Image and Vision Computing 52 (2016) 56–72

Contents lists available at ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis

Review article

Recent trends in gesture recognition: how depth data has improved classical approaches☆

T. D'Orazio*, R. Marani, V. Renò, G. Cicirelli

ISSIA - C.N.R.Italy

ARTICLE INFO

Article history:
Received 25 May 2015
Received in revised form 5 May 2016
Accepted 6 May 2016
Available online 26 May 2016

Keywords:
Gesture recognition
RGB-D data
Features extraction
Classification approaches
On-line experiments

ABSTRACT

This paper analyzes with a new perspective the recent state of-the-art on gesture recognition approaches that exploit both RGB and depth data (RGB-D images). The most relevant papers have been analyzed to point out which features and classifiers best work with depth data, if these fundamentals are specifically designed to process RGB-D images and, above all, how depth information can improve gesture recognition beyond the limit of standard approaches based on solely color images. Papers have been deeply reviewed finding the relation between gesture complexity and features/methodologies suitability. Different types of gestures are discussed, focusing attention on the kind of datasets (public or private) used to compare results, in order to understand weather they provide a good representation of actual challenging problems, such as: gesture segmentation, idle gesture recognition, and length gesture invariance. Finally the paper discusses on the current open problems and highlights the future directions of research in the field of processing of RGB-D data for gesture recognition.

Contents

☆ This paper has been recommended for acceptance by Stanley Sclaroff.

* Corresponding author at: Institute of Intelligent Systems for Automation — National Research Council of Italy (CNR-ISSIA), via Amendola 122 D/O, 70126 Bari (BA), Italy.
E-mail address: dorazio@ba.issia.cnr.it (T. D'Orazio).

http://dx.doi.org/10.1016/j.imavis.2016.05.007
0262-8856/

1. Introduction

A gesture is defined as a form of non-verbal communication in which visible bodily actions communicate particular messages, either in place of, or in conjunction with speech. A gesture can include movements of hands, face, or other parts of the body. Gestures are the oldest means of human communication. Nowadays gestures are still important as people use them also in an unconscious way in everyday life, but they can be essential in many situations which involve communications in hazardous contexts. From the scientific point of view, gestures are used and then analyzed in several domains such as sign language recognition, vision-based augmented reality, smart surveillance, virtual environments, and human–computer interaction.

Different definitions of the term *gesture* have been provided in literature and sometimes this term has been interchangeably used as a synonym of the term *action*. In this paper, the definition provided in [1] has been used: a gesture is a physical movement or posture of hands, arms, face or body, made with the intent of conveying meaningful information. We point out the distinction between gestures, which are intentional movements of the body, and actions which are unconscious elementary movements of the body and can be used to understand human daily activities such as running, walking, skating, jumping, or, in a home environment, go to bed, get up, eat a meal, drink water, sit down, stand up, take off the jacket and put on the jacket and many others. According with this definition of gesture, in this review we consider and classify the papers that propose algorithms for gesture/action recognition where the gesture or action terms meet our definition of intentional movement or body posture for communication.

Gestures can be static, when the user assumes a certain pose or configuration, or dynamic with a pre-stroke, stroke and post-stroke phase, as pointed out in [2]. Some gestures also have both static and dynamic elements, as in sign language applications. The automatic recognition requires in the first case the characterization of the spatial disposition of the body parts performing the gesture, whereas in the second case it requires the observation of the sequence of movements generated by the human body.

Many good reviews on action recognition approaches summarized the researches carried out for the recognition of human movements such as walking, jumping, running, and so on [3,4]. Gesture recognition surveys have also been published [1,2,5], giving particular emphasis on hand gestures and facial expressions by the analysis of images acquired by conventional RGB cameras. Although intensity images contain rich information, they are very sensitive to lighting conditions, different point of views, camera resolutions, and cluttered backgrounds. As a consequence, tasks such as people segmentation, motion detection, or interest point detection can be affected by these factors and perform well only in very specific and limited situations. The recent introduction of low cost depth sensors, such as the widespread Microsoft Kinect sensor [6], allowed the development of new gesture recognition approaches. Depth images provide a 3D model of the scene which can be easily used to simplify many tasks such as people segmentation and tracking, body part recognition, motion estimation and so on. Recent reviews on human activity recognition and motion analysis from 3D data have been published in [7,8,9]. Human activities are characterized by sequences of atomic actions, by person–object interactions and by person–person interaction or group activities. A 3D gesture recognition survey, published in [10] provides recent trends on the general issues of sensing, recognition, and experimentation.

In this paper, we will review the literature which uses depth information for gesture recognition approaches from a different perspective. We will focus our attention on the main problems related to the application of gesture recognition approaches in real contexts: the identification of the beginning/ending parts of a gesture; the invariance to gesture length; the normalization with respect to different speeds during gesture executions. We will give particular attention to the most recent literature on gesture recognition which sees a number of publications on approaches based on depth data extracted by RGB-D sensors. The aim of this review is to highlight the main advantages of using depth data as additional information to traditional RGB data and to point out both technological and methodological limits which prevent real application of these approaches to commercial interfaces.

The rest of this review is organized as follows. Section 2 reviews the types of gestures and public datasets that have been used in literature. Then, the challenging problems related to the development of an automatic gesture recognition system will be considered. In particular, Section 3 describes the RGB-D features that better and distinctively characterize a specific movement or posture, setting them apart from similar items. In Section 4, Gesture Recognition is seen as a classification problem in which examples of gestures are used into supervised learning schemes (such as SVM or NN) to model the gestures and to address the recognition problem as a class association problem. In Section 5, the temporal Segmentation of dynamic gestures is approached as the task of determining, in a video sequence, the starting and ending frames of each gesture execution. Finally, after a discussion about the general topic of gesture recognition, Section 6 provides insights into open problems and future research directions. Section 7 reports final conclusions and remarks.

2. Types of gestures and datasets

According to the definition of gestures as intentional movements or posture with the intent to communicate a semantic message, different parts of the body can be involved in the communication. Intentional gestures can be executed by movements of hands, arms, head, torso, and full body. Static gestures are characterized only by postures or shapes of the involved body parts. For example, hand gestures can be characterized by the positions or orientations of the fingers (see Fig. 1). Similarly body gestures can be characterized by the relative positions of hands and legs with respect to the torso. Dynamic gestures are instead characterized by a movement which includes a starting and ending pose of the involved body parts (see the first and last frames of the gestures in Fig. 2).

The recent literature on gesture recognition proves the large interest in the use of public datasets as this allows the scientific community to compare different approaches. Table 1 summarizes the most used datasets. Some action recognition datasets are also cited in the Table as they contain, among the others, some actions which can be considered gestures such as hand waving, hand clapping, boxing, and so on. For this reason, in this review, some action recognition approaches will be also discussed as they can be applied to gesture recognition systems.

The gesture datasets consists of well segmented gestures, where each clip contains a different gesture or at least more repetitions of the same gesture performed by the same subject. One of the most active group in the publication of action/gesture dataset is the Microsoft Research thanks to the recent development and commercialization of their own 3D sensors (see the Microsoft Kinect and the Microsoft Kinect One). In [14] they propose different gesture and action datasets which contain both hand and body gestures/actions extracted also from daily activities. The MSRC-12 dataset [13] consists of sequences of gestures performed by 30 people and captured at a sample rate of 30 Hz. These gestures are categorized into two categories: iconic and metaphoric gestures. The iconic gestures directly correspond to real world actions, whereas the metaphoric ones represent abstract concepts. The Action 3D (MSR3D) dataset proposed by Li et al. [15] contains both depth maps and skeletal joint locations. It consists of depth map sequences with a resolution of 320 × 240 pixels recorded by using a depth sensor at 15 Hz. Twenty actions are performed by ten subjects two to three times for a total of 567 depth map sequences. Many of the actions provided by these Microsoft datasets meet our definition of gestures, while others can be considered as part of daily activities: high arm wave, horizontal arm wave, hammer, catch, tennis swing, forward punch, high throw, draw X, draw tick, tennis serve, draw circle, hand clap, two hand wave, side boxing, golf swing, side boxing bend, forward kick, side kick, jogging, and pick up and throw.

The ChaLearn challenge [23] provides a set of datasets for gesture and sign language recognition from video, mostly focusing on hand and arm gestures. The recent dataset introduced in [17] consists of M = 20 Italian cultural or anthropological signs, recorded by a Kinect sensor. These data include multiple modalities: RGB, depth maps, skeleton models, and audio. The dataset contains three parts: training (7754 gestures), validation (3362 gestures), and test data (2742 gestures). The Northwestern-UCLA Multiview 3D event dataset [18] contains RGB, depth and human skeleton data captured simultaneously by three Kinect cameras. This dataset includes 10 action categories: pick up with one hand, pick up with two hands, drop trash, walk around, sit down, stand up, donning, doffing, throw, and carry. Each action is performed by 10 actors. This dataset contains data taken from a variety of viewpoints.

In order to provide researchers with an inclusive testbed to develop and to benchmark new algorithms across multiple modalities under known capture conditions in various research domains, a new multimodal dataset has been introduced. The Berkeley Multimodal Human Action Database (MHAD) [22] consists of temporally synchronized and geometrically calibrated data acquired with an optical motion capture system, multibaseline stereo cameras from multiple views, depth sensors, accelerometers and microphones.

A data set of the Australian Sign Language (ASL) is available in [21]. The source of the data is the raw measurement from a Nintendo PowerGlove. It contains 95 different gesture classes and 6650 gesture samples. Although this dataset is not obtained with depth cameras, it returns 3D points of the hand surface which can be considered an extension of depth images of the hands performing the gestures.

All the considered datasets contain dynamic gestures as indicated in the last column of the Table 1, except the one in [19] which used the Kinect ability to segment hands and collect the ten digit hand gestures executed by ten persons for 10 repetitions, and the dataset in [21], which contains 3D coordinates of hand postures performed by 5 different persons.

3. Feature extraction

Selecting features is crucial for gesture recognition, since body gestures are very rich in shape and motion variations. According to [24], in the context of gesture recognition, the methods can be classified into two main categories: *Depth-map based methods* and *Skeleton based methods*. The former ones are primarily based on features extracted directly from the space volume. The latter ones, instead, use features such as angles, coordinates, and other combinations, extracted by skeletons which synthetically represent the human silhouette. In the past, skeletons were already used by the computer vision community [25], even if their extraction based on morphological operation was computationally expensive and often not effective due to the errors in the silhouette segmentation of conventional cameras. In the last years, thanks to the advent of low cost RGB-D sensors, skeleton-based approaches became very popular. The software frameworks, diffused together with the depth sensors, provide the 3D positions of human skeleton joints in real time, solving at the same time many problems of people segmentation from RGB images, such as the presence of shadows, light reflection, similarity of colors between foreground and background objects which can greatly compromise the people silhouette segmentation.

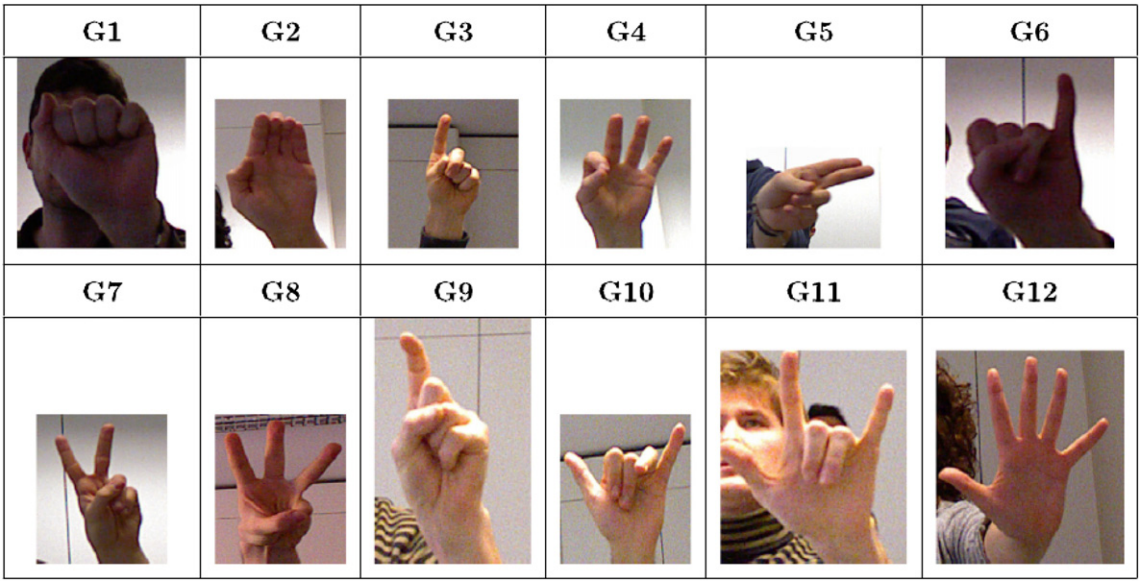

Fig. 1. Static hand gestures from the American Sign Language (ASL) contained in the database that has been acquired for the experimental results in [11].

