Workshop on Robot Perception of Humans Baden-Baden, Germany,
June 11, 2018. In conjunction with IAS-15

A Limb-based Approach for Body Pose Recognition Using a Predefined Set of Poses

Mattia Guidolin, Marco Carraro, Stefano Ghidoni, and Emanuele Menegatti

University of Padova, Intelligent Autonomous Systems Laboratory (IAS-Lab),
Via Ognissanti 72, 35129, Padova, Italy

Abstract. This paper proposes a novel approach for the body pose recognition of multiple persons. Our system takes as input the 3D joint locations of each person's skeleton representation, as estimated by OpenPTrack, an open source project for RGB-D people tracking. The poses of the upper and lower limbs are computed separately by comparing them to the ones stored in a pre-recorded database. The two partial poses are then combined to obtain the full pose of each person on the scene. The system provides real-time outcomes, is markerless, and does not need any assumption on the orientation, initial position, or number of persons on the scene. It can be used as a base for more complex action recognition algorithms, for intelligent surveillance and security devices, or in human-computer interaction.

Keywords: limb-based, body pose recognition, human-computer interaction, RGB-D, markerless, real-time

1 Introduction

Human pose and action recognition are fields with a large range of applications, from surveillance and security purposes, to controlling machines and interacting with computer-based systems. The identification of a person's pose eliminates the necessity to use external input devices, such as mice, keyboards, and joysticks, typically needed to interact with computers or robotic systems [13]. One of the main advantages of this fact is the possibility to interact with a system in a hands-free way. As another example, the knowledge of a person's pose can be used to increase the level of safety of dangerous machines that require the operator to behave in specific ways. If an unsafe pose of the operator is recognized, the system can be forced to shut down safely. Finally, action recognition algorithms usually need to know all the poses assumed in a sequence of frames, to be able to identify if a specific action is performed. Each action is described as a temporal sequence of distinct poses.

In order to recognize the current pose of a person, a description of his or her body joints has to be known. The simplest representation of a human body is the stick figure, which consists of line segments linked by joints [6]. Figure 1 shows the configuration of the joints used in our algorithm.

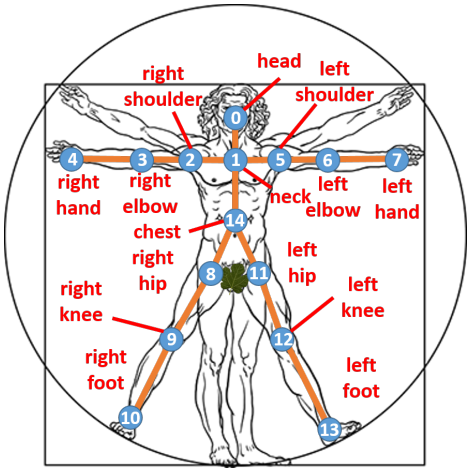

Fig. 1. The human model used in our system. Source: [3].

The estimation of the joint positions may be performed intrusively or non-intrusively. Intrusive manners include, for example, the usage of optical motion capture technologies (e.g., BTS Bioengineering, Vicon). These technologies require human subjects to wear optical markers mounted at the specified limb positions. The optical markers are tracked by custom-made cameras made of an Infrared (IR) pass-filter coupled with optical lenses. Although these technologies produce highly accurate estimation of 3D human pose, they are expensive, require extensive setup and are intrusive at best, thus making them not suitable for surveillance or human-computer interaction purposes [4, 6]. Furthermore, they usually work off-line, requiring manual inputs for association and disambiguation of the marker positions obtained by each camera. Recently, due to the introduction of low-cost RGB-D cameras (e.g., the Microsoft Kinect), markerless motion capture using camera networks has attracted the attention of many researchers (e.g., Zhang *et al.* [21], Asteriadis *et al.* [1], Munaro *et al.* [11]).

