International Journal of Advanced Robotic Systems

OPEN ACCESS ARTICLE

A Kinect-based Gesture Recognition Approach for a Natural Human Robot Interface

Regular Paper

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

1 Institute of Intelligent Systems for Automation, Bari, Italy
2 The Polytechnic University of Bari, Bari, Italy
*Corresponding author(s) E-mail: grace@ba.issia.cnr.it

Received 02 October 2014; Accepted 17 November 2014

DOI: 10.5772/59974

Abstract

In this paper, we present a gesture recognition system for the development of a human-robot interaction (HRI) interface. Kinect cameras and the OpenNI framework are used to obtain real-time tracking of a human skeleton. Ten different gestures, performed by different persons, are defined. Quaternions of joint angles are first used as robust and significant features. Next, neural network (NN) classifiers are trained to recognize the different gestures. This work deals with different challenging tasks, such as the real-time implementation of a gesture recognition system and the temporal resolution of gestures. The HRI interface developed in this work includes three Kinect cameras placed at different locations in an indoor environment and an autonomous mobile robot that can be remotely controlled by one operator standing in front of one of the Kinects. Moreover, the system is supplied with a people re-identification module which guarantees that only one person at a time has control of the robot. The system's performance is first validated offline, and then online experiments are carried out, proving the real-time operation of the system as required by a HRI interface.

Keywords Feature extraction, Human gesture modelling, Gesture recognition, Gesture segmentation

1. Introduction

In recent decades, the development of highly advanced robotic systems has seen them spread throughout our daily lives in several application fields, such as social assistive robots [1, 2], surveillance robots [3] and tour-guide robots [21], etc. As a consequence, the development of new interfaces for HRI has received increasing attention in order to provide a more comfortable means of interacting with remote robots and encourage non-experts to interact with robots. Up to now, the most commonly-used human-robot interfaces ranged from mechanical contact devices, such as keyboards, mice, joysticks and dials, to more complex contact devices, such as inertial sensors, electromagnetic tracking sensors, gloves and exoskeletal systems. Recently, the latest trend has been to develop different human-robot interfaces that are contactless, non-invasive, more natural and more human-centred. This trend is increasingly prominent thanks to the recent diffusion of low-cost depth sensors, such as the Microsoft Kinect. This 3D camera allows the development of natural human-robot interaction interfaces, as it generates a depth map in real-time. Such HRI systems can recognize different gestures accomplished by a human operator, simplifying the interaction process. In this way, robots can be controlled easily and naturally.

In this paper, we will focus on the development of a gesture recognition system by using the Kinect sensor with the aim of controlling a mobile autonomous robot (PeopleBot platform). At present, gesture recognition through visual and depth information is one of the main active research topics in the computer vision community. The Kinect provides synchronized depth and colour (RGB) images whereby each pixel corresponds to an estimate of the distance between the sensor and the closest object in the scene together with the RGB values at each pixel location. By using this data, several natural user interface libraries which provide human body skeleton information have been developed. The most commonly-used libraries are OpenNI (Open Natural Interaction), which is used in this paper, NITE Primesense, libfreenect, CL NUI, Microsoft Kinect SDK and Evoluce SDK.

2. Related work

In recent years, the scientific literature on the development of gesture recognition systems for the construction of HRI interfaces has expanded substantially. Such an increase can be attributed to the recent availability of affordable depth sensors such as the Kinect, which provides both appearance and 3D information about the scene. Many papers presented in the literature in the last couple of years have used Kinect sensors [5-13] and they select several features and different classification methods to build gesture models in several applicative contexts.

This section will overview those studies mainly focused on gesture recognition approaches in the HRI context [14] and (if not explicitly expressed) which use the Kinect camera as a RGB-D sensor. The ability of the OpenNI framework to easily provide the position and segmentation of the hand has stimulated many approaches to the recognition of hand gestures [8, 15, 16]. The hand's orientation and four hand gestures (open hand, fist, pointing index and pointing index and thumb) are recognized in [15] to interact with a robot which uses the recognized pointing direction to define its goal on a map. First, the robot detects a person in the environment and the hand tracker is initialized by detecting a waving hand as a waving object in the foreground of the depth image. Next, by using an example-based approach, the system recognizes the hand gestures that are translated into interaction commands for the robot. In [16], static hand gestures are also recognized to control a hexagon robot by using the Kinect and the Microsoft SDK library. Xu et al., in [8], also propose a system that recognizes seven different hand gestures for interactive navigation by a robot, although in contrast to the previously cited works, the gestures are dynamic. This involves the introduction of a start-/end-point detection method for segmenting the 3D hand gesture from the motion trajectory. Successively, hidden Markov models (HMMs) are implemented to model and classify the segmented gesture. HMMs are also applied in [17] for dynamic gesture classification. First, an interactive hand-tracking strategy based on a Camshift algorithm and which combines both colour and depth data is designed. Next, the gestures classified by HMM are used to control a dual-arm robot. Dynamic gestures based on arm tracking are instead recognized in [18]. The proposed system is intended to support natural interaction with autonomous robots in public places, such as museums and exhibition centres. The proposed method uses the arm joint angles as motor primitives (features) that are fed to a neural classifier for the recognition process. In [19], the movement of the left arm is recognized to control a mobile robot. The joint angles of the arm with respect to the person's torso are used as features. A preprocessing step is first applied to the data in order to convert feature vectors into finite symbols as discrete HMMs are considered for the recognition phase. Human motion is also recognized by using an algorithm based on HMMs in [20]. A service robot with an on-board Kinect receives commands from a human operator in the form of gestures. Six different gestures executed by the right arm are defined and the joint angles of the elbow are used as features.

The work presented in this paper in principle follows the previously cited works. The aim of this work lies in the development of a more general and complex system which includes three Kinect cameras placed in different locations of an indoor environment and an autonomous mobile robot that can be remotely controlled by one operator standing in front of one of the Kinects. The operator performs a particular gesture which is recognized by the system and sent to the mobile robot as a proper control command. The quaternions of the arms' joints (returned by the OpenNi framework) are used as input features to several NNs, each one trained to recognize one predefined gesture. Ten gestures have been selected among the signals of the USA army [21].

Such a system involves some challenging tasks, such as:

- The real-time recognition of gestures;
- The spatial and temporal resolution of gestures;
- The independence of the gesture classifier from users.