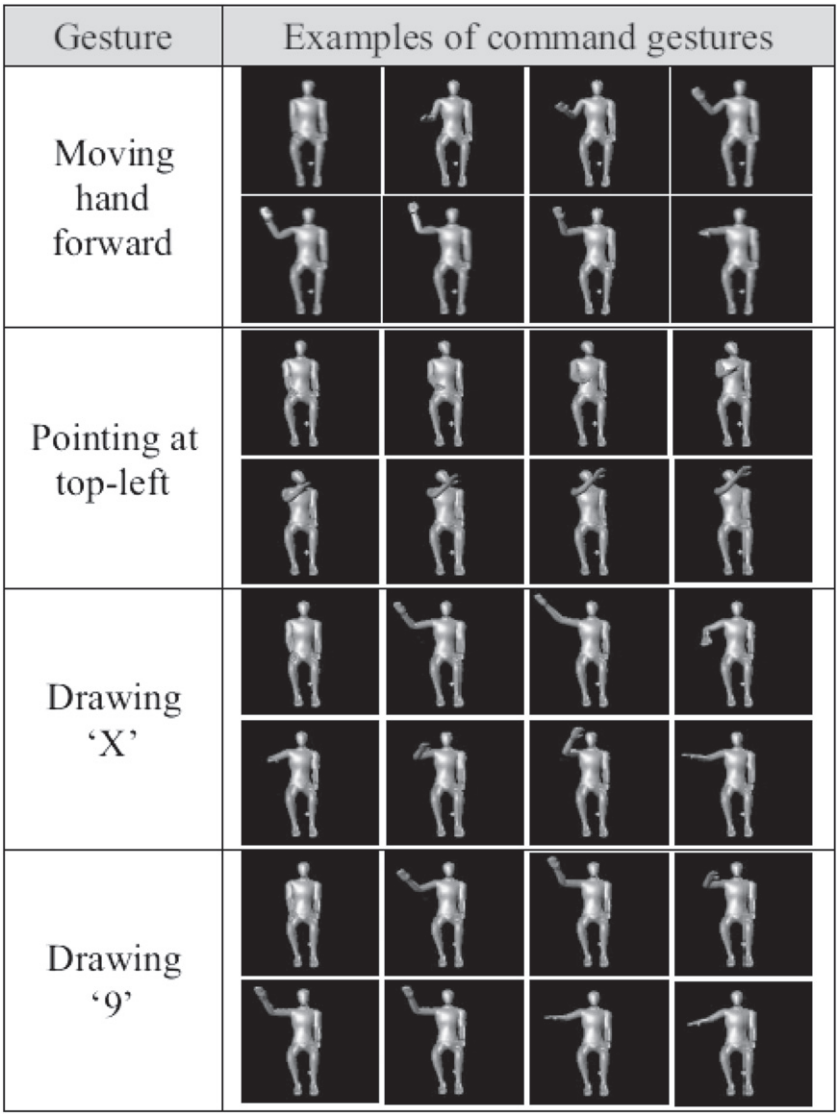

Fig. 2. Examples of command gestures executed by the whole body and provided by the database in [12].

However, these skeleton extraction procedures work well when people are well separated from background and assume postures such that the arts are clearly visible and separated from the torso. If this is not the case, they can introduce different errors for wrong skeleton joints detections or noise in the features extraction.

3.1. Depth-map based features

Depth data processing (DDP) provides a rich informative description of the scene, especially if compared to the analysis of a bi-dimensional image because it adds the knowledge of the third coordinate z to the well known (x,y) frame representation. Many applications exploit this information in order to:

- extract the part of the body which is interested in the gesture execution;
- extract three-dimensional shapes based on pose evaluation and use these shapes to match predefined models;
- recognize the gesture without processing the full pose, but applying machine learning techniques to some features extracted from the depth information.

Stereo cameras have been used to generate *depth silhouettes* that improve the binary model of standard silhouettes by filling the pixels with range information [26]. In this way depth silhouettes are able to register complex poses without tracking feature points. Also Range cameras have been considered to fuse depth and intensity information producing 3D point clouds [27]. The intensity images are used to select a Region Of Interest (ROI) for motion detection in the range data. The point clouds are therefore represented using shape context descriptors such as spherical histograms. View invariant gesture recognition is assured by representing gestures as sequences of 3D motion primitives. Also three-dimensional shape descriptors extracted by ToF Cameras can be used to represent hands in 3D views and to guarantee invariant scale and rotation matchings [28].

The recent progress in depth sensors has generated a new level of excitement in gesture recognition. With the depth information, automatic tracking systems which provide 3D shape and silhouette of people and body parts [29], have been developed. Sequences of depth maps can be used to build spatio-temporal representations in a four dimensional space which can be partitioned in space-time cells for action interpretation, as shown in Fig. 3 [30,31]. In the case of hand gestures, depth data can be used to extract the hand region as the

Table 1
An overview of the action/gesture datasets used in the literature.

Dataset name	Capture device	Cardinality	Gesture type
MSRC-12 Kinect gesture [13]	Kinect	12 gestures performed by 30 subjects	Whole body 30 Hz (dynamic)
MSRGesture3D [14]	Kinect	12 gestures performed by 10 subjects 2–3 times	Hand (dynamic)
MSRDailyActivity3D [14]	Kinect	16 activities performed by 10 subjects 2 times	Whole body (dynamic)
MSRAction3D [15]	Kinect	20 action types performed by 10 subjects 2–3 times	Whole body (dynamic)
MSRAction pairs [14]	Kinect	Pairs of similar actions selcted among MSR 3D datasets	Whole body (dynamic)
Chalearn gesture dataset (CGD2011) [16]	Kinect	50,000 performed by a single user in front of a fixed camera	Whole body (dynamic)
Chalearn multimodal gesture recognition [17]	Kinect	7754 training gestures 3362 validation gestures 2742 test gestures	Whole body (dynamic)
Northwestern-UCLA multiview action 3D dataset [18]	3 Kinect cameras	10 action categories performed by 10 subjects Multiple viewpoints	Whole body (dynamic)
NTU hand gesture recognition dataset [19]	Kinect	10 gestures performed by 10 subjects	Hand (static)
Sheffield Kinect gesture dataset (SKIG) [20]	Kinect	2160 gestures performed by 6 subjects 10 categories of gestures 3 different backgrounds 3 poses 2 illumination setups	Hand (dynamic)
Australian Sign Language dataset [21]	Nintendo PowerGlove	95 signs 6650 sign samples	Hand (static)
MHAD dataset [22]	Stereo, Kinect, accelerometers, microphones	11 actions performed by 12 subjects 5 repetitions	Whole body (dynamic)

area of the body closer to the camera [32,33,34,35] and use the geometric size to identify the wrist area and remove the points that lie beyond it [36]. The knowledge of the depth allows also some normalization steps performing scale and rotation transformations to have view invariance. Depth data and derived curvature features can be used to divide the hand regions in different parts (fingers, palms and wrists) and extract a set of feature descriptors which characterize the shape and the pose of the hand gesture [11]. Depth data allow an initial segmentation to extract SIFT features that can be hierarchically quantized in a vocabulary tree to recognize the hands performing different gestures such as drawing the numbers from "one" to "ten" in [37]. Also, the use of synthetic 3D hand model has been evaluated

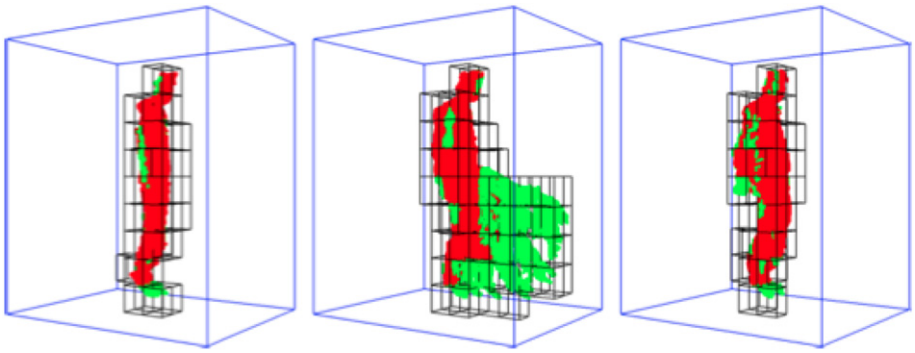

Fig. 3. Space time cells of a depth sequence in [30].

to perform pixel classification and assign each pixel to a hand part, such that each skeleton joint is at the center of one of the labelled parts [38]. 3D point clouds which contain points in the 3D real-world coordinate system are used in [39] to represent the external surface of human body. A robust feature, the Body Surface Context (BSC), is generated by describing the distribution of relative locations of the neighbors for a reference point in the point cloud in a compact and descriptive way. Finally, multi camera systems are able to solve occlusion problems which reduce the recognition performances of single camera approaches [40]; salient points on the fingers can be used to estimate the hand pose in challenging situations containing hands and objects in action.

3.2. Depth-color features

Joining color and depth data in more complex representations can certainly boost the discriminating capabilities of the features.

Although there exist some works using Spatio-Temporal Interest Point (STIP) features for depth-based action recognition, only very limited types of STIP features have been investigated. STIP detectors can be applied separately on RGB and depth volumes to define descriptors which are concatenated to model sets of gestures [41],[42], or can be applied only on RGB channels and then combined with depth map based descriptors [43]. Features that contain both depth and color in a compact representation are also explored. In [44], 4D spatiotemporal cuboids are represented with a feature descriptor of intensity and depth gradients. STIPs from depth videos (DSTIPs) are defined in [45] to describe local 3D depth cuboids using a Depth Cuboid Similarity Feature (DCSF). STIP detectors are extended in Depth Layered Multi-channel STIPs in [46]. Also Motion History Images (MHI) can be generalized to include depth changing directions in a combined feature representation as 3D-MHIs [46]. Nevertheless, the application of interest point detectors to depth data with the actual sensor resolutions has still big limitations. In [47] a comprehensive evaluation of STIP features for action recognition in 3D has proved that noisy depth data and background greatly influence interest point detection which may not perform well in situations without enough motion.

The concept of superpixel, as aggregation of adjacent points of an image that hold a common property, is extended to incorporate both color texture and depth map in a new representation for hand gesture recognition in [48]. Uniform pixels for color, depth and closeness are aggregated by a clustering algorithm to generate equally sized superpixels which retain in a compact way as much information as possible.

3.3. Skeleton-based features

The study of skeleton-based gesture dates back to the early work by Johansson [49], which demonstrates that a large set of gestures and actions can be recognized solely from the joint positions [24]. This concept has been extensively explored since then. In contrast to the depth map-based methods, the majority of the skeleton-based methods model temporal dynamics explicitly. This is because, the natural correspondence of skeleton joints is known across time, while lower level depth features can be hard to match between frames or could disappear entirely from frame to frame. For this reason in the past, many researchers tried to create a 3D Human Model in order to detect some information about joints. In [50], the authors built a 3D articulated human body model which consists of 17 body segments. A linear combination of 2D depth image and 3D human model prototypes was used to reconstruct the 3D human body. The features used to recognize the gestures were the three angle values of selected body points located at joints of left wrist, left elbow, left shoulder, etc. However, the extraction of human skeleton from 2D or 3D images is a complex task that requires many computational resources and strictly depends on the noise in the images. With the recent distribution of Microsoft Kinect sensors on the market, skeleton based approaches have become much more popular [51]. Together with the sensors, some software platforms are available that detect and track one or more persons in the scene and extract the corresponding human skeleton in real-time. In particular, several platforms have been largely utilized such as the official SDK for Kinect [52], and others open source such as OpenNI [53], Libfreenect [54] or ViiM-SDK [55]. The direct availability of real-time information about joint coordinates and orientations has provided a great impetus to research and many papers have been published in the last years. In this review we have grouped the approaches according to the extracted features: coordinates, angles and quaternions of skeleton joints.

3.3.1. Position features

Joint coordinates are immediately available by the Kinect software platforms and in many kind of gestures are discriminant enough to guarantee recognition. Coordinates are generally provided with respect to the camera reference system, therefore, they depend on different factors such as: person's height, arm length, distance and orientation of the gesturer with respect to the camera (see Fig. 4) [56]. Each of these factors may impact the gesture model in a different way, requiring normalization steps to handle variations in different gesture executions. The number of total skeleton joints varies from 15 to 20 to represent the whole human body, according to the selected platform. However, the number of joints necessary to characterize the gestures depends on the gesture difficulty, the involved parts of the body and the similarities among the considered gestures.

