighted temporal advance count means a more accurate approximation to the gesture candidate, and thus can be used as a quantitative measure for motor skills transfer through imitation. However, it must be taken into account that in order to avoid a division by zero this proximity must be limited to a certain minimal value. Note that multiple states in the observed motion sequence can get matched to the same state in gesture model, which would mean that there would not be advance in that case, but this feature is precisely the one that allows to handle time warping in performed gestures.

Algorithm 1 Weighted Temporal Advance Algorithm

```
procedure WeightedTemporalAdvance($sequence, gesture_C$)
    $nVotes_C \Leftarrow 0$
    $nearestStateIndex_C \Leftarrow -1$
    $previousIndex_C \Leftarrow -1$
    $sumDistances_C \Leftarrow 1$
    $nearestStateDistance_C \Leftarrow 0$
    for $i = 1$ to $p$ do
        $nearestStateIndex_C \Leftarrow$getNearestStateIndex($sequence_{[i]}$)
        $nearestStateDistance_C \Leftarrow$getNearestStateDistance($sequence_{[i]}$)
        if $nearestStateIndex_C > previousIndex_C$ then
            $nVotes_C \Leftarrow nVotes_C + 1$
            $sumDistances_C \Leftarrow sumDistances_C + nearestStateDistance_C$
        end if
        $previousIndex_C \Leftarrow nearestPoseIndex_C$
    end for
    return $nVotes_C/(sumDistances_C/nVotes_C) = nVotes_C^2/sumDistances_C$,
  where $sumDistances_C > 0$
end procedure
```

3 Semantic Observation of Temporal Advance

The weighted temporal advance will allow to compute the level of confidence in the continuous data stream for a semantic gesture spotting. Firstly, we spot when occurs a variation in the state sequence higher than a certain threshold, and start the observation from the instant in which that variation was zero. To do so we apply the algorithm used in [11] for the starting point determination. Having this threshold allows us to filter small state variations due to noise. Then, we can start the semantic observation from that point until the system takes a decision, which could be a gesture detection or doing a reset. Therefore, the observed segment, i.e. the buffer, increases its size dynamically as new motion-features are being obtained from the motion capture system. Hence, gesture spotting and classification are solved in parallel. There are six conditions that the observation must accomplish so that a data stream segment is labeled with a gesture candidate:

(a) It has the highest weighted temporal advance.
(b) The weighted temporal advance is over a threshold.
(c) The number of temporal advances without the distance weight is at least a certain portion of the number of gesture states.
(d) The observed data stream has at least a certain number of states.
(e) The dataflow has not been still for at least a certain time.
(f) The number of frames in the buffer is not excessive.

If all these conditions are met, apart from labeling the segment, the system also resets the weighted temporal advance counting and *forgets* the previous data, which means that in case a cyclic gesture is being done, when the system tries to detect again the starting point of the new cycle, the lastest instant that it may take into consideration will be the latest of the previous segment. On the contrary, if the system accomplishes conditions (c) and (d), but not (b), or it does not satisfy conditions (e) or (f), the counting is reset, but no answer is delivered, because there was not enough confidence on the best candidate. Meanwhile, while these situations are not met algorithm 1 is applied to the increasing buffer. The matching procedure is not sensitive to the starting location, which is of special interest especially for cyclic actions, which can start at any state, because the weighted temporal advance will be increased independently of it.

This algorithm is fast enough for human-computer interaction with off-the-shelf equipment, but in case it would be necessary, it may also be possible to alleviate the computational cost by applying the counting every N frames while the buffer is increasing and not every frame. Alternatively, taking advantage of current GPU and multi-core CPU platforms, it is also possible to parallelize the measurements with respect to gesture candidates, to attain faster framerates, or otherwise for increasing the database size with a higher number of candidates.

4 Experimental results

In order to evaluate the presented gesture spotting and classification method, a set of continuous dataflow captures containing a series of hand gestures performed several times is used. The number of correct spotting and classifications are computed, but also the number of *deletions*, *insertions* and *substitutions*. Deletions occur when a gesture has not been spotted, insertions when the system has spotted a gesture when it should not, and substitutions when it has spotted a gesture correctly but it has not classified it with the right label. We build the continuous data streams by concatenating previously segmented gestures so that the obtained results can be visualized in an easier way (otherwise the continuous dataflows should be segmented manually afterwards). In this way we exactly know when start and end real gestures and which they are. There may appear unnatural discontinuities at the boundaries of actions, especially for non-cyclic actions, but these are not relevant for this test because, as stated in Section 3, the weighted temporal advance will be increased independently of the gesture starting point.