All these problems have been investigated and tackled in depth. One of the main aims of this work is to find a solution to one of the crucial issues related to real-time application, viz., gesture segmentation. This deals with the detection of the starting and ending frame of each gesture and the normalization of the lengths of different gestures. An algorithm based on a fast Fourier transform (FFT) has been applied to solve this problem. Furthermore, the proposed system has been provided with the ability to avoid false positives when the user is not involved in any gesture. Moreover, the system is supplied with a person re-identification [22] module which guarantees that only one person at a time controls the robot through gestures. This is a peculiarity that has been added to the system in order to guarantee the precise control of the robot. Real experiments demonstrate both the real-time applicability and the robustness of the proposed approach in terms of detection performance. The system performance is first validated offline using video sequences acquired by a Kinect camera. Next, online experiments are carried out proving the real-time operation of the system as required by a human-robot interactive interface.

The proposed system is described in section 3, whereas section 4 presents the experiments carried out in our office-like environment. Finally, section 5 outlines some conclusions and discussions.

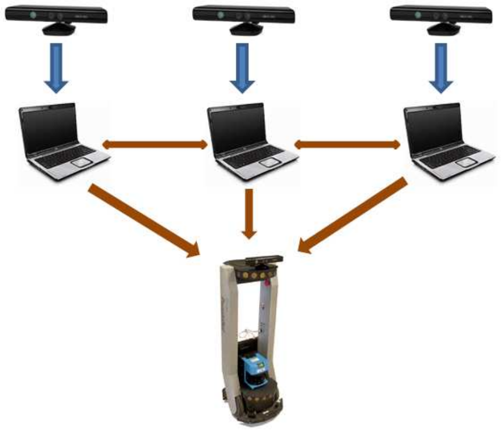


Figure 1. Scheme of the system architecture: the Kinects are connected to the computers which in turn wirelessly communicate with each other and with the mobile robot

3. The proposed system

In Figure 1, a scheme of the system architecture is shown. A human operator can take control of the mobile agent from any one of the three cameras, which are connected to three computers where the gesture recognition module processes the image sequences continuously and provide motion commands to the robot controller. The human operator who is to control the mobile agent stays in front of one of these cameras and executes an initialization gesture that allows the Kinect sensor to calibrate and activate the tracking procedure. From that moment onwards, all the gestures obtained by the real-time analysis of the skeleton are converted into control commands and are sent to the mobile agent.

A display close to each Kinect sensor visualizes the environment map with the current robot position (see Figure 2) and provides messages to the operator to signal whether any gestures have been recognized and translated in a command or not. Only one person at a time is allowed to control the mobile agent. Therefore, the first time he/she enters the field of view of one camera, he/she has to execute the initialization gesture. Once the initialization gesture is recognized by the system, the RGB image silhouette of the person is used by the person re-identification module to build the signature of that person. As such, the person re-identification module guarantees that the person controlling the robot is the same if either the person exits the field of view of one Kinect or else enters that of another Kinect. The OpenNI framework instead guarantees the tracking of the person during the execution of gestures. If another user has to take the robot control, it is necessary that he/she executes the initialization gesture, and the process starts again.

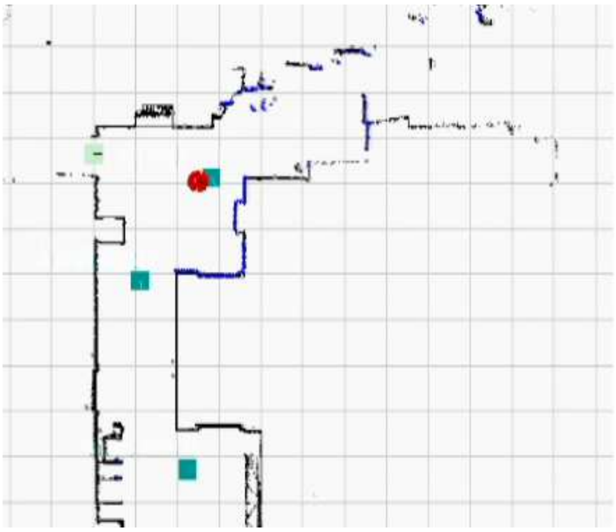


Figure 2. Snapshot of a section of the environment map highlighting the current robot position (red circle) and some goal positions (green squares)

Gesture segmentation is another fundamental cue for the success of a gesture recognition method and it must be solved before recognition in order to make the model-matching more efficient. Therefore, a periodicity analysis is carried out in order to extract and normalize the gesture lengths and to acquire data comparable to the generated models. First, some features are extracted from the image sequences and are provided to different NN classifiers, each one trained to recognize a single gesture. In addition, in order to be independent of the starting frame of the sequence, a sliding window and consensus analysis are used to make a decision in relation to the recognized gesture. In this way, the proposed system has no need for a particular assumption about special boundary criteria, no need for a fixed gesture length, and no need for multiple window sizes, thus avoiding an increase in computational load.

In the following subsection, a brief overview of a person re-identification module will be given. Gesture recognition and gesture segmentation approaches will be detailed.

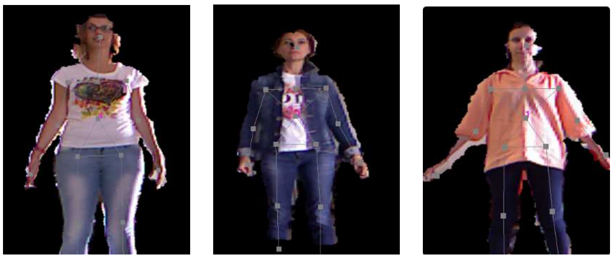


Figure 3. Sample images of segmented silhouettes

3.1 Person re-identification module

In order to allow the control of the robot by different cameras, a person re-identification algorithm has been implemented in order to allow the recognition of the same person in different parts of the environment. In this work, a modified version of the methodology published in [23] has been applied. The Kinect cameras together with the OpenNI framework allow for the easy segmentation of people entering in the camera's field of view. In fact, the extraction of the corresponding RGB silhouette is immediate and avoids problems such as people-motion detection and background subtraction, which are instead investigated in [23]. Thanks to the OpenNI framework, each time a person enters into the Kinect's field of view, the modules of the person segmentation and tracking associate a new ID to that person. Furthermore, the bounding box containing his/her RGB segmented silhouette is available for further processing. In Figure 3, some sample images of segmented silhouettes are shown. By using these images, a signature (or feature set) which provides a discriminative profile of the person is obtained and is used for recognition when new instances of persons are encountered. In our approach, the signature is based on the evaluation of colour similarity among uniform regions and the extraction of robust relative geometric information that is persistent when people move in the scene. The signature can be estimated on one or more frames, and a distance measure is introduced to compare signatures extracted by different instances of people returned by the human-tracking module. As such, the method can be summarized in the following steps:

- First of all, for each frame a segmentation of the silhouette in uniform regions is carried out;
- For each region, some colour and area information is evaluated;
- A connected graph is generated: nodes contain the information of each region, such as colour histograms and area occupancy, while connections among nodes contain information on the contiguity of regions.
- A similarity measure is introduced to compare graphs generated by different instances that considers some relaxation rules to handle the varying appearance of the same person when observed by different point of views.