In [57], seven upper body joints out of the total 20 provided by the Kinect are kept to recognize 8 aircraft gestures used by the military air force. Each coordinate is normalized with respect to its maximum and minimum values. In [58], all the 20 joints are considered to recognize three simple body gestures and a Z-score normalization is applied to deal with parameters of different units and scales of body-joint positions. A sequence of coordinates of the person's right hand is stored in [59], to recognize a set of six gestures, that appear in interaction with waiters (such as asking for a bill, cancelling an order, and so on). A simple geometric transformation is applied in [60] to the hand coordinates to set the reference system centered on the human torso, instead of the default sensor-centered reference frame. This transformation provides invariance to the starting point of a physical gesture. In other words, the user can perform the hand gestures (push, grasp and tap) at any distance or angle from the sensors, and these gestures are always measured with regards to his torso position. Also in [61,62], only few joints (hand and elbow) are considered significant for the hand gesture recognition. In each frame, the 3D distances of these joints from the spine joint, which serves as a

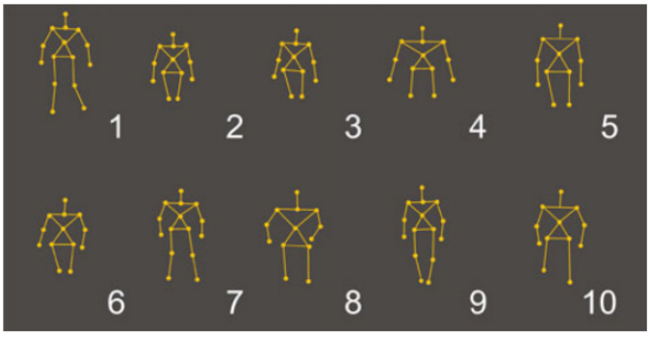

Fig. 4. Different sizes and proportions of detected skeletons build of body joints of different participants in the experiment in [56].

reference point, are computed to be invariant to the user's height and camera distances. A mixture of joint coordinates and angles is used in [56], to recognize eight different gestures which are characterized by either static poses or dynamic variations of joint positions. In [63] the neck joint of the skeletal model is employed as the origin of coordinates (OC). The remaining joints coordinates are computed with respect to the OC. This transformation allows to relate pose models that are at different depths, being invariant to translation, scale, and tolerant to physical differences of subjects. In a similar way, in [64], the feature vectors are firstly normalized with respect to the distance between the left and the right shoulders to account for the variations due to people different sizes. Then, a second normalization subtracts the shoulder center from all the elements to account for cases where the user is not in the center of the depth image. However, the joints position data is considered unstable especially for tasks that require a precise localization of the body parts. In [65] a Kalman filter is applied to sequences of joint coordinates to estimate the hand position, since the application requires a precise localization of hand movement in a human manipulation interface for robot teleoperation. In [66], in order to make the skeletons scale invariant, the hip center joint is first placed at the origin of the coordinate system. Then, a skeleton template is taken as a reference and all the other skeletons are normalized such that their body part lengths are equal to the corresponding lengths of the reference skeleton.

A hierarchial dynamic framework is proposed in [67] that first extracts high level skeletal joints features and then uses the learned representation for estimating emission probability to infer action sequences. The proposed learning approach, in which all the knowledge in the model comes from the data, avoid sophisticated preprocessing or dimensionality reduction of all the possible input features.

3.3.2. Orientation features

Angular information between joint vectors have the great advantage of maximizing the invariance of the skeletal representation with respect to the camera position and of reducing the dimensionality of the search space while retaining the character of the motion.

Angles between specific pairs of direction vectors, are computed to obtain the corresponding joint angles in [68,69,70,71]. For instance, the direction vector of the lower arm is calculated between elbow and wrist position, while the direction vector of the higher arm is calculated between shoulder and elbow position.

In [72], a novel angular skeleton representation is used to map the skeleton motion data into a smaller set of features (see Fig. 5 for details). The aim is to reduce the overall entropy of the signal, to remove the dependence on camera position and to avoid unstable parameter configurations. The approach is to fit the full torso with a single rigid frame of reference, and to use this frame to represent the remainder of the human skeleton in a relative manner. The same representation is used in [73] to recognize key poses which characterize different gestures. The set of joint angles of [72] represented only by the inclination and azimuth terms are transformed in [74] in high order features. The time series data is converted into a distance matrix of joint angle similarities between each of the angles along the entire action time series. Different distance measures are applied and these new feature representations are shown to outperform the initial feature set.

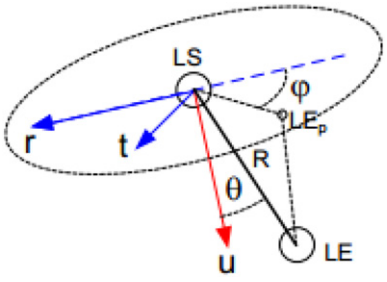

Fig. 5. A spherical coordinate system centered at the left shoulder (LS) used in [72] to represent the left elbow (LE) with a radius R, inclination θ and azimuth ϕ angles.

The Euler angles have been largely used to describe the orientation of a rigid body in a 3-dimensional Euclidean space. The Kinect software platforms provide directly the yaw, pitch and roll angles of all joints of the skeleton except the ending ones. In some papers these angles are directly used to describe the 3D rotation over the principal axes as in [75]. In [76], experiments demonstrate that yaw and roll angles of the left elbow and yaw and pitch angles of the left shoulder are sufficient to recognize five different gestures executed with one arm.

Another way to model orientation information is by means of unit quaternions which represent a system of numbers that extends the complex numbers. The quaternion is described in 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. Compared to 3-by-3 rotational matrices, quaternions are also more compact as only 4 storage units, instead of 9, are required. These properties of quaternions make their use favorable for describing rotations compressed without loss.

In [77] the quaternion of all joints, provided by the Kinect software platform, are used to recognize 6 different actions which reasonably cover the various movements of arms, legs, torso and their combinations. When the gesture movement involves only some parts of the body, different selections can be done [78]. In [79] the quaternions of the shoulder and elbow right nodes are employed as they allow the recognition of 10 gestures executed with one arm. The quaternions provided by the Kinect software platforms are generally represented in the sensor reference system. Therefore, the user position with respect to the sensor can greatly affect the gesture recognition. In order to resolve this dependency, the methods have to be view-invariant. One possible solution is the one proposed in [80]. The torso joint is taken as the origin of the reference system and the X-axis is defined as the line from the left hip to the right hip, the Y-axis as the average of the left and right leg lines and the Z-axis as the one perpendicular to the plane determined by the X and Y axes. The relative orientation of each joint is calculated by abstracting the torso orientation in the starting frame using: $Q_{new} = Q_{old} * Q'_{torso}$ in which Q' is the conjugate of Q. After this step, the quaternions are all reported in the new coordinate system. Similarly, in [81], the invariance of the quaternions with respect to the position and orientation of the gesturer is guaranteed by a dynamic transformation of the coordinate system. In particular, the coordinate axes are rotated with respect to Y' to match the orientation of X' with the shoulder orientation (see Fig. 6 for details).

3.4. Multimodal features

A concluding remarks should be done on the recent use of features coming from different modalities such as color, depth, skeleton, or audio features [82,83].

In [84], audio and skeleton joints are used to demonstrate that complementary information between two modalities improves the recognition rate on a public dataset of 20 signs [17]. In the approaches presented in [85,86], skeleton joint information, depth and RGB images, are the multimodal input observations. In [85] instead of engineering features which improve performance on the specific gesture dataset, a learning phase of the most significant features in each modality is proposed followed by a gradual fusion of modalities from the strongest to weakest cross-modality structure.

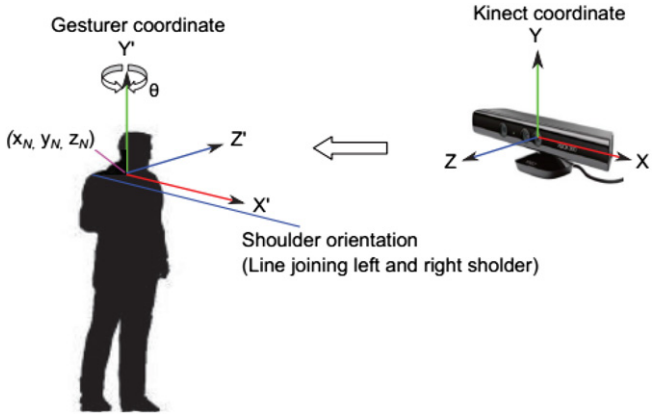

Fig. 6. Translation and rotation of the coordinate system. The origin of the coordinate system utilized is translated to the neck of the gesturer (x_N, y_N, z_N) and rotated in accordance with the rotation of the shoulder (aligning X' with shoulder orientation), by an angle θ in [81].

The results show that skeleton features are very good for segmentation but make more mistakes during recognition, on the other hand RGB-D features allow for reliable recognition but are not as good for segmentation. Then, they demonstrate that multiple-channel fusions in combination with the temporal modelling outperform individual modules. In [86], for each type of visual source several consecutive frames are considered and, after a preprocessing step based on multiple stages of convolution and sub-sampling, spatial-temporal features are extracted to capture motion information by all the channels. In [87], data modalities include intensity and depth video, as well as articulated pose information extracted from depth maps. Different data channels are used to decompose each gesture at multiple scales to provide context for upper-body body motion and more fine-grained hand/finger articulation.

4. Classification methods

After the feature extraction step during which features relevant for the gesture typology are selected, another important step has to be carried out: the model generation for the gesture recognition task. This problem is largely considered as a supervised classification process, supposing that a human expert determines the number of classes and provides the sets of samples that belong to the known classes. During a training phase, the sets of samples are used to generate the gesture models which can thereafter be applied during the actual tests for the recognition phase. Different methods can be used to generate gesture models.

4.1. Hidden Markov models (HMM)

Hidden Markov models are statistical models, especially known for their application in temporal pattern recognition, in which the systems being modelled are assumed to be Markov processes with unobserved (hidden) states. They are a very common choice for gesture recognition as they model sequential data over time [33,76,88,89,90,91,92]. HMMs are characterized by the following components $\lambda = \{\pi, A, B, N, M\}$: N is the number of states in the model, M is the number of distinct observation symbols per state, π is the initial state distribution, A is the observation symbol probability distribution and B is the state transition probability distribution. The majority of the methods uses a single N-states HMM for each gesture. The choice of the number of states N depends on the complexity of the process to be modelled: a high number of states could generate a model that is too specific; on the other side with a small number of states there is the possibility of having indistinguishable gestures. No general rule exists in literature to solve this problem. A proper tradeoff is generally found by experiments: a number of states between 5 and 8 has been demonstrated to be effective. Fully connected HMMs or partially connected topologies can be selected. Fully connected HMMs (Ergodic model) can model a wider range of processes taking advantage of a higher number of parameters, while partially connected ones are simpler and require a lighter computational load. Among the partially connected models, the left right ones are the most used for gesture recognition as any state can only iterate over itself or to the next state, thus representing a chain that is processed from one side to the other [88,93,94].

In the training phase for each HMM a set of sequences of feature observations O is provided and the Baum–Welch or Viterbi algorithms [95] are usually used to found the optimum model parameter vector λ that maximizes $P(O|\lambda)$. The probability distribution of observation symbols can be discrete or continuous. Most of the works which apply HMMs uses discrete symbols of observations. Therefore the features which characterize the gesture observations are quantized by using different quantization schemes. In [76] a K-means clustering is used to convert the feature vectors (joint angles) into the observable symbols for HMMs. The orientation of the hand centroid point projection in the image plane is transformed, in [88,96], in a chain code from 1 to 8 by dividing the coordinate system into eight equal portions. The observations provided to the HMM are the orientation chain codes of each observed gesture. A similar approach is used in [91] with a uniform quantization of the orientations in 12 sectors (every $\pi/6$ a direction is quantified). In the case of continuous observations, instead, $P(O|\lambda)$ is commonly represented as a Gaussian or mixture of M Gaussians [97]. The skeleton coordinates are transformed in [92] into feature sequences by considering the features as observations of Gaussian distributions. Therefore, the parameters of the GMM can be estimated using the Expectation Maximization algorithm.