In this paper, we propose a pose recognition system that can use one or multiple RGB-D cameras to classify the poses of both the upper and lower limbs, separately, for each person present on the scene. The two partial poses are then combined to obtain the full pose. The system provides real-time outcomes, is markerless, and does not need any assumption on the orientation, initial position, or number of persons on the scene. It can be used as a base for more advanced action recognition algorithms that also take into account the temporal aspect, for intelligent surveillance and security devices to know if a person is behaving in a potentially dangerous way, in human-computer interaction, or for every other application that might benefit from the knowledge of the current pose of a person at a higher level with respect to the mere position of each joint. We plan to exploit our algorithm to perform an automatic ergonomics and posture evaluation for factory workers via the RULA (Rapid Upper Limb Assessment) method [9]. In

labor-intensive workplaces, it is important to monitor the employees' postures in order to prevent the onset of limb disorders. The maintenance of inadequate working postures for extended periods of time can lead to serious injuries. By providing a set of rules to calculate simple numeric scores, the RULA method allows to identify such inadequate poses. Our algorithm can be exploited to give a direct feedback to the workers, by computing an on-line automatic evaluation of such scores.

Our system is based on OpenPTrack, an open source project for RGB-D people tracking [3]. OpenPTrack is marker-less, multi-person, independent of background, and does not make any assumption on people appearance and initial pose. It can work using one or multiple RGB-D cameras and is able to estimate the 3D joint locations of each person present on the scene, by exploiting OpenPose, a real-time multi-person 2D pose estimation system [2]. Interested readers are referred to [3] for more detailed information on how the 3D joints are extracted and the detections from each camera are fused. The 3D joint locations estimated by OpenPTrack constitute the input of our pose recognition system. By normalizing the distances between each joint and comparing the obtained skeletons to the ones of a pre-recorded database of poses, the pose of each person is recognized. The database can contain any number of poses involving arms only, legs only, or full body, and has to be recorded by one or more persons.

Most of the previous works that take into account the whole body [7, 8, 10, 17], do not separate the upper and lower-limbs in their analysis. This does not match our way of identifying other people's poses as humans. Most of the times, the pose of a person is described by both their arms and legs, but separately. A person can be standing, and at the same time waving to a friend. The *standing* pose involves the legs only (i.e., the lower limbs), while the *waving* pose involves the arms only (i.e., the upper limbs). The full pose of such person is the combination of the two partial ones. With respect to other approaches that evaluate the pose of the whole body, our algorithm permits to achieve a computationally lighter system, and, at the same time, to decrease the number of different poses needed in the database. As an example, if we were only interested in knowing if a person has his or her right arm raised, with a "full-body" approach we should record in the database multiple poses where the person has the right arm raised and is standing, sitting, etc., or else the right arm raised would be recognized only in specific situations. Contrarily, by using our approach, the right arm raised can be detected independently of the legs pose, thus making it possible to record just a single *right_arm_up* pose in the database. Our system is focused on the instantaneous poses that can form a gesture, and does not take into account the temporal aspect. The goal is to assign a semantically meaningful label to a predefined set of poses, and extract this information automatically, in a robust way, and real-time. This gives the system a higher level knowledge of the person state, with respect to the mere 3D locations of each joint of the body. One of the advantages of our system is that it does not require many training data, compared to machine learning-based ones. We can easily add new poses to the database in a matter of minutes, by recording just a few pose samples.

The remainder of the paper is organized as follows. Section 2 reviews the literature regarding pose, action, and gesture recognition. Section 3 describes our approach to the pose recognition problem. Section 4 shows some experimental results. Section 5 presents our conclusions and the possible future works.

2 Related work

Over the past 20 years, a large variety of pose recognition algorithms have been developed. They differ both in the part of the body that is considered (hands, arms, full body), and on the technique chosen to obtain the input data (single or multiple RGB cameras, depth cameras, RGB-D cameras).