In order to recognize the same person in different images or when the person exits and re-enters in view of the same camera, a decision about the similarity measures has to be taken. By experimental validation, some sets of signatures of different persons are evaluated and a threshold on possible variations of inter-class signatures (the signatures of the same person) is estimated. If the similarity of two different signatures is under this threshold, the person is recognized. For more details about the person re-identification method, see [23].

3.2 The gesture recognition approach

The OpenNI framework provides the human skeleton with the joint coordinates useful for gesture recognition. Different gestures executed with the right arm are selected from the "Arm-and-Hand Signals for Ground Forces" [21]. Figure 4 shows the gestures that were chosen for the experiments. Throughout this paper, we will refer to these gestures using the following symbols G_1 , G_2 , G_3 ,..., G_{10} .

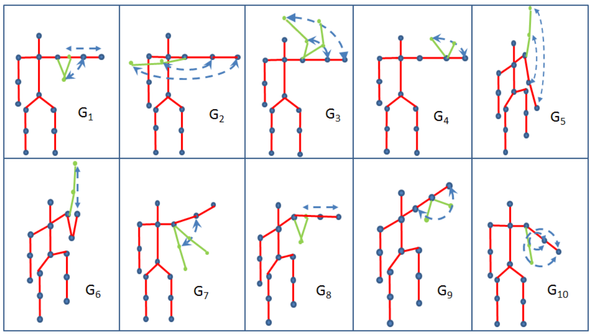


Figure 4. Ten different gestures selected from the army visual signals report [21] are shown. Gestures G_5 , G_6 , G_8 and G_9 are pictured in a perspective view as the arm engages in a forward motion. In gestures G_1 , G_2 , G_3 , G_4 , G_7 and G_{10} , the arm has lateral motion instead, and so the front view is drawn.

The problem of selecting significant features which preserve important information for classification is fundamental for gesture recognition. Different information is used in the literature, and many papers consider the coordinate variations of some joints such as the hand, the elbow, the shoulder and the torso nodes. However, when coordinates are used, a kind of normalization is needed in order to be independent of the position and the height of the people performing the gestures. For this reason, in many cases joint orientations are preferred as they are more robust and are independent of the position and the height of the gesturer. In this paper, two types of features are compared - the angles and the quaternions of the joint nodes.

In the first case, three joint-angles are selected: α defined among hand/elbow/shoulder joints, β among elbow/shoulder/torso joints, and γ among elbow/right shoulder/

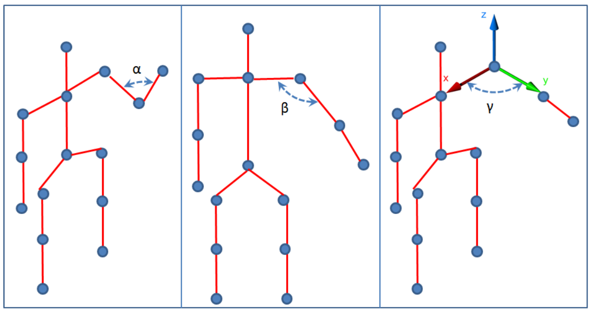


Figure 5. Joint angles used as features: α angle among hand/elbow/shoulder joints, β angle among elbow/shoulder/torso joints, and γ among elbow/right shoulder/left shoulder joints in the XY plane

left shoulder joints (see Figure 5). These three joint angles produce a feature vector V_i for each frame i :

$$V_i = [\alpha_i, \beta_i, \gamma_i]$$

In the second case, the quaternions of the right shoulder and elbow nodes are selected. A quaternion is a set of numbers that comprises a 4D 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. The quaternions of the right shoulder and elbow nodes produce a feature vector for each frame i defined by:

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

where the index s stands for the shoulder and e stands for the elbow.

Once the feature vectors have been defined, the models of the gestures are learned by using 10 different NNs, one for each gesture. Concordantly, 10 different training sets are constructed in light of the feature vector sequences of the same gesture as positive examples and the feature vector sequences of the other gestures as negative examples. To this end, different people were asked to repeat the gestures. As a consequence, the length of the gesture - in terms of number of frames - can greatly vary. Accordingly, in order to be invariant with respect to execution of the gestures, a pre-processing step was applied. The feature sequences were sampled within a fixed interval. The interval length was fixed to 60 frames, and so the gesture duration is around two seconds since the frame rate of the Kinect is 30 fps. Linear interpolation in order to re-sample (up-sampling or down-sampling) the number of frames was applied, as it marks a good compromise between computational burden and quality of results.

The architecture of the NNs was defined relative to the types of features. As such, in the case where the joint angles α,β and γ are used as features, each NN has an input layer of 180 nodes (three features for 60 frames). Conversely, in the case of quaternions, the input layer of each NN has 480 nodes (eight features for 60 frames). In both cases, each NN has one hidden layer and one node in the output layer which has been trained to return 1 if a gesture is recognized and zero otherwise. A backpropagation algorithm was used for the training phase, whereas the best configuration of hidden nodes was selected in a heuristic manner after several experiments. At the end of training, offline testing was carried out. The sequence of features of one gesture is provided to all 10 NNs and the one which returns the maximum answer is considered. If this maximum answer is above a fixed threshold, the sequence is recognized as the corresponding gesture, otherwise it is considered as a non-gesture.

3.3 Length gesture estimation and gesture segmentation

As mentioned above, the length of a gesture can vary if either the gesture is executed by the same person or else by different people. Furthermore, during the online operation of the gesture recognition module, it is not possible to know the starting frame and the ending frame of each gesture. For these reasons, two further steps are introduced.

First, in order to solve the problem of gesture length, a FFT-based approach was applied. Repeating a gesture a number of times, it is possible to approximate the feature sequence as a periodic signal. Accordingly, different people were asked to repeat the same gesture without interruption and all the frames of the sequences were recorded. Applying the FFT and tacking the position of the fundamental harmonic component, the period could be evaluated as the reciprocal value of the peak position. The estimated period was then used to interpolate the sequence of feature vectors in 60 values which could be provided to the 10 NNs.