After the HMM training phase, given a sequence of observations O, the $argmax\{P(O|\lambda)\}$ is computed to determine the corresponding gesture. This problem is known as the observation evaluation problem and can be solved using the Forward–Backward procedure [95]. Many approaches use a threshold method to discern whether a non gestural movement belong to a gesture even if it provides always a maximum. When a threshold method is insufficient, the authors of [90] propose an additional HMM, trained to provide an adaptive threshold, to discriminate between gestures and non-gestures.

Cascade of more HMMs can be used to make a more robust estimation of gestures in a coarse-to-fine approach. In [89] two HMMs are used to calculate the hand pointing direction. The first stage of HMMs takes the estimated hand position and maps it to a more accurate position by modelling the kinematic characteristics of the pointing finger. The resulting 3D coordinates are used as input for the second stage of HMMs that discriminates pointing gestures among the others. Finally, the pointing direction is estimated for the pointing state. In order to assign the observation to the HMM, the surface of the hemisphere in which the pointing gesture is performed, is divided into patches of 20° from left-to-right and top-to-bottom.

Similar to HMM, action graphs are another statistical approach based on a set of action classes, a set of key postures, an observation likelihood model, and a transition probability matrix between key postures for any given action class. Compared to HMM, action graphs have the advantage that require less training data and allow different actions to share the states [36]. In [30], the training procedure for an action graph consists of learning both the key poses and the transition matrix (see Fig. 7). The key poses, obtained by a K-means classification, are the nodes of the action graph, while the $P(j \mid i)$ is

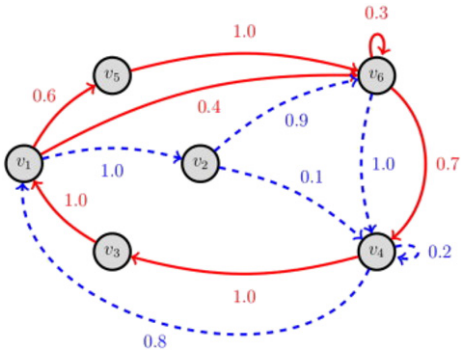

Fig. 7. Example of an action graph where two distinct actions are modelled with the same set of salient postures in [30]. The different graphs are identified by the transitions in solid and dotted lines. States V_1, V_4 and V_6 are shared by the two actions, and each action is defined by more than one path and the loop in some states indicates the probability that the duration of a posture has varied between different performances of the same action.

the transition probability evaluated as the ratio between the number of transitions from key pose *i* to key pose *j* in the training data over the number of repetitions of the state *i* in the training set.

4.2. Support vector machine (SVM)

Support vector machines are well known discriminative classifiers largely used to recognize patterns formally defined by a separating hyperplane. In other words, given labelled training data (as the result of a supervised learning), the algorithm outputs an optimal hyperplane which categorizes new examples.

Generally multiclass SVM approaches are used for gesture recognition tasks [31,57,71,73,78,98], as they reduce the classification problem into multiple binary classifications either by applying a one-versus-all (OVA-SVM) strategy (distinguishes between one class and the rest of classes, for a total of N classifiers for N classes) or a one-versus-one (OVO-SVM) strategy (between every pair of classes for a total of $N \times (N-1)/2$ classifiers for N classes). Classification of new instances in the case of OVA-SVM can be done by using a winner-takes-all strategy [73], in which the classifier with the highest output function assigns the class. For the OVO-SVM approach, classification can be done by a max-wins voting strategy, in which every classifier assigns the instance to one of the two classes, then the vote for the assigned class is increased by one, and finally the class with the maximum vote determines the instance classification [29,57,78,99,100]. The OVO-SVM strategy requires $O(N^2)$ classifiers whereas the OVA-SVM only $O(N)$; nevertheless, the first solution is generally preferred as it provides more effective classifiers in terms of final recognition performances.

The choice of linear or nonlinear SVMs depends on the complexity of the classification problem. In many contexts, linear classifiers are flexible enough to separate binary classes. However, if datasets are not linearly separable, the introduction of nonlinear kernel functions is necessary in order to allow the fit of a maximum margin hyperplane in a transformed feature space where linear classifiers are applicable. The most commonly used kernels are the polynomial kernels and Gaussian Radial Basis Functions (RBF) [78,100]. Linear and RBF kernels are compared in [57] to recognize 8 gestures by using the normalized coordinates of the arms joints. The choice of linear-SVM over RBF-SVM is justified by two reasons: complexity of finding the best parameters and also the general guidelines which state that the accuracy of a linear kernel is better than RBF when the data dimension is much higher than the sample size. A multiple kernel learning (MKL) algorithm is proposed in [99] in conjunction with a multiclass SVM to enhance the discrimination power of the classifier when multiple features result incompatible. The fusion of 3D shape features and 3D motion features is carried out at the kernel level to recognize complex activities.

Approaches which apply the OVA-SVM strategy can use the sign of the output values to establish if the new instance belongs to the trained class [73]: if all the classifiers return negative values, then the query does not belong to any trained gesture, and could be associated to an intermediate gesture. On the contrary, the OVO-SVM approaches do not solve the problem of meaningless gestures, unless, as in [29], a further class is added in order to recognize examples of movements which can be considered preparatory or idle before that significant gestures are performed.

4.3. Artificial neural network (ANN)

Artificial neural networks (also known as Neural Networks NNs) represent another alternative methodology to solve classification problems in the context of gesture recognition [101]. ANNs emulate the parallel architecture of neurons in the human brain. They consist of a set of adaptive weights, i.e. numerical parameters, that are tuned by a learning algorithm, and are capable of approximating non-linear functions of their inputs. Different parameters define the ANN topologies: the interconnections between different layers of neurons, the learning process, and the activation function of each neuron. Feed-forward NNs are models in which the information moves in the forward direction, i.e. from the input nodes data goes through the hidden nodes (arranged in one or more layers) and then to the output nodes. Recurrent NNs are models in which loops occur because of feedback connections. Feed-forward NNs are memory-less as their response to the input is independent of the previous state. Recurrent NNS, because of the feedback paths, modify repeatedly the neuron outputs, leading the network to enter a new state.

The choice of the network topology, the number of nodes/layers and the node activation functions depend on the problem complexity and can be fixed by using iterative processes which run until the optimal parameters are found. The input nodes are strictly related to the number of elements in the feature vectors. For static gestures, the number of input nodes is the same of the features extracted from each instance, while for dynamic gestures, which spread over a number of observations, the number of input nodes is multiplied by the number of observed frames.

The number of output nodes is related to the number of gestures. When few gestures have to be recognized, multi class feed-forward NNs are preferred as in [58], where three gestures are considered, or as in [102] where 5 static hand gestures representing the vowels of the alphabet are selected. In [103] a coarse to fine approach with feed-forward multilayer NNs is proposed: the first ANN with two output nodes performs a preliminary recognition of the gesture side (with left or right arm) and in sequence two NNs with four output nodes recognize the gesture typology. A different approach is used in [79], where a cascade of 10 different binary NN classifiers are trained to recognize ten different gestures. Each NN is trained with examples of one gesture against all the remaining gestures, providing a single output value spanning in the range [0, 1]. The class association is performed by selecting the NN which provides the maximum output value over a fixed threshold.

A Recurrent NN is used in [104] both for modelling large-scale temporal dependencies of features (since gestures are considered as a set of characteristic dynamic poses) and for combining data coming from different modalities (depth, skeleton and audio features).

In [105], Extreme Learning Machines (ELM) are applied to classify motion on a frame level and make the final classification decision considering the whole sequence. ELMs are single-hidden layer feed-forward neural networks whose input weights and first hidden layer biases do not need to be learned but are assigned randomly: it

makes the learning and classification extremely fast and particularly suitable for on line applications.

Deep learning architectures have recently been proposed specifically for gesture recognition from multi-modal data. A Recurrent NN is used in [104] both for modelling large-scale temporal dependencies of features (since gestures are considered as a set of characteristic dynamic poses) and for combining data coming from different modalities (depth, skeleton and audio features). In [87] a multi-scale deep neural model is proposed to combine single-scale paths connected in a parallel way. Each path independently learns a representation and performs gesture classification at its own temporal scale given input from RGB-D video and articulated pose descriptors.

4.4. Distance-based approaches

Distance-based approaches, starting from the assumption that the features characterizing the models are well separated, apply distance metrics to measure the similarity between samples and gesture models. These methods have to solve the problem of variable lengths of the sequences in order to apply any metric for comparisons. Several solutions have been proposed either transforming the features in a different space, or using ad hoc procedure to align the sequences.

A filling algorithm is used to align initially all the sequences of joint coordinates of different-sized gestures [106]. Then, the resulting sequences are provided to an eigenspace-based method which generates a new subset of features whose centroids are representative of the different gesture classes. The gesture recognition is performed by projecting the sample sequence in the trained eigenspace and determining its minimum distances with the gesture class centroids. Also 3D contour models can be used to characterize static gestures of the hands [107] and simplify the computational complexity of gesture matching. In this case, the contour is coded into strings and the correspondent sample gesture can be found by the nearest neighbor method. The Earth Movers Distance (EMD) is a measure of the distance between two probability distributions over a region. It can be used as a metric to recognize hand gestures as it is robust to local distortions, articulation, orientation and scale changes [34], [48]. This distance is based on a part-based representation (such as superpixels in [48]) which represents a hand shape and its time series curve, as a signature, with each finger part as a cluster, and enables the computation of global features. The EMD adds penalties on empty holes to alleviate the partial match on global features.

A new distance measure is proposed in [108] based on conditional distance and warp vectors. As models and queries are both conditioned on an anchor sequence, the distance measure takes into account how a particular model varies with respect to every other model in the model base (see Fig. 8).

4.5. Dynamic time warping (DTW)

Dymamic time warping is the most used technique to find the optimal alignment of two signals and, together with a distance metrics, it can be used to solve a classification problem. The algorithm consists of the following steps:

1. The average length L_{avg} of a training samples is calculated.
2. An equally spaced vector is calculated using the length of the signal to be warped.
3. The signal values corresponding to the equally spaced points are calculated by linear interpolation of the signal to be warped. The warped signal has a length equal to L_{avg}.

In order to use DTW for gesture recognition a target sample of the feature sequence for each gesture is necessary, then the algorithm makes the test sequence uniform in time with the reference ones, and invariant to the gesture speed and size. The similarity between the test sequence with each class is done in [75,81,109] by measuring the Euclidean distance and providing the corresponding probability of the test sequence of being an instance of the considered class. The maximum value of probability is calculated to classify a gesture.

If the sequences are multi-dimensional (as it happens when different features are considered to characterize the gestures), using an Euclidean distance gives equal importance to all dimensions. In [64], a weighted DTW method is proposed to weight, in a different way, features that are relevant for each body activity. The relevance is defined as the contribution of a feature to the motion pattern of that gesture class. So, the weights vary to maximize between-classes variance and to minimize within-class variance. In [110] the set of gestures are preliminary characterized by sequences of key poses; then a DTW distance between two sequences of key poses is defined by combining the Euclidean distance between couples of key poses in all the possible alignments of the test and reference sequences.

When there is a large intra class variability, the description of a certain gesture can have dramatic changes. In this case, DTW fails as it compares each sequence of features with the reference pattern. In [41], a probability based DTW is introduced to handle intra-class variability. Different gesture samples belonging to the same category are aligned with the median length sequence using classical DTW with Euclidean distance. The feature vectors extracted by the set of warped sequences are then modelled by means of a G-component Gaussian Mixture Model (GMM) (see Fig. 9). The resulting model is composed by a set of GMMs, which cannot be compared with a test sequence using the classical Euclidean distance. For this reason, a soft-distance is considered to evaluate the probability of a point of belonging to each G-component in the GMM.

4.6. Rule-based approaches

In many cases, when gestures are simple or quite distant, the interpretation of body movements is natural and there is no need to use complex mathematical and statistical approaches for the correct recognition. Moreover, all the approaches which require a training phase for the model generation impose the need of large training sets (often manually tuned) and require additional intensive training every time a new gesture has to be recognized. In many of these contexts, rule-based approaches are discriminant enough to unambiguously classify sets of gestures.

