A Real Time Gesture Recognition System for Human Computer Interaction

Carmela Attolico[1], Grazia Cicirelli[2], Cataldo Guaragnella[1], and Tiziana D'Orazio[2(✉)]

[1] DEI - Politecnico di Bari, via Orabona 7, 70126 Bari, Italy
[2] Institute of Intelligent Systems for Automation - CNR, via Amendola 122/D-I, 70126 Bari, Italy
dorazio@ba.issia.cnr.it

Abstract. Every form of human gesture has been recognized in the literature as a means of providing natural and intuitive ways to interact with computers across many computer application domains. In this paper we propose a real time gesture recognition approach which uses a depth sensor to extract the initial human skeleton. Then, robust and significant features have been compared and the most unrelated and representative features have been selected and fed to a set of supervised classifiers trained to recognize different gestures. Different problems concerning the gesture initialization, segmentation, and normalization have been considered. Several experiments have demonstrated that the proposed approach works effectively in real time applications.

1 Introduction

Recognition of human gesture from video sequences is a popular task in the computer vision community since it has wide applications including, among others, human computer interface, video surveillance and monitoring, augmented reality, and so on. In the last decade, the use of color cameras made this one a challenging problem due to the complex interpretation of real-life scenarios such as multiple people in the scene, cluttered background, occlusion, illumination and scale variations and so on [7,15]. Many papers presented in literature have mostly been concerned with the problem of extracting visual features and combine them in space and time for making a decision on what actions are present in the video. The promising results were obtained using, for both training and testing, action recognition databases containing segmented video clips each showing single person performing actions from start to finish [11,12]. The recent availability of depth sensors has provided a new impetus to this research field, avoiding many of the problems described above and allowing applications in real time contexts. In particular, inexpensive Kinect sensors have been largely used by the scientific community as they provide an RGB image and a depth of each pixel in the scene. Open source frameworks, such as OpenNI, are available to process depth sensory data and allow the achievement of complex tasks such as people segmentation, real time tracking of a human skeleton, scene information, and so

F. Schwenker et al. (Eds.): MPRSS 2014, LNAI 8869, pp. 92–101, 2015.
DOI: 10.1007/978-3-319-14899-1_9

on [8]. The direct availability of real-time information about joint coordinates and orientations has provided a great impetus to research and many papers on gesture recognition have been published in the last years.

The hand orientation and four hand gestures (open, fist,...) are used in [3] for a gesture recognition system integrated on an interactive robot which looks for a person to interact with, ask for directions, and detects a 3D pointing direction. In [13], the 3D hand trajectories of a Graffiti gesture set are assigned to a binary decision tree for a coarse classification and then provided to a MultiClass SVM for the final decision. The motion profiles obtained from the Kinect depth data are used in [5] to recognize different gestures by a multi class SVM. The motion information is extracted by noting the variation in depth between each pair of consecutive frames. Aircraft marshaling gestures, used in the military air force, are recognized in [4] considering seven upper body joints. The method requires the data stream editing by a human observed who marks the starting and ending frame of each gesture. The nodes of the skeleton, in [16], are converted in joint angle representation that provides invariance to sensor orientation. Then a multiclass SVM is used to classify key poses which are forwarded as a sequence to a decision forest for the final gesture recognition. Also in [9] joint angles are considered for the recognition of six different gestures but different HMMs have been used to model the gestures. The HMM which provides the maximum likelihood gives the type of gesture. Four joint coordinates relative to the left and right hands and elbows are considered in [14]. The normalized distances among these joints form the feature vector which is used in a nearest neighbor classification approach. A rule based approach is used in [10] to recognize key postures and body gestures by using an intuitive reasoning module which performs forward chaining reasoning (like a classic expert system) with its inference engine every time new portion of data arrives from the feature extraction library.