Utsumi *et al.* [16] proposed a hand pose recognition system that used multiple RGB cameras. They considered the hand's center of gravity, orientation, and fingertip points as feature points, and assumed that they were all placed on a plane that they called the "hand plane". Ng and Ranganath [12] proposed another approach to hand gesture recognition. They decomposed the task of gesture recognition by first identifying the separate hand poses. The pose information was then incorporated with hand motion to recognize gestures from image sequences.

Huo *et al.* [5] presented an approach to capture human motions without markers. They used feature points for the purpose of pose classification. However, they limited their analysis to hands and torso only. Song *et al.* [15] developed a multi-signal gesture recognition system that attended to both bodies and hands. They performed 3D upper body pose estimation and hand pose classification together. However, they assumed the subject to be standing 50 feet away from the camera.

Wan and Sawada [18] used a vision-based human motion capture system to obtain a 3D motion measurement of the human upper body, for the purpose of gesture recognition. The usage of a motion capture system required the user to wear 7 reflective markers. Sigalas *et al.* [14] employed a 9 parameter model to track both arms (4 parameters for each arm) as well as the orientation of the torso. They used the information to develop a vision-based gesture recognition system. Weng and Fu [20] chose to use a Time of Flight (ToF) camera as their input device, and estimated upper body poses. The estimated poses were then used to recognize a set of six different actions. Van den Bergh *et al.* [17] proposed both a 2D system based on silhouettes of the user, and a 3D system based on visual hulls, to classify a set of human poses. The full body was considered, but in the 50 pose classes defined, the user was always standing, thus focusing the analysis on the upper body pose.

Li *et al.* [8] presented a method to recognize human actions that used silhouettes from sequences of depth maps. They focused on a small set of representative 3D points sampled from the depth map, and then compared the obtained data to a database of 20 actions performed by multiple subjects. Munaro *et al.* [10] proposed a system for real-time human action recognition based on 3D motion flow estimation. They used colored point clouds as input, and classified 15 different

actions. Kim *et al.* [7] developed human pose estimation and gesture recognition algorithms that used only depth information. Several key frames were extracted from the input data, and then compared with key frames of registered gestures. However, they assumed that all gestures were made toward the sensor within the human pose estimation range (±30 degrees). Wang *et al.* [19] proposed an approach to recognize an action from video frames. They first estimated the joint locations, then grouped the estimated joints into five body parts (e.g. left arm, etc.), and finally applied data mining techniques in the spatial domain to obtain sets of distinctive co-occurring spatial configurations of body parts. Their approach on the separation of the different body parts is the closest to our work. However, they limited their analysis to videos only, and the computation time of their algorithm was not specified.

3 System design

Figure 2 shows an overview of the proposed system. It can be split in two parts: i) manipulation of the estimated joints, and ii) recognition of the upper and lower-limbs poses. The manipulation is necessary for allowing the system to recognize the pose of people with different body characteristics and oriented at different angles with respect to the world reference. The recognition is accomplished by comparing the manipulated joint locations to the ones recorded for each pose in the database. In particular, the recognition of the current upper-limbs pose of a person is achieved by calculating the Euclidean distances between the links that describe the arms and each upper-limbs pose present in the database. Similarly, the lower-limbs pose is recognized by calculating the distances between the links that describe the legs and each lower-limbs pose in the database. The final output of our system is a label describing the current upper-limbs pose of each person (if a match is found), the current lower-limbs pose (if a match is found), or a combination of the two (if both upper-limbs and lower-limbs matches are found at the same time).

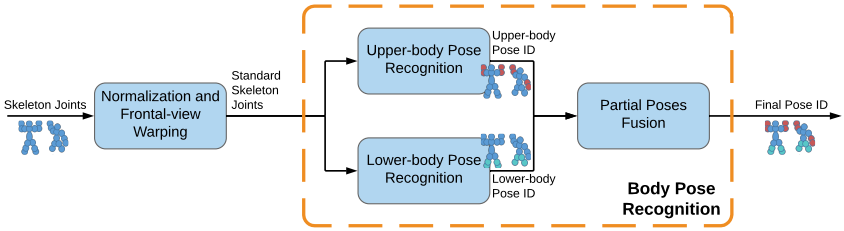

Fig. 2. System overview. The 3D joint locations obtained from OpenPTrack are normalized and rotated in order to have a frontal view of each person. The arms and the legs joints are then compared, separately, with the corresponding poses recorded in the database. The two recognized poses are merged to obtain the final result.