When real-life natural gestures are performed, an intuitive way of writing a set of rules can be used to generate a reasoning module and allow these rules to be interpreted on line. In [56] all the rules of the knowledge base, describing 9 gestures (performed with one or both arms), are organized in scripts having the form of text files. Every time new portions of data arrive from the feature extraction library, a forward chaining reasoning, like a classical expert system, performs inference engine. Similarly, the use of a preliminary representation of gestures as sequences of *key poses* linked by rules can allow the efficient recognition of gestures. In [73], the set of gestures is constrained to be represented as sequences of key poses so that the sequence defining a gesture cannot be extended to a longer sequence that represents another gesture. In this way different sequences can characterize different user performances for the same gesture. This gesture representation is provided to a decision forest classifier, where nodes represent key poses and leaves are associated to gestures. The preliminary key pose recognition is carried out by multi-class SVM. Also in [111] random decision forests (RDF) are used for the recognition of hand gesture in the ASL dataset. First, a realistic 3D hand model is used to represent the hand with 21 different parts. Then, a RDF is trained on synthetic depth images

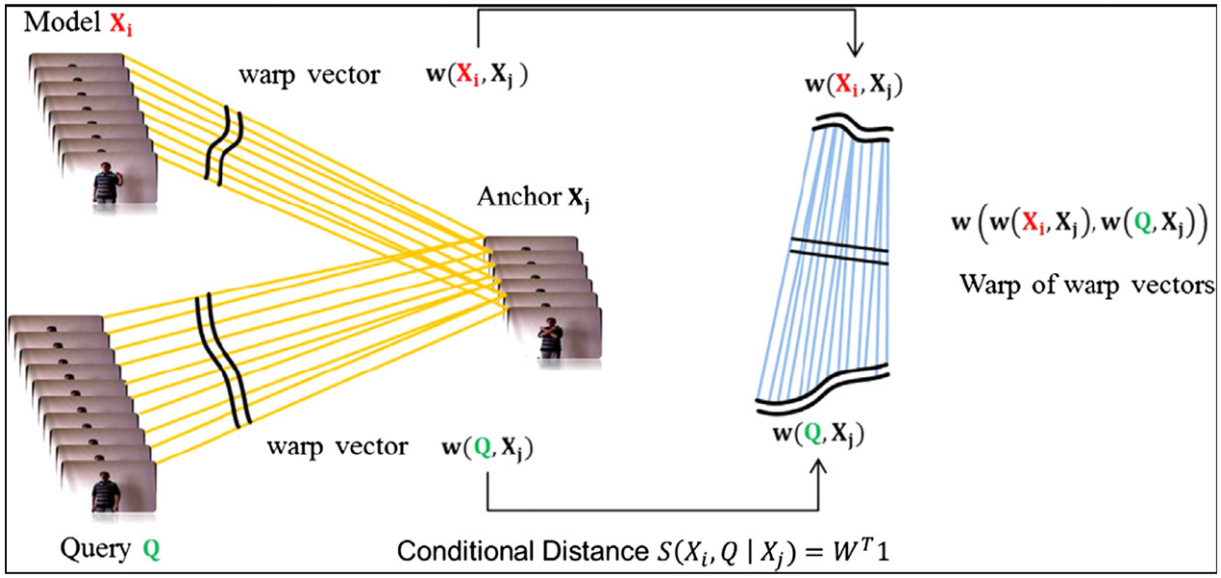

Fig. 8. Conceptual illustration of conditional distance between three sequences: model sequence X_i, anchor sequence X_j and a query sequence Q in [108].

generated by animating the hand model, which is used to perform per-pixel classification and to assign each pixel to a hand part. The classification results are fed into a local mode finding algorithm to estimate the joint locations for the hand skeleton.

In [112] 2.5D graphs are used to recognize static action images. The graph consists of a set of nodes that are key-points for the human body represented by view-independent 3D positions and rich 2D appearance features. The edges are relative distances between

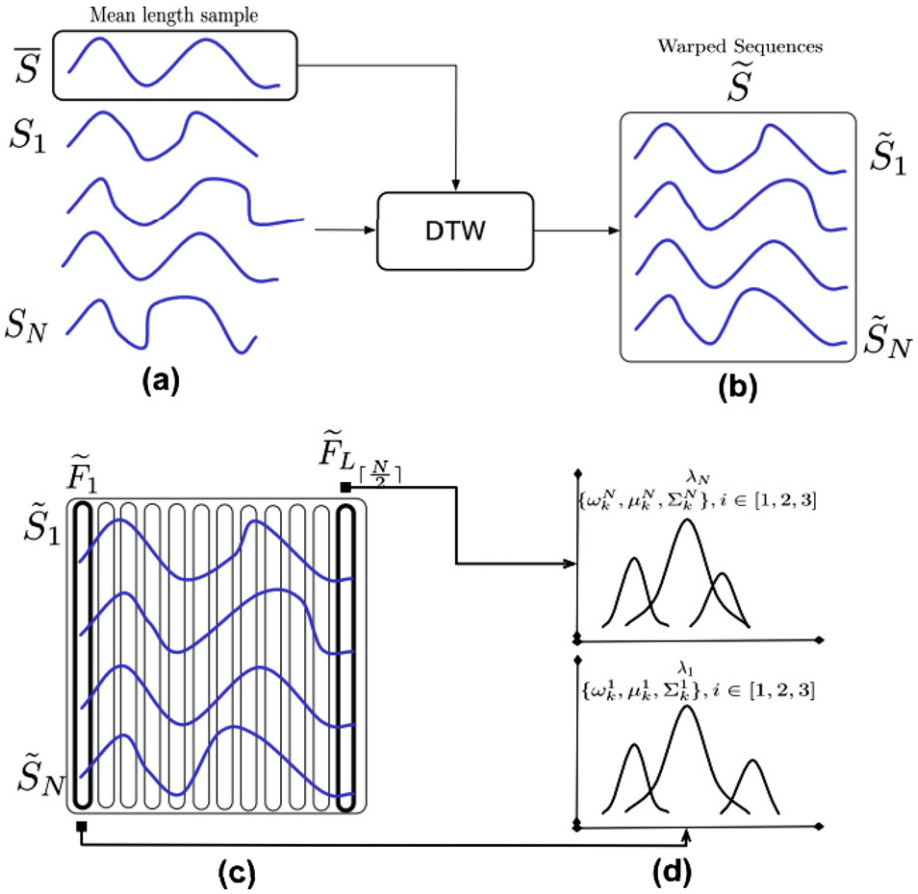

Fig. 9. The probability based DTW in [41] to handle intra-class variance: (a) different sequences of a certain gesture category and the median length sequence. (b) Alignment of all sequences with the mean length sequence by means of Euclidean DTW. (c) Warped sequences set $\tilde{S}$ from which each set of *rth* elements among all sequences are modelled. (d) Gaussian Mixture Model with 3 basis components.

the key-points. Then, estimating the similarity between two action images results in a matching among their corresponding graphs. For each action class, a minimum set of dominating images, able to cover the intra-class pose variations and capture all inter-class distinctions, is selected.

5. Gesture temporal variation and segmentation

The last but not less important point that we consider in this paper concerns the gesture segmentation task. When a gesture is performed in a continuous way it is necessary to identify its starting and ending frame. In addition, during the transition phase between consecutive gestures, intermediate movements could occur as well. These transition motions need to be eliminated in order to avoid false matching with reference patterns. Moreover, gestures of the same type may dynamically vary in shape and duration even if they are executed by the same gesturer. It is worth noting that many of the papers cited in this review do not consider at all either the segmentation or the length variation problems. Usually, tests are performed on public datasets where the instances of different gestures are well separated and directly available for experiments. So different feature-classifiers can be easily compared with other solutions proposed in literature to assess the best performances. However, the problems of segmentation and length variation must be considered if real time and online gesture recognition systems have to be developed.

Generally speaking, classification methods such as those based on NNs, SVMs, HMMs, suffer from these problems and have to solve both the segmentation and the spatial/temporal normalization of gesture sequences. In many cases, during the training phase for the gesture model generation the data steams are marked manually by human observers and some constraints on the gesture lengths are imposed. In [61] and [79] the authors impose that a gesture must be recorded/observed in a fixed time interval (few seconds), so that the incoming video sequences are quickly exploitable because every gesture has the same temporal length. DTW algorithms [59], [110],[41] manage different gesture lengths, but they require the knowledge of the starting/ending frames to align the sequences.

A common way to approach the temporal structure of gestures and manage the length variations is to group features in similar configurations [73], i.e. states, in order to define and to learn temporal transition functions between states. Approaches which use body poses as states, require a preprocessing phase to identify the key poses, but avoid the temporal representation problem as the same gestures performed at different velocities are naturally represented by the same sequence of body postures. By using the same philosophy, in [105] the classification of the frame-wise features and the posterior modelling of the classifier outputs on the sequence level allow to deal with different sequences lengths, and to classify motion sequences with arbitrary speeds and lengths.

Some ad hoc solutions have been experienced for boundary detection such as to identify the transition phases or the starting point of a gesture. For example, in [88], the start/end point of a hand gesture trajectory is identified measuring the hand centroid point velocity, and setting an initial rectangular region of the image as reference for the system in order to start capturing the hand point coordinates [29]. The use of "rest" states, as local minima of 3D motion of the limbs, has been considered as natural transition between primitive gestures. In a similar way, idle gestures can be introduced as transition phase between consecutive executions of the gestures of the vocabulary. In [30], a SVM classifier is trained to recognize neutral postures against non neutral postures. Then, a state machine is used for online experiment to maintain the system in a idle state, when a neutral posture is observed, and in an action state, when a pose becomes non-neutral (see Fig. 10).

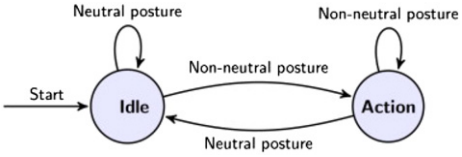

Fig. 10. The on-line action recognition system in [30]. Idle and action states are used to maintain the system in idle while seeing a neutral pose.

Another strategy for recognizing gestures in continuous sequences is by using a sliding window approach. In this case, the video sequence is divided in multiple overlapping segments where classification can be performed. Peaks in the resulting scores are interpreted as the gesture locations [113]. In [57], a set of rules to improve the automatic recognition of gestures in a data stream is added to the sliding window approach. For example, if the liftoff gesture (recognized in the stream of data) is followed by any other gesture with comparably high score value, then only the gesture that followed the liftoff would be returned as intended.

The difference between boundary detection approaches and sliding window approaches stays essentially in the possibility of managing the temporal representation of gestures. In the first case the identification of the start/ending frames allows to handle the gesture length variation in a successive processing phase, such as DTW or length sampling/subsampling steps. Sliding window approaches, instead, require multiple window sizes to achieve robustness against different durations of gesture execution. In this case, sliding window approach can be computationally intensive. In addition, sliding window approaches produce good accuracy when the sequence of gestures are performed in continuous with minimum accidental movements between consecutive gestures [57]. When the sequence is noisy since many unintentional movements are performed, the accuracy decreases as the association with gestures is always done also for meaningless frames.

6. Comprehensive discussion

In order to give a comprehensive overview of all the main problems and the most common solutions which have been proposed in literature, in Table 2 we have reported the main points of the papers that have been discussed and cited in this review. For each work, the type of gesture, i.e. static or dynamic, concerning the hands, arms, or full body is reported. The selected features and the corresponding methodologies used for building the gesture models are also explained and discussed. Then, the use of private or public datasets, is also indicated to highlight the repeatability of experiments. Finally, the last two columns of the tables highlight if the papers address both the problems of segmentation of starting/ending frames, and temporal variation of the gesture. The label "Y" means that these problems have been considered and in some way handled by the proposed methodology, while the words "no-gesture", "idle", etc. indicate the specific solution applied in the referred work. The year of publication is also reported showing that the recent availability of RGB-D sensors gave a great impulse to the research on gesture recognition.

6.1. Analysis of input data

The first point that comes out from this comprehensive analysis is about the use of depth information to solve different problems. First of all, people and body parts can be easily segmented in many scenarios in which cluttered background, shadows, light reflections produce false moving objects in RGB-based approaches. In the context of hand gestures previous studies on skin detection reveal many

Table 2
An overview of the papers discussed in this review.