In this paper we propose a gesture recognition approach which uses as significant features, the quaternions, of some skeleton joints provided by the Kinect sensor and a supervised approach to build the gesture models. We focus our attention on some issues related to the use of these methodologies for real time applications: the detection of the initial frame of each gesture, the normalization of the length of different gestures, the ability of the system to avoid false positives when the user is not involved in any gesture. The gesture segmentation is fundamental for the success of every gesture recognition method and it has to be solved before recognition to make the model matching more efficient. In this paper we propose a periodicity analysis to extract the gesture length and to normalize the test sequences in order to have sequences comparable with the generated models. In addition, in order to be independent from the starting frame of the sequence, we propose the use of a sliding window and a consensus analysis to make a decision on the recognized gesture. Real time experiments demonstrate the applicability of the proposed approach both in terms of computational load and in terms of detection performances.

The rest of the paper is organized as follows: Sect. 2 describes the gesture model generation phase, Sect. 3 the off-line and on-line tests, and finally Sect. 4 reports discussion and conclusions.

2 Model Generation Phase

In this work we propose a Gesture Recognition approach which can be used by a human operator as a natural human computer interface. We use the abilities of the Kinect framework to identify and track people in the environment and to extract the skeleton with the joint coordinates for the gesture recognition. We identify ten different gestures executed with the right arm (see Fig. 1). Frame sequences of gestures executed several times by only one person are considered for the gesture model generation (training phase). Different sequences of gestures executed by other people are used for the test phase, instead.

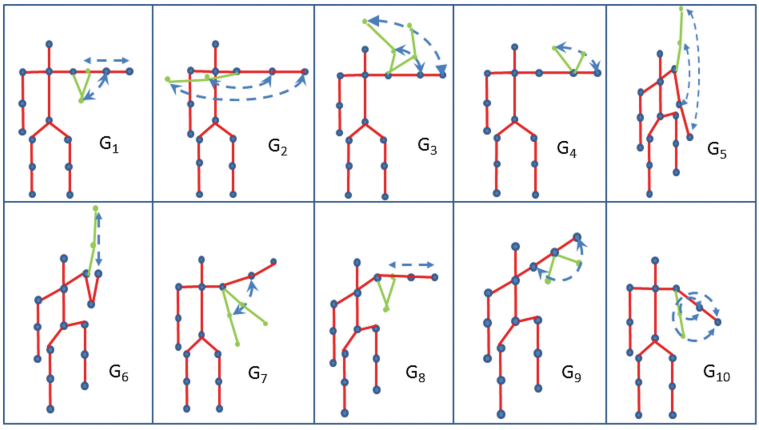

Fig. 1. Ten different gestures selected from the army visual signals report [2] are shown. Gestures G_5, G_6, G_8 and G_9 are pictured in a perspective view as the arm has a forward motion. In gestures G_1, G_2, G_3, G_4, G_7 and G_{10} the arm has lateral motion instead, so the front view is drawn.

2.1 Feature Selection

The problem of selecting significant features which preserve relevant information to the classification, is fundamental for the ability of the method to recognize gestures. Many papers of the literature consider the coordinate variations of some joints such as the hand, the elbow, the shoulder and the torso nodes [6,17]. However, when coordinates are considered, it is necessary to introduce a kind of normalization in order to be independent of the position and the height of the person in the scene. An alternative way could be the use of angles among joint nodes [1,9], but the angle representation is not exhaustive to describe rotation

in 3D space as the axis of rotation has to be specified to represent a 3D rotation. After a comparative evaluation of all the feature set provided by the Kinect framework, we selected the Quaternions of some joint nodes. A Quaternion is a set of numbers that comprises a four-dimensional vector space and is denoted by $q = a+bi+cj+dk$, where a, b, c, d are real numbers and i, j, k are imaginary units. The quaternion q represents an easy way to code any 3D rotation expressed as a combination of a rotation angle and a rotation axis. Quaternions offer fundamental computational implementation and data handling advantages over the conventional rotation matrices. Quaternions provide a straight forward way of representing rotations in a three-dimensional space. Considering the defined gestures, the quaternions of the arm and shoulder joint nodes maintain the information about the direction the relative bone is pointing to. For this reason, the quaternions of the right shoulder and elbow nodes have been selected as features. As a consequence, for each frame i an eight-dimensional feature vector has been defined:

$$V_i = [a_i^s, b_i^s, c_i^s, d_i^s, a_i^e, b_i^e, c_i^e, d_i^e]$$

where superscripts s and e stands for shoulder and elbow respectively.

2.2 Feature Normalization

The execution of the gestures by different people or by the same person can greatly vary. Different velocities can be used by each user when executing gestures. So the length of each gesture execution, in terms of number of frames, can be variable also for the same gesture. For this reason, the frame sequence containing each gesture, has been normalized in both phases: training and testing.

During the training phase, one single person was asked to repeat each gesture for several times and with a 2-s pause among the executions. The sequences relative to gesture execution were extracted and normalized to the same length (60 frames) by using a Spherical Linear Interpolation (SLERP) [18]. SLERP provides a simple and elegant interpolation between points on a hypersphere. A quaternion, indeed, describes a hypersphere, i.e. a four-dimensional sphere with a three-dimensional surface. If q_1 and q_2 are two quaternions and t is a parameter moving from 0 to 1, a reasonable geometric condition to impose is that q_t lies on the hyperspherical arc connecting q_1 and q_2. The formula for obtaining this, is given by:

$$Slerp(q_1, q_2; t) = \frac{\sin(1-t)\theta}{\sin\theta} q_1 + \frac{\sin t\theta}{\sin\theta} q_2$$

where $q_1 \cdot q_2 = \cos\theta$. So, by using this $Slerp$ equation the sequences of frames containing one gesture execution can be normalized to the same length (*sub-sampling* and *over-sampling*).

2.3 Gesture Length Estimation

During the testing phase, different people were asked to repeat the gestures without interruption and all the frames of the sequences were recorded. As already

mentioned each people execute gestures by using different velocities, so the length of the sequence containing one gesture execution cannot be *a priori* known. Therefore a gesture length estimation method must be applied in order to determine when the gesture starts and ends. An algorithm based on Fast Fourier Transform (FFT) has been applied in order to automatically evaluate the time period of each gesture execution. In particular, sequences of 300 frames have been considered for the period evaluation. By tacking the position of the fundamental harmonic component, the period is evaluated as the reciprocal value of the peak position. In Fig. 2(a) and (b) the value a of the elbow joint quaternion of one gesture is pictured, whereas Fig. 2(c) and (d) show their respective FFT. As can be seen in Fig. 2(a) and (b) two different velocities have been used for the gesture execution. If k is the peak position in the frequency sampled domain of the FFT representation, the period estimation can be given considering that:

$$\frac{k}{N} = \frac{f_0}{f_c} = \frac{T_c}{T_0}$$

where N is the number of samples of the frequency domain of the transformed time series, f_c the used sampling frequency, f_0 the fundamental frequency component and T_0 and T_c the time periods of the corresponding frequencies. As $T_0 = n \times T_c$ where n is the number of samples of the gesture duration, the period of the gesture can be simply computed by $n = N/k$. So, $n = 50$ for the example represented in Fig. 2(a); whereas $n = 27$ for the case shown in Fig. 2(b).