3.1 Skeleton manipulation

The output of the pose estimation system is a set of skeleton tracks for each person on the scene. For a specific frame, we can define the skeleton track related to person k as:

$$\boldsymbol{S}_k = \{\boldsymbol{J}_0, \boldsymbol{J}_1, \ldots, \boldsymbol{J}_{14}\}$$

where $\boldsymbol{J}_0, \ldots, \boldsymbol{J}_{14}$ are the 3D coordinates of the joints that describe the human model depicted in Figure 1. The set of links $\boldsymbol{L}$ that connect each pair of joints is defined as:

$$\begin{aligned}\boldsymbol{L} &= \{\boldsymbol{l}_0, \boldsymbol{l}_1, \ldots, \boldsymbol{l}_{13}\} \\ &= \{(\boldsymbol{J}_{14}, \boldsymbol{J}_1), (\boldsymbol{J}_1, \boldsymbol{J}_2), (\boldsymbol{J}_2, \boldsymbol{J}_3), (\boldsymbol{J}_3, \boldsymbol{J}_4), (\boldsymbol{J}_1, \boldsymbol{J}_5), \\ &\quad (\boldsymbol{J}_5, \boldsymbol{J}_6), (\boldsymbol{J}_6, \boldsymbol{J}_7), (\boldsymbol{J}_{14}, \boldsymbol{J}_8), (\boldsymbol{J}_8, \boldsymbol{J}_9), (\boldsymbol{J}_9, \boldsymbol{J}_{10}), \\ &\quad (\boldsymbol{J}_{14}, \boldsymbol{J}_{11}), (\boldsymbol{J}_{11}, \boldsymbol{J}_{12}), (\boldsymbol{J}_{12}, \boldsymbol{J}_{13}), (\boldsymbol{J}_1, \boldsymbol{J}_0)\}\end{aligned}$$

where $(\boldsymbol{J}_i, \boldsymbol{J}_j)$ indicates the vector connecting joint $\boldsymbol{J}_i$ to joint $\boldsymbol{J}_j$.

Since each person can have different height and limb lengths, as well as a different orientation with respect to the origin, the skeletons have to be normalized and rotated. The final output of this process is a set of normalized skeleton tracks in which each segment has length equal to 1 and whose orientation is always the same.

Normalization The first step is the normalization of each vector in $\boldsymbol{L}$. Each link length is normalized, while maintaining its original direction. The normalized skeleton track is constructed by starting from the chest joint ($\boldsymbol{J}_{14}$) and connecting the normalized vectors.

The result of the normalization process is a set of normalized skeleton tracks (one for each person k) in which each link has length equal to 1:

$$\overline{\boldsymbol{S}}_k = \left\{\overline{\boldsymbol{J}}_0, \overline{\boldsymbol{J}}_1, \ldots, \overline{\boldsymbol{J}}_{14} \mid \forall (i,j) \in \boldsymbol{L}, \|\overline{\boldsymbol{J}}_i - \overline{\boldsymbol{J}}_j\| = 1\right\}$$

where $\overline{\boldsymbol{J}}_0, \ldots, \overline{\boldsymbol{J}}_{14}$ are the new 3D coordinates of the joints after the normalization process.

Frontal-view warping The normalized skeleton can now be rotated in order to obtain a frontal view, independently on the real orientation of the person.