Year	Reference	Type of gesture	Features	Methodology	Dataset	Start/end seg.	Temporal Variation (Temp. Var.)
2013	[68]	Full body	Joint angles	Inverse kinematics	Private	–	–
2013	[99]	Full body	Motion and shape features	MultiClass SVM	MSR-Action3D, MSR-DailyActivity3D	–	–
2014	[30]	Full body	Spatio/temporal occ. pat.	Action graph	MSR-Action3D	–	–
2012	[40]	Both hands	Salient points on nails	Levenberg–Marquard algorithm	Private	–	–
2012	[57]	Both arms	Coordinates	SVM	Private	Sliding window	–
2011	[98]	Both arms	Depth image sub.	Multiclass SVM	Private	–	–
2013	[59]	Right hand	Coordinates	Distance with learned trajectories with DTW	Private	Sliding window	Y
2012	[93]	Right hand	Coordinates	K-NN with DTW, HMM	Private	–	–
2013	[64]	Both arms	Coordinates	Weighted DTW	Private	–	–
2014	[110]	Full body	Coordinates	Bag of key poses with DTW	MSR-Action3D	–	Y
2012	[112]	Full body	Coordinates	2.5 graph similarity	Pascal2011 Action	–	–
2015	[105]	Full body	Joint coordinates, temporal distances	ELM	MSRC-12, MHAD	–	Y
2015	[113]	Right arm	Joint quaternions	ANN	Private	Sliding window	Y
2015	[106]	Full body	Joint coordinates	PCA, distance metric	Private	–	Y
2013	[11]	Hand (static)	Distance and curvature Features	SVM	Private	–	–
2014	[100]	Hand (static)	Distance, elevation, curvature, and palm area features	SVM	Private	–	–
2014	[79]	Right arm	Joint angles, quaternion	ANN	Private	–	–
2012	[102]	Hand (static)	Binary images	ANN	Private	–	–
2014	[65]	Hand	Quaternion, velocity	Inverse kinematics	Private	–	–
2014	[94]	Right arm	Joint angles	HMM	Private	–	–
2012	[75]	Right arm	Joint angles	Fusion of HMM, DTW	Private	–	–
2012	[76]	Left arm	Joint angles	HMM	Private	–	–
2014	[56]	Both arms	Joint coordinates and angles	Rule based	Private	–	–
2013	[37]	Hand (static)	SIFT features	K-means clustering	Private	–	–
2015	[86]	Hand sign, arms	Color, depth, body joint positions	3D CNN, GMM-HMM	Private	–	–
2013	[41]	Both arms	STIP features	Probability-based DTW	ChaLearn	Idle gesture	Y
2008	[27]	Both arms	3D point cloud, spherical histogram	Correlation probability distance	Private	Y	–
2012	[70]	Both arms	Joint angles	Linear regression Analysis	Private	–	–
2014	[90]	Both arms	Hand coordinate velocity	HMM	Private	Y	–
2011	[38]	Hand (static)	Hand skeleton joints	ANN, SVM	ASL images (private)	–	–
2013	[111]	Hand (static)	Depth	RDF	ASL images	–	–
2015	[108]	Both arms	HOG	Conditional distance	ChaLearn	–	Y
2012	[36]	Hand	Cell occupancy silhouette	Action graph	ASL images (private)	–	Y
2012	[61]	Both arms	Coordinates	Nearest neighbour	Private	Y	–
2006	[50]	Full body	Joint angles	GMM + HMM	KU gesture database and private	Garbage gesture	–
2010	[15]	Full body	Bags of 3D points	Action graph	Private	–	–
2012	[35]	Hand (static)	Contour and finger positions	Distance measure	Private	–	–
2015	[42]	Full body	STIP features on RGB and depth	SVM	RGBD-HuDaAct	–	–
2013	[20]	Full body	Spatio-temporal RGB and depth features	GP for feature selection, SVM	MSRDailyActivity3D	–	–
2014	[73]	Full body	Joint angles	SVM for key pose detection, decision forest	Private	–	Y
2013	[69]	Full body (static)	Joint positions, distances and angles	K-means	Private	–	–
2014	[103]	Upper body	Joint positions and velocity	ANN	VisApp2013 dataset	–	–
2008	[26]	Both arms	Depth silhouette	HMM, SVM	Private	–	–
2013	[104]	Full body	Depth data, skeleton joints	RNN	Private	–	–
2014	[87]	Upper body	Intensity, depth, articulated pose	Convolutional NN	ChaLearn	–	Y
2012	[97]	Both arms	Joint coordinates	HMM	Private	–	–
2011	[46]	Full body	3D-MHI, (DLMC-STIP	SVM	RGBD-HuDaAc	–	–
2013	[62]	Arm	Coordinates	Decision tree, SVM	Private	Y	–

Table 2 (*continued*)

Year	Reference	Type of gesture	Features	Methodology	Dataset	Start/end seg.	Temporal Variation (Temp. Var.)
2013	[74]	Both arm	Joint angles, HOG on depth image	SVM	MSR-Action3D, MSR-HandGesture, UCF-Kinect	–	–
2013	[89]	Both arm	Joint coordinates, velocities	Cascade HMM	Private	No-gesture	–
2012	[58]	Full body	Coordinates	Optimal classifier among NN, SVM, decision tree	Private	–	–
2013	[81]	Arm	Quaternions	DTW, multi-class probability estimates	Private	–	Y
2014	[83]	Both arms	Color depth skeleton	SVM	ChaLearn	Sliding window	Y
2013	[72]	Full body	Joint angles	Distance metric	Private	–	Y
2011	[34]	Hand (static)	Finger detection	Finger earth mover's distance (F-EMD), template matching	Private	–	–
2011	[63]	Full body	Joint coordinates	Weighted DTW	Private	Y	Y
2013	[60]	Left arm	Joint coordinates	GMM and HMM	Private	No-gesture	Y
2011	[77]	Both arms	Quaternions	DTW	Private	–	–
2015	[66]	Full body	Joint coordinates	Grassmann manifold, SVM	MSR-action3D, UT-kinect and UCF-kinect	–	Y
2013	[92]	Full body	Joint coordinates	GMM-HMM	Private	–	–
2014	[39]	Full body	Point cloud, body surface context	kNN, SVM	MSR Action3D, MSRDailyActivity3D	–	–
2013	[80]	Full body	Quaternions	ANN	3DLife grand challenge	–	–
2010	[28]	Hand	2D, 3D shape and volumetric descriptor	SVM	Private	–	–
2013	[78]	Upper body	Quaternions	SVM	Private	–	–
2013	[109]	Upper body	3D EMoSIFT	BoF based model	ChaLearn	Y	Y
2015	[48]	Hand (static)	Depth-color superpixel	SP-EMD	Private	–	–
2013	[29]	Left arm	Motion history images	SVM	Private	Interest region	Y
2012	[31]	Full body, hand	Random occupancy Pattern features	SVM	MSR-Action3D, Gesture3D	–	–
2014	[18]	Full body	Multiview spatio temporal graph	Dynamic programming	Northwestern-UCLA Multiview 3D, Action3D	–	–
2012	[91] [33]	Hand	Palm coordinates	HMM	Private	No-gesture	–
2014	[84]	Upper body	Audio, skeleton joint coordinates	DBN, HMM	ChaLearn	–	–
2016	[85]	Upper body	Skeleton joints, depth and RGB images	Deep dynamic NN, HMM	ChaLearn	Y	Y
2013	[45]	Full body	Depth STIP	SVM	MSRDailyActivity3D	–	–
2012	[88]	Both hands	Hand centroid coordinates	HMM	Private	Initial region	Y
2012	[96]	Both hands	Hand orientation	HMM	Private	–	–
2011	[82]	Full body	Color, optical flow spatio-temporal grad. pose features	Decision forest	TUM kitchen dataset	–	–
2012	[107]	Hand	Contour descriptor	Contour matching	Private	–	–
2011	[44]	Full body	4D spatio-temporal cuboid	LDA	Private	–	–
2012	[43]	Full body	STIP on RGB and depth map features	SVM	Private	–	–

In the first column the *Year* of publication, then the *Reference*, the *Type of gesture*, the *Features*, and the *Methodology* are summarized. The column *Dataset* indicates if private or public datasets are considered in the experiments. The last two columns, *Start/end segmentation* and *Temp. var.*, report if the papers have considered the problem of detecting the initial frame of the gesture and manage possible temporal variations.

challenges, such as low accuracy of predicting the body movements of users and the weakness against cluttered backgrounds or various lighting conditions [114]. The depth information and skeleton tracking provided by recent depth sensors, address these problems. Also the human body silhouettes are detected as connected regions well separated from the background. In this way, silhouettes can be used to extract curvature features, or depth features including volumetric descriptors, 3D SIFT, STIP, and bags of 3D points [15,37,41,42]. In addition, the availability of skeletons of the body silhouettes together with all the related information such as joint coordinates and angles, provides quickly significant features which characterize the motion. Many works consider a selection or combination of these features to build gesture models and perform recognition. Actually the type of gesture affects the feature selection, as more complex the gestures in terms of trajectories more complex features are necessary. Table 2 highlights that most of the papers considers only one kind of feature such as geometric, depth or intensity based feature. The fusion of different sensor modalities such as depth and color to generate more discriminant features has been explored only in a few papers [43,48,104]. The reason may be that although the color texture and depth map are captured simultaneously, they are not registered accurately. Therefore calibration procedures must be applied before the joint use of color and depth information.

6.2. Considerations on methodologies

Another important point concerns the methodologies used to generate gesture models. Supervised classifiers, such as SVM, HMM,

or ANN as well as distance based approaches have been largely used. There are no strict rules or solid indications on the most appropriate solutions. The analysis of the literature reveals that the choice depends on the complexity of features and the separability of gestures. Comparisons among different classifiers have been done in different situations with different sets of gestures and features, then they lead to conflicting conclusions on the most performant approaches [38,58,93]. But even if the conclusions in a fixed scenario were considered as valid, they would be limited to that set of experiments: different people in different session would provide different results.

Machine learning approaches (including SVM, HMM, and ANN) require huge amounts of training data, but at the same time they are more robust to manage intra-class variability and to generalize when new gesture instances occur. Distance based approaches may require a small number of training examples and find large applications when there are few gestures which are well separated in time.

Among the supervised classifiers, ANN and SVM require more strict constraints on the variability of the gesture length. As they are trained with fixed length sequences they are prone to error when test sequences differ in length from the training ones. HMM are more suitable to handle the gesture time invariance as they can deal with sequential data.

The last consideration concerns the possibility of using multiple classifiers to improves performances of each individual approach. If the input data were noiseless, all the methodologies could have perfect generalization performances. Actually, such generalization is quite impossible as classifiers work on data produced by different people in different sessions. At the best, different classifiers produce good results most of the time, and due to their different nature, they make errors on different instances. For this reason, according to the context and by measuring the classifiers diversity, a strategic combination of classifiers can reduce the total errors, improving the performance of a single classifier. Some recent attempts have been done in [75], where DTW and HMM sharing the same features and fusion rules are designed to make a global decision.

6.3. Typology of experiments

Many datasets of gesture/action sequences of RGB-D images provide a common benchmark that allows comparisons among different features and classifiers in order to assess the best combination feature/methodology. These datasets consist of sequences already segmented and contains one instance of each gesture. For this reason, all the works involving experiments on these sequences do not consider at all the problem of detecting the starting/ending frame of the gesture. Only in [41,50], the recognition of an idle or garbage gesture between consecutive instances is used to provide the temporal segmentation. Starting from this consideration, it is evident that performing gesture recognition for online applications remains a challenging task. Experiments on private datasets, containing sequences of consecutive executions of different gestures, are important in order to manage the problems of segmentation and temporal variations of gestures. As reported in Table 2, few works discuss these problems, and few of them propose a solution. Some suggest sliding window approaches [57] [59], some propose the introduction of a non-gesture [33,60,89,91], while in [109] the definition of an initial starting region to separate preparatory movements from actual gesture movements is suggested. By the analysis of experiments, another consideration is evident: there is a common lack of clear references to computational times or explicative notes about delays between gesture executions and system decisions. These comments could provide a great insight for the application of the proposed approaches to real human machine interfaces.

Another important problem related to gesture recognition is the invariance to the orientation of the gesturer with respect to the camera. Some papers introduce different kinds of normalization in order to be independent of the distances involved. But regarding the orientation, just a few papers discuss the invariance to small rotations of the subjects performing gestures with respect to the depth sensors. This problem has not been discussed in this review due to the lack of literature works that investigate it. However this lack of works is due to the intrinsic limitation of the depth sensors, which provide all the information when the subject is facing the camera. In addition, the software for skeleton extraction is unstable when the subject is not completely visible, so even in presence of small rotations the arms and the body are seen as a whole (occlusion problem) so the skeleton detection has a low reliability. One possible solution is the use of a multiple camera setup as proposed in [115], where a Kinect camera is used together with a monocular camera in a stereo camera setup to cope with occlusion problems. As the Kinect was originally built for a gaming console, it was assumed that the user is always facing the camera, so using Kinect alone does not suffice to handle occlusions. Another fundamental limitation of low cost depth cameras, such as the Kinect, is that they are highly affected by lighting conditions preventing their use in outdoor contexts under natural light.

6.4. Future research

Future directions of researches must focus on the multi modal fusion of data in order to overtake the intrinsic instability of each modality and to increase performances in on-line applications. The concurrent use of features extracted from RGB, depth and skeleton data can greatly make current methodologies more robust in challenging contexts. By one hand depth data can be used to segment the scene overcoming the limitation of classical RGB-based approaches, by the other hand the color information can give further insight on patterns which univocally characterize the gestures. The results can be more stable also when the structured light used to generate depth fails or the algorithms for skeleton extractions give unreliable postures.