Furthermore, the performance of the gesture recognition module can worsen during the online operation if the sequence of frames provided to the classifiers does not contain the exact gesture. In other words, the starting and ending frames are not know *a priori*. As such, a sliding window approach was applied as shown in Figure 6. The video sequences were divided into multiple overlapping segments of n frames, where n is the period of the gesture evaluated in the previous step with the FFT. Next, these segments were resized with the linear interpolation in order to generate windows of the same size as those used during the training phase, and were then fed to all 10 NNs. Consensus decision-making was applied to recognize the sequence as one gesture. This was based - again - on a sliding window approach and was used to evaluate the number of consecutive concordant answers of the same NN.

4. Experimental results

Different experiments were executed: offline experiments for the performance evaluation of the gesture recognition module; online experiments for the gesture-length estimation and segmentation; finally, a selection of gestures was used as robot commands to control in real-time the PeopleBot mobile platform, allowing its navigation in the environment.

4.1 Gesture recognition: offline experiment

The gesture recognition module was tested using a database of 10 gestures performed by 10 different people. Some gestures performed by five subjects were used to train the 10 NNs, while the remaining gestures together with all the sequences of the other five people were used for the test. Initially, the experiments were distinguished by first considering the five people whose gestures were in the training sets (the first five people) and then the other five people. This difference is important, as the execution of the same gestures can be very different if either they are executed by the same people in different sessions or by new people.

Sequences of 24, 26, 18, 22, 20, 20, 21, 23, 23 and 25 executions of gestures G_1, G_2, G_3, G_4, G_5, G_6, G_7, G_8, G_9 and G_{10}, respectively, performed by the first five people were used as positive examples in the 10 training sets for the 10 NNs. In other words, the first training set contains 24 positive examples of gesture G_1 and the remaining executions (i.e., 26 of G_2, 18 of G_3, 22 of G_4, and so on) as negative examples. The same strategy was then used for building the training sets of the other nine NNs.

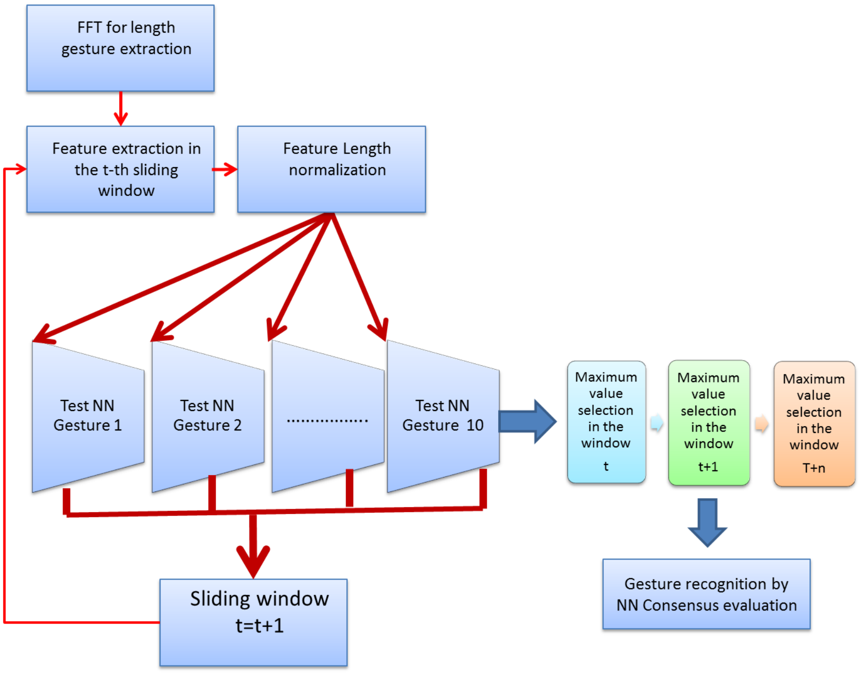

FFT for length gesture extraction
Feature extraction in the t-th sliding window
Feature Length normalization
Test NN Gesture 1
Test NN Gesture 2
……………….
Test NN Gesture 10
Sliding window t=t+1
Maximum value selection in the window t
Maximum value selection in the window t+1
Maximum value selection in the window T+n
Gesture recognition by NN Consensus evaluation

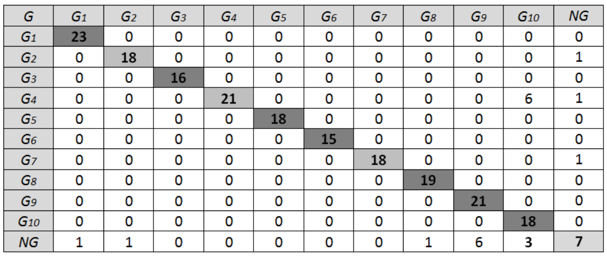
Figure 6. The proposed approach for the on line gesture recognition module

G	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10
G1	23	0	0	0	0	0	0	0	0	0
G2	0	19	0	0	0	0	0	0	0	0
G3	0	0	16	0	0	0	0	0	0	0
G4	0	0	0	16	0	0	0	0	0	6
G5	0	0	0	0	18	0	0	0	0	0
G6	0	0	0	0	0	14	0	0	1	0
G7	0	0	0	0	0	0	19	0	0	0
G8	0	0	0	0	0	0	0	19	0	0
G9	1	0	0	7	0	0	0	0	13	0
G10	0	0	0	0	0	0	0	0	0	18

Figure 7. The scatter matrix for the recognition of the 10 gestures when the *joint angles* are used as features. Tests performed on gesture sequences executed by the same people as used in the training set.

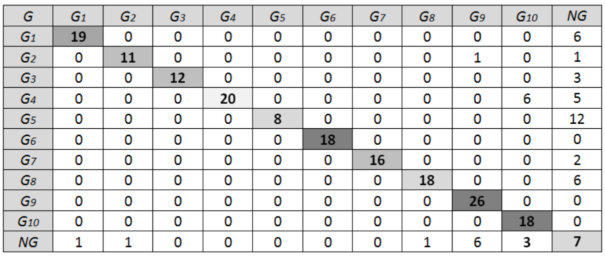
G	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10
G1	23	0	0	0	0	0	0	0	0	0
G2	0	19	0	0	0	0	0	0	0	0
G3	0	0	16	0	0	0	0	0	0	0
G4	0	0	0	21	0	0	0	0	0	1
G5	0	0	0	0	18	0	0	0	0	0
G6	0	0	0	0	0	15	0	0	1	0
G7	0	0	0	0	0	0	19	0	0	0
G8	0	0	0	0	0	0	0	19	0	0
G9	0	0	0	0	0	0	0	0	21	0
G10	0	0	0	0	0	0	0	0	0	18

Figure 8. The scatter matrix for the recognition of the 10 gestures when the *quaternions* are used as features. Tests performed on gesture sequences executed by the same people as used in the training set.