The orientation $\boldsymbol{o}$ of each person is computed as the normalized cross product between the vector connecting the chest joint ($\boldsymbol{J}_{14}$) to the left shoulder ($\boldsymbol{J}_5$), and the vector connecting the chest joint ($\boldsymbol{J}_{14}$) to the right shoulder ($\boldsymbol{J}_2$):

$$\boldsymbol{o} = \frac{(\boldsymbol{J}_{14} - \boldsymbol{J}_5) \times (\boldsymbol{J}_{14} - \boldsymbol{J}_2)}{\|(\boldsymbol{J}_{14} - \boldsymbol{J}_5) \times (\boldsymbol{J}_{14} - \boldsymbol{J}_2)\|}$$

Once the orientation of the person is known, the angle α between the orientation and the x axis can be computed as the arccosine of the dot product between the orientation $\boldsymbol{o}$ and the x axis unit vector:

$$\alpha = \arccos(\boldsymbol{o} \cdot (1,0,0)^T)$$

By rotating the normalized skeleton of an amount equal to $-\alpha$, and translating each joint $\overline{\boldsymbol{J}}_i$ of an amount equal to $-\overline{\boldsymbol{J}}_{14}$, we obtain the frontal view of each person k, with the chest joint always located on the origin $(0, 0, 0)$:

$$\hat{\boldsymbol{S}}_k = \left\{ \hat{\boldsymbol{J}}_i = \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} (\overline{\boldsymbol{J}}_i - \overline{\boldsymbol{J}}_{14}), 0 \leq i \leq 14 \right\}$$

The result is a set of normalized and conveniently rotated and translated skeleton tracks $\hat{\boldsymbol{S}}_k$.

3.2 Pose recognition

Of the 14 links describing each skeleton, only 4 are taken into account in the pose recognition algorithm. Such links are the ones connecting the elbows to the wrists ($\boldsymbol{l}_3 = (\boldsymbol{J}_3, \boldsymbol{J}_4)$ for the right arm and $\boldsymbol{l}_6 = (\boldsymbol{J}_6, \boldsymbol{J}_7)$ for the left arm) for the upper limbs, and the ones connecting the knees to the ankles ($\boldsymbol{l}_9 = (\boldsymbol{J}_9, \boldsymbol{J}_{10})$ for the right leg and $\boldsymbol{l}_{12} = (\boldsymbol{J}_{12}, \boldsymbol{J}_{13})$ for the left leg) for the lower limbs. The signatures for the upper and lower-limbs poses, defined as the links taken into account for the recognition of the poses, are respectively:

$$\hat{\boldsymbol{\Sigma}}_{\boldsymbol{u}} = \left\{ \hat{\boldsymbol{l}}_3, \hat{\boldsymbol{l}}_6 \right\}, \quad \hat{\boldsymbol{\Sigma}}_{\boldsymbol{l}} = \left\{ \hat{\boldsymbol{l}}_9, \hat{\boldsymbol{l}}_{12} \right\}$$

The database of recorded poses is defined as:

$$\boldsymbol{D} = \boldsymbol{D}_{\boldsymbol{u}} + \boldsymbol{D}_{\boldsymbol{l}} = \{ \boldsymbol{\Sigma}_1^*, \boldsymbol{\Sigma}_2^*, \ldots, \boldsymbol{\Sigma}_n^* \}$$

where each $\boldsymbol{\Sigma}^* \in \boldsymbol{D}$ can be the signature of an upper-limbs pose or a lower-limbs pose, and n is the number of recorded poses. $\boldsymbol{D}_{\boldsymbol{u}}$ indicates the subset containing all the n_u upper-limbs poses, and $\boldsymbol{D}_{\boldsymbol{l}}$ the one containing all the n_l lower-limbs poses ($n = n_u + n_l$).