In the same way, the use of ensemble of classifiers is certainly a strategy which should be considered. If individual classifiers make errors on different instances, then a strategic combination of these classifiers can reduce the total error. A measure of diversity among classifiers allows to establish if decision boundaries are adequately different and if the fusion can provide a more accurate classification. A special attention should be put to developing deep learning algorithms which learn invariant and discriminative hierarchical representations and would be helpful when the input from one or more modalities is missing or noisy or to cope with gestures performed at different speeds.

The last but not less important point is the necessity of developing methods independent of the sensor technology. Most of the recent works use Microsoft Kinect sensors together with the related software platforms for feature extraction. As a consequence, the methods applied to extract the features inherit the same limitations of the sensor. Thanks to technological developments and future studies, range sensors will greatly improve in both resolution and robustness. This will hopefully allow for the generalization of both Kinect-like software frameworks and already developed gesture recognition methodologies in order to obtain more robust results.

7. Conclusions

In the last years gesture recognition approaches have found applications in many fields especially with the large spreading of low cost RGB-D sensors. In this paper we have analyzed the recent literature to understand the applicability of gesture recognition approaches in real contexts. The analysis have revealed that several factors greatly

affect their application: 1) the capability of managing continuous acquisitions; 2) the invariance to different executions of gestures in terms of both amplitude and velocity; 3) the robustness of methodologies to recognize gestures performed by people different from those used to build the gesture models; 4) the invariance to the distance and orientation between sensors and gesturers. For these reasons, as soon as new sensors will be available on the market, public datasets involving real conditions will be necessary to provide benchmark for challenging comparisons of methodologies. Moreover, the availability of open source code implementing different methodologies and the sharing of the information about the relative parameters will also allow the reproduction of published results on different scenarios easing comparisons and further improvements.

References

[1] G.R.S. Murthy, R.S. Jadon, A review of vision based hand gesture recognition, Int. J. Inf. Technol. Knowl. Manag. 2 (2) (2009) 405–419.
[2] S. Mitra, T. Acharya, Gesture recognition: a survey, IEEE Trans. Syst. Man Cybern. Part C Appl. Rev. 37 (3) (2007) 311–324.
[3] D. Weinland, R. Ronfard, E. Boyer, A survey of vision-based methods for action representation, segmentation and recognition, Comput. Vis. Image Underst. 115 (2011) 224–241.
[4] R. Poppe, A survey on vision-based human action recognition, Image Vis. Comput. 28 (2010) 976–990.
[5] M.H. Hassan, P.K. Mishra, Hand gesture modeling and recognition using geometric features: a review, Can. J. Image Process. Comput. Vis. 3 (1) (2012) 12–26.
[6] J. Han, L. Shao, D. Xu, J. Shotton, Enhanced computer vision with Microsoft Kinect sensor: a review, IEEE Trans. Cybern. 43 (43) (2013) 1318–1334.
[7] J.K. Aggarwal, L. Xia, Human activity recognition from 3D data: a review, Pattern Recogn. Lett. 48 (2014) 70–80.
[8] L. Chen, H. Wei, J. Ferryman, A survey of human motion analysis using depth imagery, Pattern Recogn. Lett. 34 (2013) 1995–2006.
[9] M. Ye, Q. Zhang, L. Wang, J. Zhu, R. Yang, J. Gall, A survey on human motion analysis from depth data, Time-of-Flight and Depth Imaging. Sensors, Algorithms, and Applications, 2013, 149–187.
[10] J.L.a. Viola, 3D Gestural Interaction: The State of the Field, ISRN Artificial Intelligence, 2013.
[11] F. Dominio, M. Donadeo, G. Marin, P. Zanuttigh, G.M. Cortelazzo, Hand gesture recognition with depth data, Proceedings of the 4th ACM/IEEE International Workshop on Analysis and Retrieval of Tracked Events and Motion in Imagery Stream, 2013. pp. 9–16.
[12] B.W. Hwang, S. Kim, S.W. Lee, A full body gesture database for automatic gesture recognition, 7th International Conference on Automatic Face and Gesture Recognition, (FGR), 2006. pp. 243–248.
[13] S. Fothergill, H. Mentis, P. Kohli, S. Nowozin, Instructing people for training gestural interactive systems, Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 2012. pp. 1737–1746.
[14] MSR, Action Recognition Datasets and codes, http://research.microsoft.com/en-us/um/people/zliu/actionr ecorsrc/.
[15] W. Li, Z. Zhang, Z. Liu, Action recognition based on a bag of 3D points, IEEE Computer Society Conference on Computer Vision and Pattern Recognition — Workshops, 2010. pp. 9–14.
[16] ChaLearn Gesture Dataset (CGD2011), ChaLearn California, 2011.
[17] S. Escalera, J. Gonzlez, X. Baró, M. Reyes, O. Lopes, I. Guyon, V. Athistos, H.J. Escalante, Multi-modal Gesture Recognition Challenge 2013: Dataset and Results, ICMI. 2013.
[18] J. Wang, X. Nie, Y. Xia, Y. Wu, S.C. Zhu, Cross-view action modeling, learning and recognition, 2014. pp. 2649–2656.
[19] Z. Ren, J. Meng, J. Yuan, Z. Zhang, Robust hand gesture recognition with Kinect sensor, Proc. of ACM Intl. Conf. on Multimedia (ACM MM 11), Scottsdale, Arizona, USA, 2011.
[20] L. Liu, L. Shao, Learning discriminative representations from RGB-D video data, Proc. International Joint Conference on Artificial Intelligence (IJCAI), Beijing, China, 2013.
[21] M.W. Kadous, Australia Sign Language Signs, http://kdd.ics.uci.edu/databases/auslan/auslan.data.html.
[22] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, R. Bajcsy, Berkeley MHAD: a comprehensive multimodal human action database, IEEE Workshop on Applications of Computer Vision (WACV), 2013. pp. 53–60.
[23] ChaLearn, Multi-modal Gesture Recognition Challenge 2013, http://gesture.chalearn.org/ .
[24] M. Ye, Q. Zhang, L. Wang, J. Zhu, R. Yang, J. Gall, A survey on human motion analysis from depth data, time-of-flight and depth imaging. Sensors, Algorithms, and Applications, 2013, 149–187.
[25] C. Castiello, T. D'Orazio, A.M. Fanelli, P. Spagnolo, M.A. Torsello, A model free approach for posture classificatin, IEEE Conf. on Advances Video and Signal Based Surveillance, AVSS 2005.
[26] R. Munoz-Salinas, R. Medina-Carnicer, F.J. Madrid-Cuevas, A. Carmona-Poyato, Depth silhouettes for gesture recognition, Pattern Recogn. Lett. 29 (3) (2008) 319–329. ISSN 0167-8655.
[27] M.B. Holte, T.B. Moeslund, P. Fihl, Fusion of range and intensity information for view invariant gesture recognition, Computer Vision and Pattern Recognition Workshops, 2008. CVPRW '08. Conference on IEEE Computer Society, 2008. pp. 1–7. ISSN 2160-7508.
[28] P. Suryanarayan, A. Subramanian, D. Mandalapu, Dynamic hand pose recognition using depth data, Pattern Recognition (ICPR), 2010 20th International Conference on, 2010. pp. 3105–3108. ISSN 1051-4651.
[29] H. Wang, J. Fu, Y. Lu, S. Li, Depth sensor assisted real-time gesture recognition for interactive presentation, J. Vis. Commun. Image R. 24 (2013) 1458–1468.
[30] A.W. Vieira, E.R. Nascimento, G.L. Oliveira, Z. Liu, M.F.M. Campos, On the improvement of human action recognition from depth map sequences using spacetime occupancy patterns, Pattern Recogn. Lett. 36 (2014) 221–227.
[31] J. Wang, Z. Liu, J. Chorowski, Z. Chen, Y. Wu, Robust 3D action recognition with random occupancy patterns, Proceedings of the 12th European Conference on Computer Vision — Volume Part II, ECCV'12, 2012. pp. 872–885.
[32] T. Hongyong, Y. Youling, Finger tracking and gesture interaction with Kinect, IEEE 12th International Conference on Computer and Information Technology, 2012. pp. 214–218.
[33] X. Wu, C. Yang, Y. Wang, H. Li, S. Xu, An intelligent interactive system based on hand gesture recognition algorithm and Kinect, IEEE Fifth International Symposium on Computational Intelligence and Design, 2012. pp. 294–298.
[34] Z. Ren, J. Yuan, J. Meng, Z. Zhang, Robust part-based hand gesture recognition using Kinect sensor, IEEE Trans. Multimedia 15 (5) (2013) 1110–1120.
[35] Y. Li, Multi-scenario gesture recognition using Kinect, 17th International Conference on Computer Games (CGAMES), 2012. pp. 126–130.
[36] A. Kurakin, Z. Zhang, Z. Liu, A real time system for dynamic hand gesture recognition with a depth sensor, Proceedings of the 20th European Signal Processing Conference (EUSIPCO), 2012. pp. 1975–1979.
[37] M. Hamissi, K. Faez, RealTime hand gesture recognition based on the depth map for human robot interaction, Int. J. Electr. Comput. Eng. (IJECE) 3 (6) (2013) 770–778.
[38] C. Keskin, F. Kirac, Y.E. Kara, L. Akarun, Real time hand pose estimation using depth sensors, IEEE International Conference on Computer Vision Workshops (ICCV Workshops), 2011. pp. 1228–1234.
[39] Y. Song, J. Tang, F. Liu, Shuicheng. Yan, Body surface context: a new robust feature for action recognition from depth videos, IEEE Trans. Circuits Syst. Video Technol. 24 (6) (2014) 952–964.
[40] L. Ballan, A. Taneja, J. Gall, L. Gool, M. Pollefeys, Motion capture of hands in action using discriminative salient points, European Conference on Computer Vision (ECCV), Lecture Notes in Computer Science 7577, 2012. pp. 640–653.
[41] A. Hernandez-Vela, M. ngel Bautista, X. Perez-Sala, V. Ponce-Lopez, S. Escalera, X. Bara, O. Pujol, C. Angulo, Probability-based dynamic time warping and bag-of-visual-and-depth-words for human gesture recognition in RGB-D, Pattern Recogn. Lett. 50 (2013) 112–121.
[42] M-Yuan, H. Liu, F. Sun, RGB-D action recognition using linear coding, Neurocomputing (2015) 79–85.
[43] Y. Zhao, Z.C. Liu, L. Yang, H. Cheng, Combing RGB and depth map features for human activity recognition, Asia-Pacific Signal Information Processing Association Annual Summit and Conf., 2012. pp. 1–4.
[44] H. Zhang, L.E. Parker, 4-dimensional local spatio-temporal features for human activity recognition, IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems, 2011. pp. 2044–2049.
[45] L. Xia, J. Aggarwal, Spatio-temporal depth cuboid similarity feature for activity recognition using depth camera, IEEE Conference on Computer Vision and Pattern Recognition, 2013. pp. 2834–2841.
[46] B. Ni, G. Wang, P. Moulin, RGBD-HuDaAct: a color-depth video database For human daily activity recognition, IEEE International Conference on Computer Vision Workshops, 2011. pp. 1147–1153.
[47] Y. Zhu, W. Chen, G. Guo, Evaluating spatiotemporal interest point features for depth-based action recognition, Image Vis. Comput. (2014) 453–464.
[48] C. Wang, Z. Liu, S. Chan, Superpixel-based hand gesture recognition With Kinect depth camera, IEEE Transaction on Multimedia (1) (2015) 29–39.
[49] G. Johansson, Visual motion perception., Sci. Am. (1975)
[50] S.-W. Lee, Automatic gesture recognition for intelligent human-robot interaction, 7th International Conference on Automatic Face and Gesture Recognition, (FGR), IEEE. 2006, pp. 645–650.
[51] Z. Zhang, Microsoft Kinect Sensor and Its Effect, IEEE Multimedia (2) (2012) 4–12.
[52] J. Webb, J. Ashely, Beginning Kinect Programming with the Microsoft Kinect SDK, 1st ed., Apress, Berkely, CA, USA .
[53] OpenNI organization, OpenNI User Guide, 2010, URL http://www.openni.org/documentation.
[54] OpenNI organization, OpenKinect Project, 2012, URL http://openkinect.org.
[55] Computer Vision Interaction, ViiM SDK, http://www.covii.pt/viim/.
[56] T. Hachaj, M.R. Ogiela, Rule-based approach to recognizing human body poses and gestures in real time, Multimedia Systems 20 (1) (2014) 81–99.
[57] S. Bhattacharya, B. Czejdo, N. Perez, Gesture classification with machine learning using Kinect sensor data, IEEE Third International Conference on Emerging Applications of Information Technology (EAIT), 2012. pp. 348–351.
[58] O. Patsadu, C. Nukoolkit, B. Watanapa, Human gesture recognition using Kinect camera, Computer Science and Software Engineering (JCSSE), 2012 Conference on International Joint, IEEE. 2012, pp. 28–32.
[59] S. Bodiroža, G. Doisy, V.V. Hafner, Position-invariant, real-time gesture recognition based on Dynamic Time Warping, Proceedings of the 8th ACM/IEEE