2.4 Neural Network Training

The models for the gesture recognition are constructed by using ten different supervised Neural Networks (NN), one for each gesture. Each training set is built considering a set of feature sequences of one gesture as positive examples and the remaining sequences of all the other gestures as negative examples. Each NN has an input layer of 480 nodes corresponding to the eight-dimensional feature vectors of 60 frames, an hidden layer of 100 nodes and an output layer where there is one node returning a 1 if the gesture is recognized, zero otherwise. A Backpropagation algorithm has been used for training and the best configuration of hidden nodes and network parameters was selected in an heuristic way after several experiments. At the end of the learning phase, in order to recognize a gesture, a sequence of features is fed to all the 10 NNs and the one which returns the maximum answer wins providing the recognized gesture. However notice that this procedure always gives a result, even if a gesture does not belong to any of the ten classes. For this reason a threshold has been introduced in order to decide if the maximum answer must be considered reliable or not. In the case the maximum answer is under the threshold (fixed equal to 0.7) the gesture is classified as a No-Gesture (NG).

3 Experiments

Two different sets of experiments have been carried out: (1) off-line experiments in order to test the gesture recognition algorithm on recorded sequences of

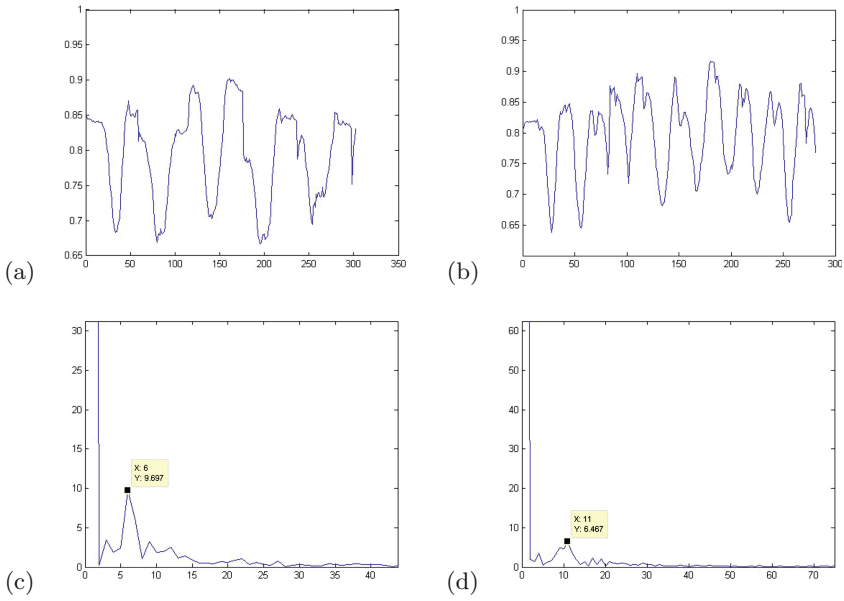

Fig. 2. (a, b): plot of the a component of the elbow joint quaternion of one gesture executed at two different velocities; (c, d): the respective FFT.

gestures executed by different people included and not included in the training set; (2) on-line experiments in order to test the ability of the proposed algorithm in real-time during the on-line acquisition of frames by the Kinect sensor. This step is fundamental to test the gesture segmentation approach presented in Sect. 2.3 and allows us to use the proposed system in real situations and not only on stored databases.

3.1 Off-Line Experiments

The proposed algorithm has been tested using a database of 10 gestures performed by 10 different people. A selection of sequences of gestures executed by only one person have been used to train the NNs, while all the remaining sequences of gestures performed by all the other people have been used for the test.

In Table 1 the percentages of gesture recognitions are reported. This case refers to the tests carried out considering only one person for gesture executions, the same one whose gestures were used for training the NNs. As can be seen, the majority of gestures are 100 % correctly recognized, only gestures G_7 and G_{10} have lower recognition percentages. In Table 2, instead, the recognition percentages of gesture executed by people not considered for the training set are listed. Also in this case some erroneous results occur. Sometimes False Positives, as for G_{10} in Table 2, are due to the complexity and similarity of some

Table 1. Percentages of gesture recognitions: the test set contains instances of gestures executed by the same person considered for the training set.