The upper-limbs pose scores are computed by calculating the Euclidean distances $\boldsymbol{r}_{\boldsymbol{u},i}$ between the vectors of the upper-limbs signature $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{u}}$ and the corresponding vectors describing each upper-limbs pose $\boldsymbol{\Sigma}_i^* = \{\boldsymbol{l}_{3,i}^*, \boldsymbol{l}_{6,i}^*\} \in \boldsymbol{D}_{\boldsymbol{u}}$ in the database. Similarly, the lower-limbs pose scores are computed by calculating the Euclidean distances $\boldsymbol{r}_{\boldsymbol{l},j}$ between the vectors of the lower-limbs signature $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{l}}$ and the corresponding vectors describing each lower-limbs pose $\boldsymbol{\Sigma}_j^* = \{\boldsymbol{l}_{9,j}^*, \boldsymbol{l}_{12,j}^*\} \in \boldsymbol{D}_{\boldsymbol{l}}$ in the database:

$$\forall \boldsymbol{\Sigma}_i^* \in \boldsymbol{D}_{\boldsymbol{u}}, \boldsymbol{r}_{\boldsymbol{u},i} = \{ \|\hat{\boldsymbol{l}}_3 - \boldsymbol{l}_{3,i}^*\|, \|\hat{\boldsymbol{l}}_6 - \boldsymbol{l}_{6,i}^*\| \}$$
$$\forall \boldsymbol{\Sigma}_j^* \in \boldsymbol{D}_{\boldsymbol{l}}, \boldsymbol{r}_{\boldsymbol{l},j} = \{ \|\hat{\boldsymbol{l}}_9 - \boldsymbol{l}_{9,j}^*\|, \|\hat{\boldsymbol{l}}_{12} - \boldsymbol{l}_{12,j}^*\| \}$$

This yields two scores for each upper-limbs pose $(\boldsymbol{r}_{\boldsymbol{u},i}(0), \boldsymbol{r}_{\boldsymbol{u},i}(1))$, and two scores for each lower-limbs pose $(\boldsymbol{r}_{\boldsymbol{l},j}(0), \boldsymbol{r}_{\boldsymbol{l},j}(1))$. The final scores $r_{u,i}^*, r_{l,j}^*$ are selected as the maximum values between the two obtained:

$$r_{u,i}^* = \max_{\boldsymbol{r}_{\boldsymbol{u},i}} \{ \boldsymbol{r}_{\boldsymbol{u},i}(0), \boldsymbol{r}_{\boldsymbol{u},i}(1) \}$$
$$r_{l,j}^* = \max_{\boldsymbol{r}_{\boldsymbol{l},j}} \{ \boldsymbol{r}_{\boldsymbol{l},j}(0), \boldsymbol{r}_{\boldsymbol{l},j}(1) \}$$

If at least one pose i exists such that $r^*_{u,i} < threshold$, then the upper-limbs pose is selected as the one that gives the lowest score. If no pose such that $r^*_{u,i} < threshold$ is found, then the upper-limbs pose is labeled as *unknown*. Similarly, if at least one pose j exists such that $r^*_{l,j} < threshold$, then the lower-limbs pose is selected as the one that gives the lowest score. If no pose such that $r^*_{l,j} < threshold$ is found, then the lower-limbs pose is labeled as *unknown*. The final pose of each person can be the recognized upper-limbs pose (if there is any), the recognized lower-limbs pose (if there is any), or the combination of the two (if both upper and lower-limbs poses are found). Figure 3 shows the output of our system. In this specific case the upper-limbs pose is recognized as *arms_down*, the lower-limbs pose as *standing*, and the final pose is labeled as *standing_with_arms_down*.

Fig. 3. Skeleton joints of a person and the corresponding body pose label, as shown in our system. The upper-limbs pose is recognized as *arms_down*, and the lower-limbs pose as *standing*. The system provides real-time output.

The availability of accurate 3D joint positions makes it possible to classify poses without the need of machine learning methods. The main advantage of this fact is that the typical amount of training data of machine learning-based methods is not needed. This means that the addition of a new pose in the database is an extremely simple task. It requires just a single person to record few pose samples that describe the new pose.