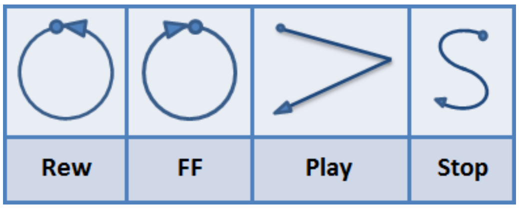

Fig. 1. The dynamic gestures to be performed

Both a triple-axis accelerometer (Wiimote: `http://www.nintendo.com/wii`) and a 3D tracker (Flock of Birds: `http://www.ascension-tech.com`) are used for the experiment with the same gestures in order to compare results with different motion-features. The motion-features used in the triple-axis accelerometer are directly the data coming from the sensor, while in the case of the 3D tracker the velocity vectors derived from captured 3D positions are used. Fig. 1 shows the gesture classes to be performed in the experiment. For each device, four users perform 20 times these four gestures and therefore there are $20 \times 4 \times 4 = 320$ samples in total (80 repetitions per gesture). The two confusion matrices obtained from the leave-one-out training validation method [19] with all these samples are shown in table 1. It can be seen that gesture classification using the weighted temporal advance algorithm obtains very high rates: 98.75% using the triple-axis accelerometer and 100% with the 3D tracker.

Assigned	**Real Class (3-Axis Accel.)**			
Class	**Rew**	**FF**	**Play**	**Stop**
Rew	**80**	0	**2**	0
FF	0	**80**	0	0
Play	0	0	**78**	**2**
Stop	0	0	0	**78**

Assigned	**Real Class (3D Tracker)**			
Class	**Rew**	**FF**	**Play**	**Stop**
Rew	**80**	0	**0**	0
FF	0	**80**	0	0
Play	0	0	**80**	**0**
Stop	0	0	0	**80**

Table 1. Confusion matrices of the labeled gestures captured with the triple-axis accelerometer and the 3D tracker respectively, using leave-one-out

For the dataflow automatic segmentation, for each device, a part of the recorded samples is used to build the knowledge database and the rest to build the continuous data streams to be evaluated, one for each performer. During the database training it is possible to obtain the most suitable parameter values for gesture spotting according to it. These parameters are: (a) normalized number of states per gesture in the database (NNS), (b) weighted temporal advance threshold (WTAT) and (c) temporal advance number with respect to the number of states (TANS). In order to obtain the optimal parameter values, a continuous dataflow with the database gestures (without resampling) is evaluated with different parameters combinations until the one with the highest recognition rate is obtained. In our experiments we obtain, with slight variations from case to case, NNS=12, WTAT=10 and TANS=70% for the triple-axis accelerometer and NNS=19, WTAT=5 and TANS=70% for the 3D tracker. On the other hand, the threshold of sequence variation for determining the observation starting point is set manually for each device through experimentation, so that slight movements are filtered. In this experiment we test two different alternatives for evaluating the system: (1) using only one database of 80 samples (5 performances per gesture and user) to evaluate the continuous dataflows of all users with the same gesture spotting parameter values and (2) using 4 databases of 20 samples (one per user, 5 performances per gesture) to evaluate the continuous dataflows of the corresponding users that trained the system.

Case 1	**Correct**	**Deleted**	**Inserted**	**Substituted**	**Ground Truth**
Subject 1	58 (96.67%)	1 (1.67%)	3 (5%)	1 (1.67%)	60
Subject 2	57 (95%)	3 (5%)	6 (10%)	0 (0%)	60
Subject 3	53 (88.33%)	3 (5%)	13 (21.67%)	4 (6.67%)	60
Subject 4	51 (85%)	3 (5%)	16 (26.67%)	6 (10%)	60
Total	**219 (91.25%)**	**10 (4.16%)**	**38 (15.83%)**	**11 (4.58%)**	**240**

Case 2	**Correct**	**Deleted**	**Inserted**	**Substituted**	**Ground Truth**
Subject 1	60 (100%)	0 (0%)	6 (6.67%)	0 (0%)	60
Subject 2	56 (93.33%)	3 (5%)	12 (20%)	1 (1.67%)	60
Subject 3	56 (93.33%)	2 (3.33%)	6 (10%)	2 (3.33%)	60
Subject 4	50 (83.33%)	8 (13.33%)	13 (21.67%)	2 (3.33%)	60
Total	**222 (92.25%)**	**13 (5.41%)**	**35 (14.58%)**	**5 (2.08%)**	**240**

Table 2. Spotting and classification results with the triple-axis accelerometer for different subjects using (1) an overall auto-generated configuration and (2) their own databases and auto-generated configurations respectively