%	G_1	G_2	G_3	G_4	G_5	G_6	G_7	G_8	G_9	G_{10}	NG
G_1	**100**	0	0	0	0	0	0	0	0	0	**0**
G_2	0	**100**	0	0	0	0	0	0	0	0	**0**
G_3	0	0	**100**	0	0	0	0	0	0	0	**0**
G_4	0	0	0	**100**	0	0	0	0	0	0	**0**
G_5	0	0	0	0	**100**	0	0	0	0	0	**0**
G_6	0	0	0	0	0	**100**	0	0	0	0	0
G_7	0	0	0	0	0	0	**89**	0	0	0	**11**
G_8	0	0	0	0	0	0	0	**100**	0	0	**0**
G_9	0	0	0	0	0	0	0	0	**100**	0	0
G_{10}	0	0	0	0	0	0	0	0	0	**85**	**15**

Table 2. Percentages of gesture recognitions: the test set contains instances of gestures executed by people different from the one considered for the training set.

%	G_1	G_2	G_3	G_4	G_5	G_6	G_7	G_8	G_9	G_{10}	NG
G_1	**100**	0	0	0	0	0	0	0	0	0	**0**
G_2	0	**90**	0	0	0	0	0	0	0	0	**10**
G_3	0	0	**100**	0	0	0	0	0	0	0	**0**
G_4	0	0	0	**97**	0	0	0	0	0	0	**3**
G_5	0	0	0	0	**63**	0	0	0	0	0	**37**
G_6	0	0	0	0	0	**96**	0	0	0	0	**4**
G_7	0	0	0	0	0	0	**100**	0	0	0	**0**
G_8	0	0	0	0	0	0	0	**100**	0	0	**0**
G_9	0	0	0	0	0	0	0	0	**100**	0	0
G_{10}	0	**7**	**14**	0	0	0	0	0	0	**68**	**11**

gestures executions that could be ambiguously recognized by the NNs. But for both cases (of Tables 1 and 2) the fundamental problem is due to the instability of joints detected by the Kinect framework. Indeed gesture G_{10}, for example, is performed with the arm direction perpendicular to the camera and in this case, experimental evidence demonstrates that the skeleton and then the joints are not correctly detected as the arm is not completely visible. This is also the case of gestures G_2, G_5 and G_6 which involve arm movements perpendicularly to the camera.

3.2 On-Line Experiments

During on-line experiments, the system has been stressed to recognize the gestures executed at different velocities and by different people. As the initial frame or the length of each gesture are not known, the user in front of the Kinect, was asked to perform repeatedly the gesture. As introduced in Sect. 2.3 and FFT-based approach has been applied to evaluate the period n of the performed gesture. In addition two further steps are introduced as reported in Fig. 3: an initial sliding window approach and a final gesture recognition step by consensus evaluation. The video sequences containing more repetitions of the same gesture are divided into multiple overlapping segments of n frames. Then, these segments are re-sampled by using the SLERP procedure and are fed to all the 10 NNs. A consensus decision making is applied to sequences of 300 frames: the number of consecutive concordant answers of the same NN is counted and if this number exceeds a fixed threshold (=10), an additional 11-dimensional vector of counters is used to register the gesture class (G_1 or G_2, ..., G_{10}, or NG). The gesture class with the maximum counter wins, so the corresponding gesture is recognized.