4 Experiments

For our experiments we used a database of 8 poses (6 for the upper limbs and 2 for the lower limbs) containing:

- pose 0 $\in \boldsymbol{D_u}$: *arms_down*
- pose 1 $\in \boldsymbol{D_u}$: *arms_up*

- pose 2 $\in \boldsymbol{D_u}$: *right_arm_up*
- pose 3 $\in \boldsymbol{D_u}$: *left_arm_up*
- pose 4 $\in \boldsymbol{D_u}$: *right_arm_pointing*
- pose 5 $\in \boldsymbol{D_u}$: *right_arm_grasping*
- pose 6 $\in \boldsymbol{D_l}$: *standing*
- pose 7 $\in \boldsymbol{D_l}$: *squatting*

The difference between *pose 4 (right_arm_pointing)* and *pose 5 (right_arm_grasping)* is that in the first one the right arm should be approximately parallel to the floor, while in the second one the right arm should be slightly bent towards the floor.

Our test setup consists of 3 Kinects, but, in general, any number of Kinects can be used. The Kinects are located at three corners of an area of approximately 6 by 4 meters, and connected to 3 PCs which are placed in a network. Two persons have been asked to assume some of the poses present in the database, simultaneously. For every frame in which the persons are assuming one of the recorded poses, the recognized pose has been compared to the groundtruth obtained by manually labeling each pose. The performance of our system has been evaluated by calculating the values of precision P, accuracy A, and recall R for each pose, defined as:

$$P = \frac{TP}{TP + FP}, \quad A = \frac{TP + TN}{TP + TN + FP + FN}, \quad R = \frac{TP}{TP + FN}$$

where TP is the number of true positives, TN the number of true negatives, FP the number of false positives, and FN the number of false negatives.

The precision index does not take into account the number of false negatives. It can assume high values even if the poses are not recognized most of the times, if the number of false positives is low. On the other hand, the accuracy index takes into account both the number of true positives and of true negatives. It can still assume high values even if the poses are not recognized most of the times, if the number of true negatives is high. Finally, the recall index describes the ratio between the number of times the pose is correctly recognized and the actual number of times the pose has been assumed. Hence, the recall is the most significant indicator for evaluating the performance of our system.

First experiment In the first experiment the persons assumed all the different upper-limbs poses, while standing. Table 1 shows the confusion matrix obtained. An overall precision of 0.956 is achieved, with an accuracy of 0.966 and a recall of 0.9252.

We can see that the most critical poses are *pose 2 (right_arm_up)* and *pose 5 (right_arm_grasping)*. The poor recognition of *pose 2* was caused by an error of the skeletal tracker: the left arm of one of the two persons was not correctly estimated. On the other hand, while *pose 5* was correctly recognized for one person, for the second one it was sometimes mistaken with *pose 0 (arms_down)*. This happened because, when asked to switch from *right_arm_pointing* to *right_arm_grasping*, one of the persons bent the arm way more than expected ($>$ 45

Table 1. Confusion matrix of the first experiment. The last three columns indicate the values of precision, accuracy, and recall for each assumed pose.

	Groundtruth									
	pose 0	pose 1	pose 2	pose 3	pose 4	pose 5	pose 6	P	A	R
pose 0	150	0	0	1	0	76	0	0.6608	0.9252	1
pose 1	0	156	0	0	0	0	0	1	1	1
pose 2	0	0	121	0	0	0	0	1	0.9524	0.7118
pose 3	0	0	0	197	0	0	0	1	0.9835	0.9206
pose 4	0	0	0	0	109	0	0	1	0.9893	0.9083
pose 5	0	0	0	0	11	143	0	0.9286	0.9146	0.6500
pose 6	0	0	0	0	0	0	1030	1	1	1
unknown	0	0	49	17	0	0	0			

degrees). In such case, the pose was more similar to *arms_down*, rather than *right_arm_grasping*. This fact also explains the low precision value achieved for *pose 0*.