Table 2 shows the obtained spotting and recognition results of this test using the triple-axis accelerometer. It can be seen that in both cases remarkable recognition rates are obtained (above 91%), and also that using the overall database of 80 samples a slightly lower rate (91.25%) than using smaller (20 samples) but more user oriented ones (92.25%) is achieved. Table 3 shows the obtained results with the 3D tracker and the user oriented databases (we omit the overall database results because similar conclusions to those with the tripe axis accelerometer are deduced). In this case the obtained results are even better (94.58%). This improvement is also related to the employed motion-features. In this case, these have a more direct relation with the performed movements, while in the case of the triple-axis accelerometer the captured data are influenced by gravity apart from the movements themselves. Regarding the computation time, the heaviest system, i.e. the one using the 80 sample database, runs at 82-98 Hz which is above real-time performance even if the implementation has not been parallelized. The system was implemented using C++, and tested on a 2.00 GHz Intel Celeron 1 GB RAM.

	Correct	Deleted	Inserted	Substituted	Ground Truth
Subject 1	58 (96.67%)	2 (3.33%)	5 (8.33%)	0 (0%)	60
Subject 2	58 (96.67%)	2 (3.33%)	9 (15%)	0 (0%)	60
Subject 3	56 (93.33%)	3 (5%)	5 (8.33%)	1 (1.67%)	60
Subject 4	55 (91.67%)	4 (6.67%)	4 (6.67%)	1 (1.67%)	60
Total	**227 (94.58%)**	**11 (4.58%)**	**23 (9.58%)**	**2 (0.83%)**	**240**

Table 3. Spotting and classification results with the 3D tracker for different subjects using their own databases and auto-generated configurations

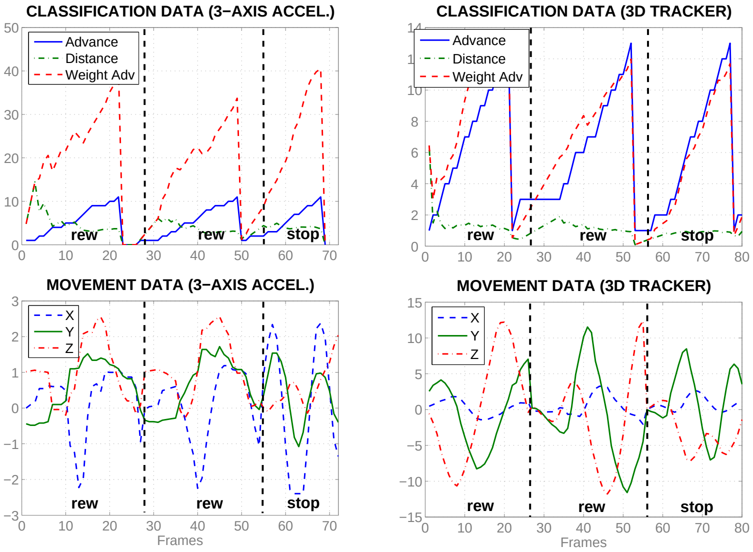

Fig. 2. Close-up of the semantic gesture spotting and classification using the triple-axis accelerometer and the 3D tracker respectively

Finally, Fig. 2 shows close-ups of how the semantic observation of the temporal advance segments the data stream with respect to the true start and end points of gestures being performed one after the other for both motion capture devices. It can be seen how both the temporal advance and the weighted temporal advance increase their values while the gestures are being recognized and how the system resets to zero once it has met the necessary conditions to take a decision. It can also be observed how the decision is taken a few frames before the real transition from gesture to gesture (marked with vertical dashed lines). It occurs this way because it has been determined during the training that the answer should be given when the number of temporal advances without the weight is a bit less than the total number of states per gesture in the database, in order to obtain better recognition rates.

5 Conclusions and Further Work

In this is work we have presented a method for gesture spotting and classification that can cope with both cyclic and non-cyclic time-varying human motion patterns in real-time applications. The spatio-temporal consistency of the data stream with respect to each of the known gestures is measured with a weighted temporal advance counting, where the weight comes from the inverse of the mean distance of advancing states with respect to their corresponding nearest ones in the gesture. A higher weighted temporal advance count means a more accurate approximation to the gesture candidate, and thus can be used as a quantitative measure for motor skills transfer through imitation. This weighted temporal advance allows to compute the level of confidence in the continuous data stream for a semantic gesture spotting.

The semantic observation starts from the instant when a variation in the state sequence higher than a certain threshold occurs until the system takes a decision, which could be a gesture detection or doing a reset, depending on certain conditions related with the temporal advance, the segment size and the state sequence variation. Hence, gesture spotting and classification are solved in parallel. Experimental results with gestures performed by various users with a triple-axis accelerometer and a 3D tracker show the potential of this approach for human-computer interaction.