G	G_1	G_2	G_3	G_4	G_5	G_6	G_7	G_8	G_9	G_{10}	NG
G_1	23	0	0	0	0	0	0	0	0	0	0
G_2	0	18	0	0	0	0	0	0	0	0	1
G_3	0	0	16	0	0	0	0	0	0	0	0
G_4	0	0	0	21	0	0	0	0	0	6	1
G_5	0	0	0	0	18	0	0	0	0	0	0
G_6	0	0	0	0	0	15	0	0	0	0	0
G_7	0	0	0	0	0	0	18	0	0	0	1
G_8	0	0	0	0	0	0	0	19	0	0	0
G_9	0	0	0	0	0	0	0	0	21	0	0
G_{10}	0	0	0	0	0	0	0	0	0	18	0
NG	1	1	0	0	0	0	0	1	6	3	7

Figure 9. The scatter matrix for the recognition of the 10 gestures with *quaternions* as features and with the threshold $Th = 0.7$. Tests performed on gesture sequences executed by the same subjects as used in the training set.



G	G_1	G_2	G_3	G_4	G_5	G_6	G_7	G_8	G_9	G_{10}	NG
G_1	19	0	0	0	0	0	0	0	0	0	6
G_2	0	11	0	0	0	0	0	0	1	0	1
G_3	0	0	12	0	0	0	0	0	0	0	3
G_4	0	0	0	20	0	0	0	0	0	6	5
G_5	0	0	0	0	8	0	0	0	0	0	12
G_6	0	0	0	0	0	18	0	0	0	0	0
G_7	0	0	0	0	0	0	16	0	0	0	2
G_8	0	0	0	0	0	0	0	18	0	0	6
G_9	0	0	0	0	0	0	0	0	26	0	0
G_{10}	0	0	0	0	0	0	0	0	0	18	0
NG	1	1	0	0	0	0	0	1	6	3	7

Figure 10. The scatter matrix for the recognition of the 10 gestures with *quaternions* as features and with the threshold $Th = 0.4$. Tests executed on gesture sequences executed by people not included in the training set.

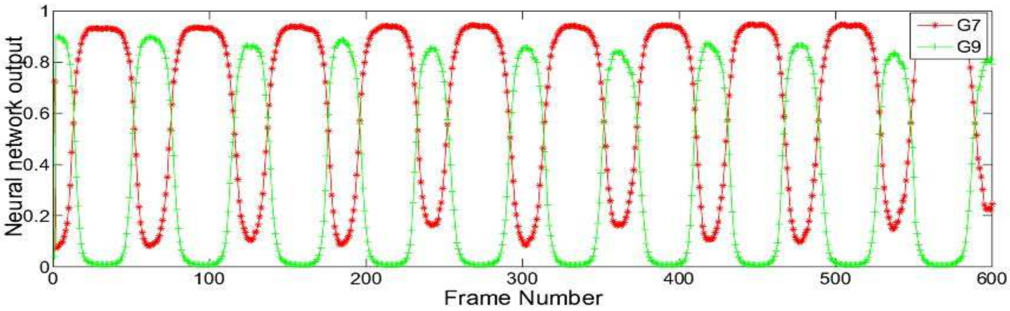


Figure 11. The results of the gesture recognition over a sequence of 600 frames during the continuous execution of gesture G_7

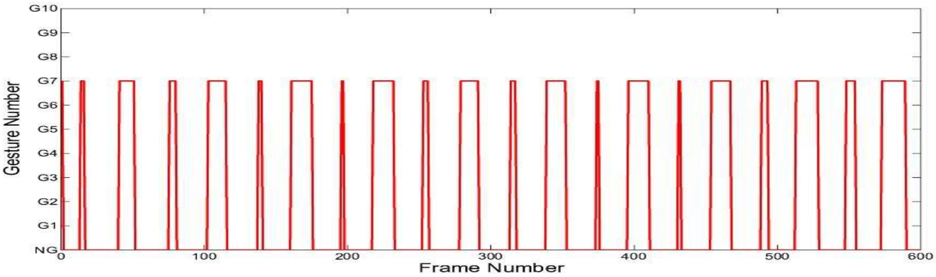


Figure 12. The results of the gesture recognition when the decision-making approach is applied: G_7 is correctly recognized

Analogously, the tests were carried out using 23, 19, 16, 22, 18, 15, 19, 19, 21 and 18 executions of gestures G_1 , G_2 , G_3 , G_4 , G_5 , G_6 , G_7 , G_8 , G_9 and G_{10} , respectively, performed by the same five people as in the training set. In Figures 7 and 8, the scatter matrices of two tests are reported: the first refers to the results obtained using the joint angles as features, whereas the second refers to those obtained when quaternions are used. As expected, the results worsen in the case of joint angles. Actually, they are not able to disambiguate among certain different gestures: as an example, G_9 is mistaken for G_4 and, conversely, G_4 is mistaken for G_{10} . In fact, these gestures involve the same β and γ joint angles. Only the α angle is variable, but this variation is not unambiguously recognized by the classifiers. The cause of this ambiguity is that the rotation axes of the joints are not involved in this case. Quaternions, in contrast, include not only the rotation angles but also the rotation axes. Indeed, the results shown in Figure 8 that refer to the case of quaternions as features are better; the only failure case is for gesture G_4 , which is mistaken for G_{10} .