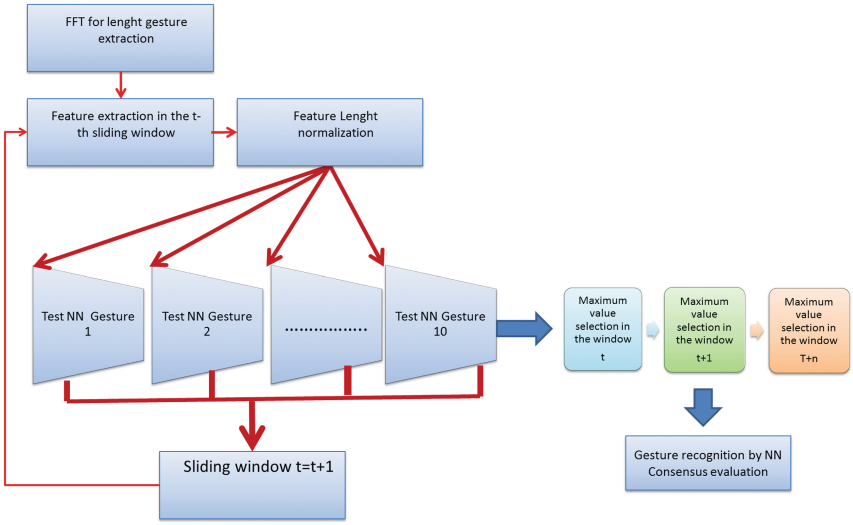

Fig. 3. The proposed approach for gesture recognition.

Figure 4 shows the results obtained when gestures are performed continuously but at different velocities. In particular they refer to the executions of gesture G_9 first slowly ($n = 64$), then faster ($n = 36$) and finally very slowly ($n = 99$). Thanks to the FFT-based algorithm capable of extracting the different periods, the system is however able to correctly recognize the gesture as shown in Fig. 4. Some wrong

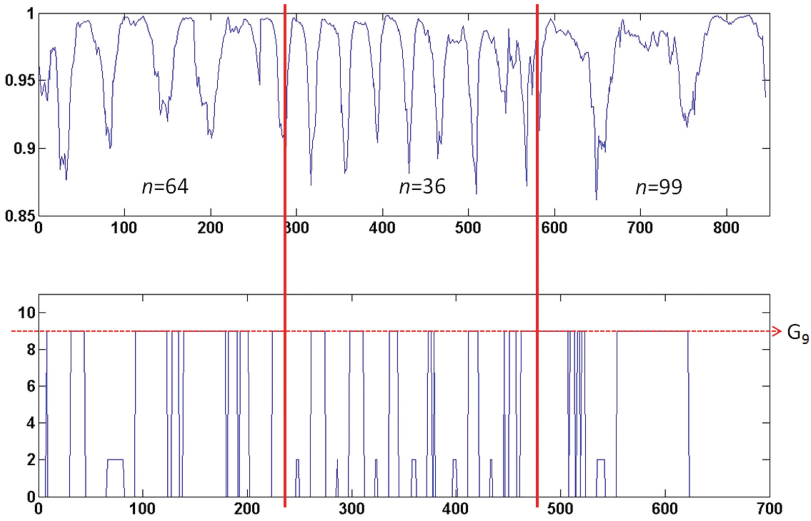

Fig. 4. Recognition results when gesture G_9 is executed at different velocities. Figure at the top plots the a feature component showing the different periodicity of the signal. Figure at the bottom shows the NN answers, gesture G_9 is recognized.

results occur, but these do not affect the final decision of the consensus based procedure which, as mentioned above, is based on a sliding window approach and on the number of consecutive concordant answers of the same neural network.

4 Discussion and Conclusions

In this paper we propose a gesture recognition system using a Kinect sensor which provides the people segmentation in an effective way and skeleton information for real time processing. We use the quaternion features of the right shoulder and elbow nodes to construct the models of 10 different gestures. In this paper we consider some problems related to the application of a gesture recognition system to real time experiments. These are the lack of knowledge of the initial frame and the length of the gesture and the ability of the algorithm to avoid false detections when the user is not involved in any gesture. The obtained results are very encouraging as the number of false positives is very small.