Future work will focus on automatizing the selection of the optimal motion-features for the spotting and recognition of gestures involving different body parts. Additionally, it will also be explored the combination of semantic body part motion descriptions in a multibody structure, extending the work done in this subject in previous approaches such as [15, 20].

References

1. Mitra, S., Acharya, T.: Gesture Recognition: A Survey. In: IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 37(3), 311–324 (2007)
2. Jaimes, A., Sebe, N.: Multimodal Human Computer Interaction: A Survey. In: Computer Vision and Image Understanding, 108(1-2), 116–134 (2007)
3. Lösch, M., Schmidt-Rohr, S.R., Knoop, S., Dillmann, R.: Feature Selection for Human Activity Recognition Using Feature Taxonomies and User Comments. In: Proceedings of the International Conference on Cognitive Systems, Karlsruhe, Germany (2008)

4. Liu, G., Zhang, J., Wang W. and McMillan L.: Human Motion Estimation from a Reduced Marker Set. In: Proceedings of the 2006 Symposium on Interactive 3D Graphics and Games. Boston, MA, USA (2006)
5. Masoud, O., Papanikolopoulos, N.: A Method for Human Action Recognition. In: Image and Vision Computing, 21(8), 729–743 (2003)
6. Weinland, D., Ronfard, R., Boyer, E.: Motion History Volumes for Free Viewpoint Action Recognition. In: Proceedings of the Workshop on Modeling People and Human Interaction, Beijing, China, 104, 249–257 (2005)
7. Rahman, M. M., Robles-Kelly, A.: A Tuned Eigenspace Technique for Articulated Motion Recognition. In: Proceedings of the European Conference on Computer Vision, LNCS 3954, Graz, Austria, 174–185 (2006)
8. Ahmad, M., Lee, S.-W.: Human Action Recognition Using Multi-View Image Sequences Features. In: Proceedings of the International Conference on Automatic Face and Gesture Recognition, Southampton, UK, 523–528 (2006)
9. Rittscher, J., Blake, A., Roberts, S. J.: Towards the Automatic Analysis of Complex Human Body Motions. In: Image and Vision Computing, 20, 905–916 (2002)
10. Ren, H., Xu, G., Kee, S.: Subject-Independent Natural Action Recognition. In: Proceedings of the International Conference on Automatic Face and Gesture Recognition, Seoul, Korea, 523–528 (2004)
11. Unzueta, L., Mena, O., Sierra, B., Suescun, Á.: Kinetic Pseudo-Energy History for Human Dynamic Gestures Recognition. In: Proceedings of the Conference on Articulated Motion and Deformable Objects, LNCS 5098. Pto. Andratx, Mallorca, Spain, 390–399 (2008)
12. Mena, O., Unzueta, L., Sierra, B., Matey, L.: Temporal Nearest End-Effectors for Real-Time Full-Body Human Actions Recognition. In: Proceedings of the Conference on Articulated Motion and Deformable Objects, LNCS 5098. Pto. Andratx, Mallorca, Spain, 269–278 (2008)
13. Kubota, N., Abe, M.: Computational Intelligence for Cyclic Gestures Recognition of A Partner Robot. In: Proceedings of the International Conference on Knowledge-Based Intelligent Information & Engineering Systems, Melbourne, Australia, 650–656 (2005)
14. Yu, H., Sun, G.-m., Song, W.-x., Li, X.: Human Motion Recognition Based on Neural Network. In: Proceedings of the International Conference on Communications, Circuits and Systems, Hong Kong, China, 2, 982 (2005)
15. Ramanan, D., Forsyth, D. A.: Automatic Annotation of Everyday Movements. In: Proceedings of the Neural Information Processing Systems Conference, Vancouver, BC, Canada (2003)
16. Guo, F., Qian, G.: Dance Posture Recognition Using Wide-Baseline Orthogonal Stereo Cameras. In: Proceedings of the International Conference on Automatic Face and Gesture Recognition, Tempe, AZ, USA, 481–486 (2006)
17. Kang, H., Lee, C.W., Jung, K.: Recognition-Based Gesture Spotting in Video Games. In: Pattern Recognition Letters, 25(15), 1701–1714 (2004)
18. Stiefmeier, T. Roggen, D.: Gestures Are Strings: Efficient Online Gesture Spotting and Classification Using String Matching. In: Proceedings of the International Conference on Body Area Networks, Florence, Italy (2007)
19. Stone, M.: Cross-Validation Choice and Assessment of Statistical Procedures. Journal of Royal Statistical Society, 36, 111–147 (1974)
20. Unzueta, L.: Markerless Full-Body Human Motion Capture and Combined Motor Action Recognition for Human-Computer Interaction. PhD Thesis, Tecnun, University of Navarra, Donostia-San Sebastián, Spain (2009)