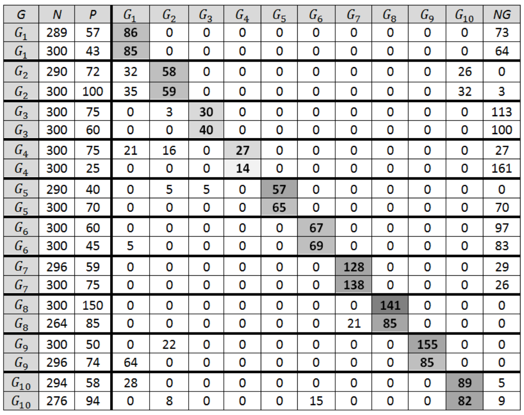
After the previous tests, quaternions were chosen as features for all the subsequent experiments. First of all, some experiments were carried out by introducing frame sequences that do not belong to any of the 10 gestures. This is for introducing a non-gesture (NG) class, which is important as people can perform idle gestures which must be associated with an NG class. Indeed, the results could become significantly worse since in any case the executed gesture is always classified by one NN, even erroneously. As such, a threshold Th is defined in order to evaluate the maximum answer among the 10 outputs of the NNs: if this maximum value is under the threshold, the gesture is classified as a non-gesture. In Figure 9, the scatter matrix obtained using $Th = 0.7$ is shown. As can be seen, some gestures that were correctly classified in the previous test (see Figure 8) are classified as non-gestures in this new case (see the last column (NG) of Figure 9). According to the threshold value, the number of false negatives (gestures recognized as non-gestures) and true negatives (non-gestures correctly classified as non-gestures) can vary greatly. Accordingly, a final test was carried out whose results are reported in Figure 10: in this case, sequences of gestures executed by people not included in the training sets are fed to the NNs. As expected, the output values returned by the 10 NNs are smaller than those obtained when the same people of the training set execute the gestures. This is because different people can execute gestures with their own personal interpretation of the gestures. However, the recognition of the 10 gestures is always guaranteed using a smaller threshold value of $Th = 0.4$.

4.2 Gesture recognition: online experiment

In order to test the ability of the system to perform recognition when people execute the gestures continuously and with different velocities, online experiments were carried out. In this case, the gesture recognition module has to work continuously, i.e., during frame acquisition, and so it has to manage different gesture lengths and has no knowledge about the starting and ending frames of the gesture sequence. As described in section 3.3, a sliding window approach was applied.

In order to clarify these problems, gestures G_7 and G_9 are used as examples since their respective NNs provide similar output values. In Figure 11, the output values of the NNs of G_7 (NN_7) and G_9 (NN_9) are reported on a sequence of 600 frames when a person executes gesture G_7 . Notice that 600 frames roughly correspond to 10 executions of a gesture. During the online experiment, sequences of 300 frames are selected for processing supposing that the gesture has been executed at least five times. In Figure 11 is shown, in red (*), the correct answers of NN_7, and in green (+), the wrong answers of NN_9. The output values of the remaining NNs are negligible as they are close to 0. It is evident that for most of the time the maximum answers are provided by the correct network (NN_7), but at some regular intervals NN_9 also provides large values. The wrong answers are justified by the fact that the sliding window - when it slides on frames - encloses sequences that can be confused with the erroneous gesture. In other words, some gestures contain common sub-sequences that can increase ambiguity; therefore, more NNs return a high output value. This happens until the sliding window ceases to enclose the correct sequence (from the first to the last frames) corresponding to the current gesture. This can be seen in Figure 11 where, since the starting and ending sections of G_7 and G_9 are the same, both NN_7 and NN_9 return high output values. The central section of G_7 and G_9 , instead, are different and so NN_7 correctly provides the maximum value and NN_9 correctly provides the minimum value. To solve this problem, the consensus decision-making approach described in section 3.3 is applied. A counter is assigned to each gesture. First, the number of consecutive concordant answers of the same NN is counted and, if it is greater than a fixed threshold (heuristically fixed to 10 in our case), then the gesture counter is incremented by one. The gesture which has the counter at the maximum value is the recognized one. In Figure 12, a graph of the decision-making process is shown: most of the time, G_7 is now correctly recognized while, in the remaining intervals, a non-gesture is returned.

As an additional test, the system was pressed to recognize all the gestures executed with different velocities. For all the gestures, two different experiments were carried out: each gesture has been performed by one user (not included in the training set) at two different velocities. Sequences of 300 frames were observed and the FFT module evaluated the period of the gestures as described in section 3.3. As has already been mentioned, the period is used to estimate the size of the sliding window which in turn is used to resize and extract the sequence of frames fed to the 10 NNs. In Figure 13, the results are reported. In the first column, N represents the number of frames extracted by the proposed system in the window of 300 observations. This number can be less than 300, as the feature extraction algorithm depends upon the results of the skeleton detection made by the OpenNI framework. If the skeleton detection fails, the corresponding frames are not considered by the gesture recognition software. As an example, during the first execution of the gesture G_1 , only 289 frames were considered (see Figure 13). In the second column of the table, the symbol P refers to the period estimated by the FFT procedure highlighting the different velocities of the gesture executions. By 'velocity' we mean the number of frames that contain the execution of one gesture. As an example, and considering again gesture G_1 , the two different velocities are represented by $P = 57$ and $P = 43$, meaning that the gesture G_1 performed slower in the first case and faster in the second case. The remaining columns of the table report how many times the corresponding NN has obtained concordant answers above the fixed threshold. The numbers reported in bold in the main diagonal demonstrate that the proposed system is able to recognize the gesture whatever its velocity may be. Some false positives are detected, but they are due first to the ability of the Kinect software to extract correctly the skeleton and the joint positions, and secondly to the similarity of many gestures in some portions of their movement. For example, gesture G_9 is often confused with G_1 and G_2 since, in some portions of the gestures, the features have the same variations. However, as we can observe in Figure 13, the maximum value of the concordant answers is always associated with the correct gesture thanks to the decision-making process described above.



G	N	P	G_1	G_2	G_3	G_4	G_5	G_6	G_7	G_8	G_9	G_{10}	NG
G_1	289	57	86	0	0	0	0	0	0	0	0	0	73
G_1	300	43	85	0	0	0	0	0	0	0	0	0	64
G_2	290	72	32	58	0	0	0	0	0	0	0	26	0
G_2	300	100	35	59	0	0	0	0	0	0	0	32	3
G_3	300	75	0	3	30	0	0	0	0	0	0	0	113
G_3	300	60	0	0	40	0	0	0	0	0	0	0	100
G_4	300	75	21	16	0	27	0	0	0	0	0	0	27
G_4	300	25	0	0	0	14	0	0	0	0	0	0	161
G_5	290	40	0	5	5	0	57	0	0	0	0	0	0
G_5	300	70	0	0	0	0	65	0	0	0	0	0	70
G_6	300	60	0	0	0	0	0	67	0	0	0	0	97
G_6	300	45	5	0	0	0	0	69	0	0	0	0	83
G_7	296	59	0	0	0	0	0	0	128	0	0	0	29
G_7	300	75	0	0	0	0	0	0	138	0	0	0	26
G_8	300	150	0	0	0	0	0	0	0	141	0	0	0
G_8	264	85	0	0	0	0	0	0	21	85	0	0	0
G_9	300	50	0	22	0	0	0	0	0	0	155	0	0
G_9	296	74	64	0	0	0	0	0	0	0	85	0	0
G_{10}	294	58	28	0	0	0	0	0	0	0	0	89	5
G_{10}	276	94	0	8	0	0	0	15	0	0	0	82	9

Figure 13. The results of the proposed gesture recognition approach when the gestures are executed with different velocities. N represents the number of observed frames. P refers to the number of frames containing the execution of one gesture (velocity of execution).