International Conference on Human-robot Interaction, IEEE Press. 2013, pp. 87–88.
[60] G. Saponaro, G. Salvi, A. Bernardino, Robot anticipation of human intentions through continuous gesture recognition, IEE International Conference on Collaboration Technologies and Systems (CTS), 2013. pp. 218–225.
[61] K. Lai, J. Konrad, P. Ishwar, A gesture-driven computer interface using Kinect, IEEE Southwest Symposium on Image Analysis and Interpretation (SSIAI), IEEE. 2012, pp. 185–188.
[62] J. Oh, T. Kim, H. Hong, Using binary decision tree and multiclass SVM for human gesture recognition, IEEE International Conference on Information Science and Applications (ICISA), 2013. pp. 1–4.
[63] M. Reyes, G. Dominguez, S. Escalera, Feature weighting in dynamic time-warping for gesture recognition in depth data, Computer Vision Workshops (ICCV Workshops), 2011 Conference on IEEE International, IEEE. 2011, pp. 1182–1188.
[64] S. Celebi, A.S. Aydin, T.T. Temiz, T. Arici, Gesture recognition using skeleton data with weighted dynamic time warping, Computer Vision Theory and Applications. VISAPP, 2013.
[65] G. Du, P. Zhang, Di. Li, Human-manipulator interface based on multisensory process via Kalman filters, IEEE Trans. Ind. Electron. 61 (10) (2014) 5411–5418.
[66] R. Slama, H. Wannous, M. Daoudi, A. Srivastava, Accurate 3D action recognition using learning on the Grassmann manifold, Pattern Recogn. (2015) 556–567.
[67] D. Wu, L. Shao, Leveraging hierarchical parametric networks for skeletal joints based action segmentation and recognition, IEEE Conference on Computer Vision and Pattern Recognition, 2014. pp. 724–731.
[68] I. Almetwally, M. Mallem, Real-time tele-operation and tele-walking of humanoid robot Nao using Kinect depth camera, 10th IEEE International Conference on Networking, Sensing and Control (ICNSC), 2013. pp. 463–466.
[69] R. Mangera, Static gesture recognition using features extracted from skeletal data, Proceedings of the Twenty-fourth Annual Symposium of the Pattern Recognition Association of South Africa, 2013.
[70] I.I. Itauma, H. Kivrak, H. Kose, Gesture imitation using machine learning techniques, 20th Signal Processing and Communications Applications Conference (SIU), 2012.
[71] I.I. Itauma, H. Kivrak, H. Kose, Road traffic control gesture recognition using depth images, IEEE Trans. Smart Process. Comput. 1 (1). (2012)
[72] M. Raptis, D. Kirovski, H. Hoppe, Real-time classification of dance gestures from skeleton animation, Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA '11, 2011. pp. 147–156.
[73] L.Miranda, T. Vieira, D. Martinez, T. Lewiner, A. Vieira, M. Campos, Online gesture recognition from pose kernel learning and decision forests, Pattern Recognition Letters 39 (2014) 65–73.
[74] E. Ohn-Bar, M.M. Trivedi, Joint angles similarities and HOG2 for action recognition, IEEE Conference on Computer Vision and Pattern Recognition Workshops: Human Activity Understanding from 3D Data (2013) 465–470.
[75] Y. Gu, Q. Cheng, W. Sheng, Classifier fusion for gesture recognition using a Kinect sensor, International Conference on Computer Applications in Industry and Engineerding, 2012. pp. 187–192.
[76] Y. Gu, H. Do, Y. Ou, W. Sheng, Human gesture recognition through a Kinect sensor, IEEE International Conference on Robotics and Biomimetics (ROBIO), 2012. pp. 1379–1384.
[77] S. Sempena, N.U. Maulidevi, P.R. Aryan, Human action recognition using dynamic time warping, IEEE International Conference on Electrical Engineering and Informatics (ICEEI), 2011. pp. 1–5.
[78] H.Y. Ting, K.S. Sim, F.S. Abas, R. Besar, Vision-based human gesture recognition using Kinect sensor, The 8th International Conference on Robotic, Vision, Signal Processing Power Applications,Lecture Notes in Electical Engineering 291 (2013) 239–244.
[79] T. D'Orazio, C. Attolico, G. Cicirielli, C. Guaragnella, A neural network approach for human gesture recognition with a Kinect sensor, International Conference on Pattern Recognition Applications and Methods (ICPRAM 2014), INSTICC. 2014, pp. 741–746.
[80] L. Sun, K. Aizawa, Action recognition using invariant features under unexampled viewing conditions, Proceedings of the 21st ACM International Conference on Multimedia, ACM. 2013, pp. 389–392.
[81] P.K. Pisharady, M. Saerbeck, Robust gesture detection and recognition using dynamic time warping and multi-class probability estimates, IEEE Symposium on Computational Intelligence for Multimedia, Signal and Vision Processing (CIMSIVP), 2013. pp. 30–36.
[82] A. Yao, J. Gall, G. Fanelli, L.V.a.n. Gool, Does human action recognition benefit from pose estimation? Proceedings of the British Machine Vision Conference, BMVA Press. 2011, pp. 67.1–67.11.
[83] X. Peng, L. Wang, Z. Cai, Y. Qiao, Action and gesture temporal spotting with super vector representation, European Conference on Computer Vision (ECCV) Workshops, in: Lecture Notes in Computer Science, 2014. pp. 518–527.
[84] D. Wu, L. Shao, Multimodal dynamic networks for gesture recognition, Proceedings of the 22nd ACM International Conference on Multimedia, 2014. pp. 945–948.
[85] D. Wu, L. Pigou, P.J. Kindermans, N. Le, L. Shao, J. Dambre, J.M. Obodez, Deep dynamic neural networks for multimodal gesture segmentation and recognition, IEEE Trans. Pattern Anal. Mach. Intell. 2016, http://dx.doi.org/10.1109/TPAMI.2016.2537340.
[86] J. Huang, W. Zhou, H. Li, W. Li, Sign language recognition using 3D convolutional neural networks, IEEE International Conference on Multimedia and Expo (ICME), 2015. pp. 1–6.
[87] N. Neverova, C. Wolf, G.W. Taylor, F. Nebout, Multi-scale deep learning for gesture detection and localization, Computer Vision — ECCV 2014 Workshops, Lecture Notes in Computer Science 8925, 2014. pp. 474–490.
[88] D. Xu, Y.L. Chen, C. Lin, X. Kong, X. Wu, Real-time dynamic gesture recognition system based on depth perception for robot navigation, IEEE International Conference on Robotics and Biomimetics (ROBIO), 2012. pp. 689–694.
[89] C.B. Park, S.W. Lee, Real-time 3D pointing gesture recognition for mobile robots with cascade HMM and particle filter, Image and Vision Computing 29 (1) (2011) 51–63.
[90] M.G. Jacob, J.P. Watchs, Context-based hand gesture recognition for the operating room, Pattern Recogn. Lett. 36 (2014) 196–203.
[91] Y. Wang, C. Yang, X. Wu, S. Xu, H. Li, Kinect based dynamic hand gesture recognition algorithm research, 4th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2012. pp. 274–279.
[92] Y. Song, Y. Gu, P. Wang, Y. Liu, A. Li, A Kinect based gesture recognition algorithm using GMM and HMM, 6th International Conference onBiomedical Engineering and Informatics (BMEI), 2013. pp. 750–754.
[93] J.M. Carmona, J. Climent, A performance evaluation of HMM and DTW for gesture recognition, Lect. Notes Comput. Sci 7441 (2012) 236–243.
[94] T. Fujii, J.H. Lee, S. Okamoto, Gesture recognition system for human-robot interaction and its application to robotic service task, International MultiConference of Engineers and Computer Scientists I, 2014.
[95] R. Plamondon, X. Li, M. Oarizeau, Training hidden Markov models with multiple observations — a combinatorial method, IEEE Trans. Pattern Anal. Mach. Intell. 22 (4) (2000) 371–377.
[96] W. Xu, E.J. Lee, Continuous gesture recognition system using improved HMM algorithm based on 2D and 3D space, Int. J. Multimed. Ubiquit. Eng. 7 (2) (2012) 335–340.
[97] N. Nguyen-Duc-Thanh, S. Lee, D. Kim, Two-stage hidden Markov model in gesture recognition for human robot interaction, Int. J. Adv. Robot. Syst. 9 (2012)
[98] K.K. Biswas, S.K. Basu, Gesture recognition using Microsoft Kinect®, 5th IEEE International Conference on Automation, Robotics and Applications (ICARA), 2011, 2011. pp. 100–103.
[99] S. Althloothi, M.H. Mahoor, X. Zhang, R.M. Voyles, Human activity recognition using multi-features and multiple kernel learning, Pattern Recogn. 47 (2014) 1800–1812.
[100] F. Dominio, M. Donadeo, P. Zanuttigh, Combining multiple depth-based descriptors for hand gesture recognition, Pattern Recogn. Lett. 50 (2014) 101–111.
[101] N.A. Ibraheem, R.Z. Khan, Vision based gesture recognition using neural networks approaches: a review, Int. J. Hum. Comput. Interact. (IJHCI)) 3 (1). (2012).
[102] M. dos Santos Anjo, E.B. Pizzolato, S. Feuerstack, A real-time system to recognize static gestures of Brazilian sign language (Libras) alphabet using Kinect, Proceedings of the 11th Brazilian Symposium on Human Factors in Computing Systems, 2012. pp. 259–268.
[103] R. Mangera, F. Senekal, Fred. Nicolls, Cascading Neural Networks for Upper–body Gesture Recognition, Proceedings of the International Conference on Machine Vision and Machine Learning, 2014. pp. 59:1–59:8.
[104] N. Nerenova, C. Wolf, G. Paci, G. Sommavilla, G. Taylor, F. Nebout, A multi-scale approach to gesture detection and recognition, ICCV Workshop on Understanding Human Activities: Context and Interactions (HACI), 2013, 484–491.
[105] X. Chen, M. Koslela, Skeleton-based action recognition with extreme learning machines, Neurocomputing (2015) 387–396.
[106] I.J. Ding, C.E. Chang, An eigenspace-based method with a user adaptation scheme for human gesture recognition by using Kinect 3D data, Appl. Math. Model. 39 (19) (2015) 5769–5777.
[107] Y. Yao, Y. Fu, Contour model based hand-gesture recognition using Kinect sensor, IEEE Trans. Circuits Syst. Video Technol. 24 (11) (2014) 1935–1944.
[108] R. Krishnan, S. Sarkar, Conditional distance based matching for one-shot gesture recognition, Pattern Recogn. (2015) 1302–1314.
[109] J. Wan, Q. Ruan, W. Li, One-shot learning gesture recognition from RGB-D data using bag of features, J. Mach. Learn. Res. 14 (2013) 2549–2582.
[110] A.A. Chaaraoui, J.R. Padilla-Lopez, P. Climent-Perez, F. Florez-Revuelta, Evolutionary joint selection to improve human action recognition with RGB-D devices, Expert Syst. Appl. 41 (3) (2014) 786–794.
[111] C. Keskin, F. Kirac, Y.E. Kara, L. Akarun, Real time hand pose estimation using depth sensors, Consumer Depth Cameras for Computer Vision, Springer. 2013, 119–137.
[112] X. Chen, M. Koskela, Action recognition with exemplar based 2.5D graph matching, European Conference on Computer Vision (ECCV), Lecture Notes in Computer Science 7575, 2012. pp. 173–186.
[113] G. Cicirelli, C. Attolicoa, C. Guaragnella, T. D'Orazio, A Kinect-based gesture recognition approach for a natural human robot interface, Int. J. Adv. Robot. Syst. 12 (2015)
[114] M. Leo, T. D'Orazio, A. Caroppo, P. Spagnolo, C. Guaragnella, Unsupervised skin colour modelling for hand segmentation, The 5th IASTED International Conference on Visualization, Imaging, and Image Processing, 2005.
[115] O. Kayal, J. Samarabandu, Use of Kinect in a multicamera setup for action recognition application, IEEE 27th Canadian Conf. on Electrical and Computer Engineering (CCECE) 2014.