References

1. Almetwally, I., Mallem, M.: Real-time tele-operation and tele-walking of humanoid robot Nao using Kinect depth camera. In: Proceedings of 10th IEEE International Conference on Networking, Sensing and Control (ICNSC), pp. 463–466 (2013)
2. Army: Visual signals: Arm-and-hand signals for ground forces. Field Manual 21–60, Washington, DC (September 1987), Headquarter Department of the Army

3. den Bergh, M.V., Carton, D., de Nijs, R., Mitsou, N., Landsiedel, C., Kuehnlenz, K., Wollherr, D., Gool, L.V., Buss, M.: Real-time 3d hand gesture interaction with a robot for understanding directions from humans. In: 20th IEEE International Symposium on Robot and Human Interactive Communication, pp. 357–362 (2011)
4. Bhattacharya, S., Czejdo, B., Perez, N.: Gesture classification with machine learning using kinect sensor data. In: Third International Conference on Emerging Applications of Information Technology (EAIT), pp. 348–351 (2012)
5. Biswas, K., Basu, S.: Gesture recognition using microsoft kinect. In: 5th International Conference on Automation, Robotics and Applications (ICARA), pp. 100–103 (2011)
6. Bodiroža, S., Doisy, G., Hafner, V.: Position-invariant, real-time gesture recognition based on dynamic time warping. In: Proceedings of the 8th ACM/IEEE International Conference on Human-Robot Interaction, pp. 87–88. IEEE Press (2013)
7. Castiello, C., D'Orazio, T., Fanelli, A., Spagnolo, P., Torsello, M.: A model free approach for posture classificatin. In: IEEE Conference on Advances Video and Signal Based Surveillance, AVSS (2005)
8. Cruz, L., Lucio, F., Velho, L.: Kinect and RGBD images: Challenges and applications. In: XXV SIBGRAPI IEEE Confernce and Graphics, Patterns and Image Tutorials, pp. 36–49 (2012)
9. Gu, Y., Do, H., Ou, Y., Sheng, W.: Human gesture recognition through a Kinect sensor. In: IEEE International Conference on Robotics and Biomimetics (ROBIO), pp. 1379–1384. IEEE (2012)
10. Hachaj, T., Ogiela, M.: Rule-based approach to recognizing human body poses and gestures in real time. Multimedia Syst. **20**, 81–99 (2014)
11. Iosifidis, A., Tefas, A., Pitas, I.: View invariant action recognition based on artificial neural networks. IEEE Trans. Neural Netw. Learn. Syst. **23**(3), 412–424 (2012)
12. Iosifidis, A., Tefas, A., Pitas, I.: Multi view action recognition based on action volumes fuzzy distances and cluster discriminant analysis. Sig. Process. **93**, 1445–1457 (2013)
13. Oh, J., Kim, T., Hong, H.: Using binary decision tree and multiclass svm for human gesture recognition. In: International Conference on Information Science and Applications (ICISA), pp. 1–4 (2013)
14. Lai, K., Konrad, J., Ishwar, P.: A gesture-driven computer interface using kinect. In: IEEE Southwest Symposium on Image Analysis and Interpretation (SSIAI), pp. 185–188 (2012)
15. Leo, M., Spagnolo, P., D'Orazio, T., Distante, A.: Human activity recognition in archaeological sites by hidden markov models. In: Aizawa, K., Nakamura, Y., Satoh, S. (eds.) PCM 2004. LNCS, vol. 3332, pp. 1019–1026. Springer, Heidelberg (2004)
16. Miranda, L., Vieira, T., Martinez, D., Lewiner, T., Vieira, A., Campos, M.: Real-time gesture recognition from depth data through key poses learning and decision forests. In: 25th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), pp. 268–275 (2012)
17. Saponaro, G., Salvi, G., Bernardino, A.: Robot anticipation of human intentions through continuous gesture recognition. In: International Conference on Collaboration Technologies and Systems (CTS), pp. 218–225. IEEE (2013)
18. Shoemake, K.: Animating rotation with quaternion curves. In: SIGGRAPH'85 Proceedings of the 12th Annual Conference on Computer Graphics and Interactive Techniques, vol. 19(3), pp. 245–254 (1985)