Gesture	Command
G_1	Initialization
G_2	Home
G_3	Go to the Goal
G_4	Turn Around
G_5	Go Wondering
G_6	Stop

Table 1. The gestures and the associated commands

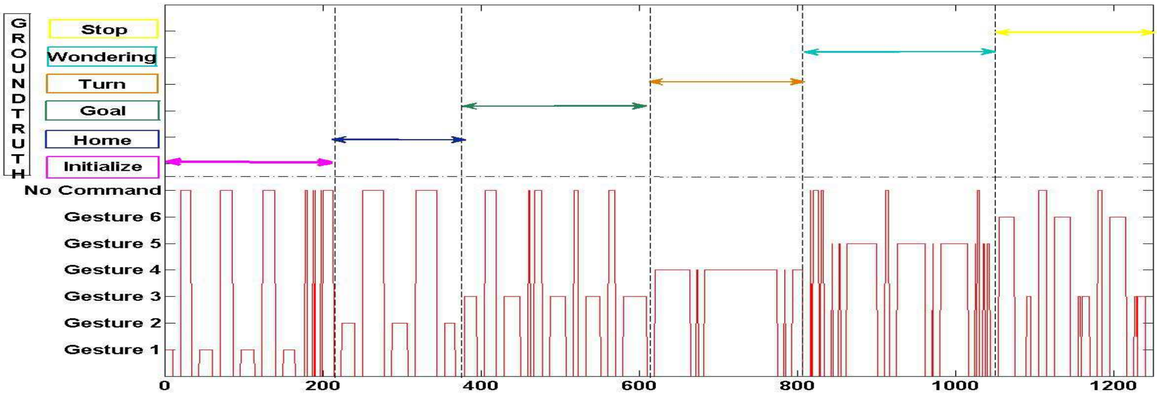


Figure 14. The results of the gesture recognition when a sequence of gestures is executed in real-time. At the top of the figure, the ground truth is pictured, whereas at the bottom the system answers are reported.

4.3 Human-robot interaction experiments

In this section, the online experiments of HRI are described. Among all the gestures, six of them (see Table 1) were selected for the real-time control of the mobile platform. In order to evaluate the performance of the gesture recognition system, in Figure 14 we report the results when a user faces the Kinect sensor and performs the six selected gestures in a continuous way in order to control the mobile vehicle. At the top of Figure 14, the ground truth of the six gestures is reported, while at the bottom the corresponding system's answers are plotted. The No-Command answer is obtained when the number of consecutive concordant answers of the same network is over the established threshold (10 in our case), but their values are below the *Th* threshold. In this case the decision is considered unreliable and no command is sent to the robot. As shown in Figure 14 the system is mostly able to recognize the corresponding gestures. Only gesture G_6 is sometimes erroneously confused with gesture G_3 . However, and as explained in the previous section, if the command decision is made after a sequence of 300 observations (frames), during which the FFT is applied, then in this case as well the correct command can be sent to the robot controller. In Figure 15, on the left column, the RGB images acquired by three Kinect sensors placed at different points in the environment are shown. On the right, the corresponding segmented silhouettes which are provided to the re-identification module are pictured. In the images at the top, the person is performing gesture G_1 (initialize), which activates the signature generation for the re-identification procedure. As such, even if the person enters another camera's field of view or else if more subjects are present in the same view (bottom of Figure 15), the system is always able to maintain focus on the same person who made the initialization and who took control of the robot. Figure 16 shows a snapshot of the robot, which reaches the goal position after the user performed gesture G_3 .

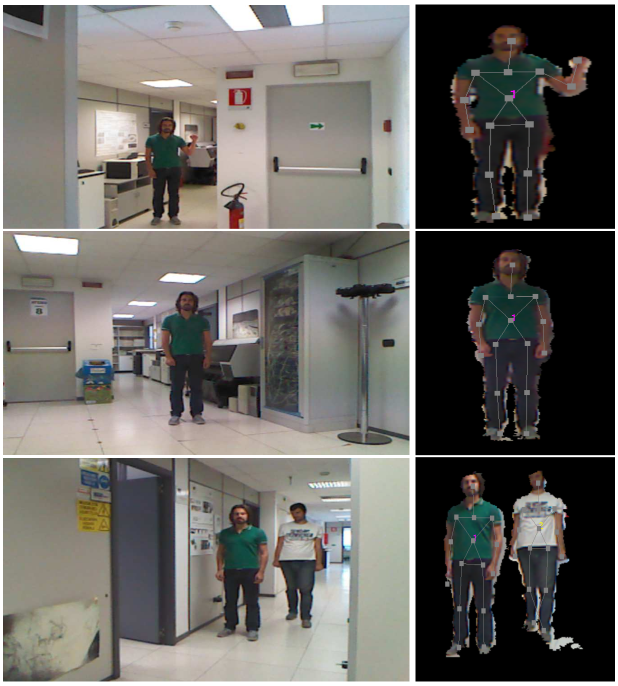


Figure 15. Some RGB images acquired by the Kinect cameras placed at different positions in the environment with the corresponding segmented images. At the top of the image is the detected silhouette and the skeleton of the person who took the control of the robot (initialization gesture).

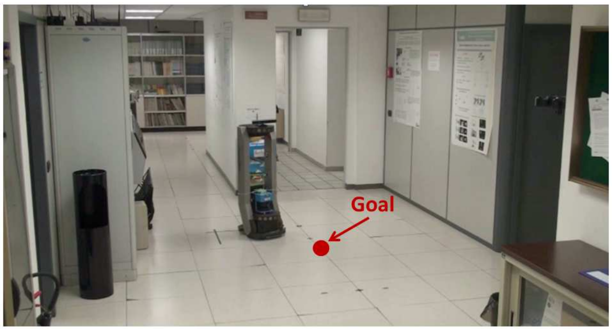


Figure 16. A snapshot of the robot reaching the goal position

5. Discussion and conclusions

In this paper, we propose a gesture recognition system based on a Kinect sensor which - in an effective way - provides both the people segmentation and the skeleton information for real-time processing. We use the quaternion features of the right shoulder and elbow nodes and different NNs to construct the models of 10 different gestures. Offline experiments demonstrate that the NNs are able to model the gestures in both cases: people either included or else not-included in the training set. Furthermore, as knowledge of the initial frame and the lengths of the gestures are not always guaranteed, we propose the use of the FFT for the period analysis and a sliding window approach to generate a decision-making process. In this way, when the sliding window encloses exactly the frames containing the gesture, the corresponding NN provides the maximum answer; meanwhile, when the window overlaps the ending or starting portions of the gesture, some false positive answers can arise. The obtained results are very encouraging, as the number of false positives is always smaller than that of true positives. Furthermore, by filtering the number of consecutive concordant answers of the NNs, the correct final decision is always taken. Offline tests have proved the ability of the system to recognize the gestures even if users different from those in the training set perform the gestures. The gesture-recognition system is used in a human-robot interface to remotely control a mobile robot in the environment. Each recognized gesture is coded into a proper command sent to the robot for navigation. Moreover, the system is supplied with a people re-identification module to ensure that only one person at a time can take control of the robot.

The main limitation of the proposed system is the need to observe additional repetitions of the same gesture in order to make a decision. This limit depends mainly on the noise and low precision of the skeleton data extracted by the Kinect sensor. More executions of the same gesture guarantee that aggregate behaviour can be extracted to characterize the gesture. In addition, gesture repetitions are necessary in order to extract the period and make the system independent of gesture length. Future work will address both the evaluation of different features directly extracted on the depth map and the use of different methodologies to align the signals, thereby allowing recognition upon the first execution of the gesture.

6. . References

[1] I. Almetwally and M. Mallem. Real-time tele-operation and tele-walking of humanoid robot Nao using Kinect depth camera. In *Proc. of 10th IEEE International Conference on Networking, Sensing and Control (ICNSC)*, 2013.

[2] M. Van den Bergh, D. Carton, R. de Nijs, N. Mitsou, C. Landsiedel, K. Kuehnlenz, D. Wollherr, L. Van Gool, and M. Buss. Real-time 3D hand gesture interaction with a robot for understanding directions from humans. In *Proc. of 20th IEEE international symposium on robot and human interactive communication*, pages 357–362, 2011.

[3] S. Bhattacharya, B. Czejdo, and N. Perez. Gesture classification with machine learning using kinect sensor data. In *Third International Conference on Emerging Applications of Information Technology (EAIT)*, pages 348–351, 2012.

[4] Biao Ma, Wensheng Xu, and Songlin Wang. A robot control system based on gesture recognition using Kinect. *TELKOMNIKA Indonesian Journal of Electrical Engineering*, 11(5):2605–2611, May 2013.

[5] L. Cruz, F. Lucio, and L. Velho. Kinect and RGBD images: Challenges and applications. In *XXV SIBGRAPI IEEE Confernce and Graphics, Patterns and Image Tutorials*, 2012.

[6] D. Di Paola, A. Milella, G. Cicirelli, and A. Distante. An autonomous mobile robotic system for surveillance of indoor environments. *International Journal of Advanced Robotic Systems*, 7(1), 2010.

[7] T. D'Orazio and C. Guaragnella. A graph-based signature generation for people re-identification in a multi-camera surveillance system. In *International Conference on Computer Vision Theory and Applications (VISAPP)*, pages 414–417, Rome, Italy, February 2012.

[8] T. D'Orazio and G. Cicirelli. People re-identification and tracking from multiple cameras: a review. In *IEEE International Conference on Image Processing (ICIP 2012)*, Orlando, Florida, Sept 2012.

[9] J. Fasola and M.J. Mataric. Using socially assistive human-robot interaction to motivate physical exercise for older adults. *Proceedings of the IEEE*, 100(8):2512–2526, August 2012.

[10] Tatsuya Fujii, Jae Hoon Lee, and Shingo Okamoto. Gesture recognition system for human-robot interaction and its application to robotic service task. In *Proc. of the International Multi-Conference of Engineers and Computer Scientists (IMECS)*, volume I, Hong Kong, March 2014.

[11] M. A. Goodrich and A. C. Schultz. Human-robot interaction: A survey. *Foundations and Trends in Human-Computer Interaction*, 1(3):203–275, 2007.

[12] Y. Gu, H. Do, Y. Ou, and W. Sheng. Human gesture recognition through a kinect sensor. In *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 1379–1384, 2012.

[13] T. Hachaj and M.R. Ogiela. Rule-based approach to recognizing human body poses and gestures in real time. *Multimedia Systems*, 20:81–99, 2013.

[14] I. I. Itauma, H. Kivrak, and H. Kose. Gesture imitation using machine learning techniques. In *Proc. of 20th IEEE Signal Processing and Communications Applications Conference*, Mugla, Turkey, April 2012.

[15] M. G. Jacob and J. P. Wachs. Context-based hand gesture reognition for the operating room. *Pattern Recognition Letters*, 36:196–203, 2014.

[16] K. Lai, J. Konrad, and P. Ishwar. A gesture-driven computer interface using kinect. In *IEEE Southwest Symposium on Image Analysis and Interpretation (SSIAI)*, pages 185–188, 2012.

[17] L. Miranda, T. Vieira, D. Martinez, T. Lewiner, A.W. Vieira, and M.F.M. Campos. Real-time gesture recognition from depth data through key poses learning and decision forests. In *25th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 268–275, 2012.

[18] J.Oh, T. Kim, and H. Hong. Using binary decision tree and multiclass svm for human gesture recognition. In *International Conference on Information Science and Applications (ICISA)*, pages 1–4, 2013.

[19] Kun Qian, Jie Niu, and Hong Yang. Developing a gesture based remote human-robot interaction system using kinect. *International Journal of Smart Home*, 7(4), July 2013.

[20] M. Sigalas, H. Baltzakis, and P. Trahanias. Gesture recognition based on arm tracking for human-robot interaction. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS)*, Taipei, Taiwan, October 2010.

[21] N. Tomatis, R. Philippsen, B. Jensen, K. O. Arras, G. Terrien, R. Piguet, and R. Y. Siegwart. Building a fully autonomous tour guide robot. In *Proc. of The 33rd International Symposium on Robotics (ISR)*. ETH-Zürich, Oct. 2002.

[22] D. Xu, Y.L. Chen, C.Lin, X. Kong, and X. Wu. Real time dynamic gesture recognition system based on depth perception for robot navigation. In *Proc. of IEEE International Conference on Robotics and Biometrics*, pages 689–694, 2012.

[23] Headquartes Department of the Army. Visual signals: Arm-and-hand signals for ground forces. Field Manual 21-60, Washington, DC, September 1987.