Second experiment In the second experiment the persons varied both the upper and lower-limbs poses, alternatively. Table 2 shows the confusion matrix obtained. An overall precision of 1 is achieved, with an accuracy of 0.9243 and a recall of 0.8107. This value of the precision index is due to the absence of false positives in the experiment.

Table 2. Confusion matrix of the second experiment. The last three columns indicate the values of precision, accuracy, and recall for each assumed pose.

	Groundtruth							
	pose 0	pose 2	pose 3	pose 6	pose 7	P	A	R
pose 0	90	0	0	0	0	1	1	1
pose 2	0	368	0	0	0	1	0.9577	0.9064
pose 3	0	0	316	0	0	1	0.9042	0.7861
pose 6	0	0	0	454	0	1	1	1
pose 7	0	0	0	0	228	1	0.7595	0.5135
unknown	0	38	86	0	216			

The most critical pose is *pose 7 (squatting)*. The poor results were due to difficult conditions for the tracking system. The squatting position is a particularly challenging one, because the legs are partially occluded. For the second person, who could be seen better by the camera network, the pose was correctly recognized, while, for the first person, the tracker gave a wrong estimation of the legs position, thus invalidating the pose recognition process. As of *pose 3 (left_arm_up)*, the lower recall value, with respect to other poses, is due to intra-class

variability issues. When asked to raise the left arm, one of the persons kept the arm lower than the other, and the recognition of the pose failed in some frames.

Figure 4 shows four of the poses assumed in the experiments. The poses were correctly classified even when the persons were partially occluded, thanks to the presence of multiple cameras. In Figure 4a and 4b the two persons were asked to point (a) and grasp (b) an imaginary object with the right hand, while standing. In Figure 4c they were asked to simply raise the right arm, whereas in Figure 4d they were asked to squat, while keeping the right arm raised. In this last case, our system correctly recognized the pose assumed by person 2 (green skeleton) as *squatting_with_right_arm_up*, while it partially failed for person 1 (brown skeleton). This was due to a wrong estimation of the joints describing the legs of the person. Still, the upper-limbs pose was correctly recognized as *right_arm_up*.

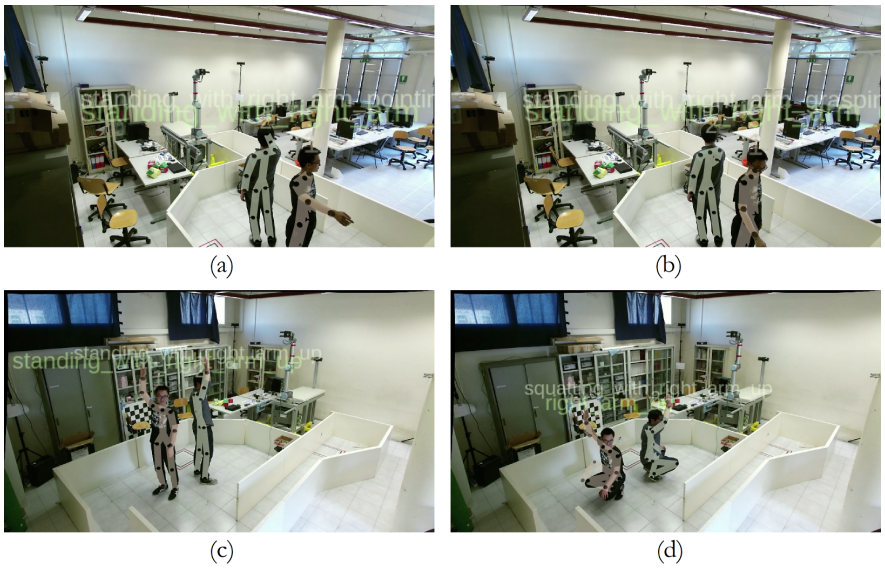

Fig. 4. Examples of correct (a, b, c) and partially correct (d) classification of some of the poses assumed during the experiments: (a) *standing_with_right_arm_pointing*, (b) *standing_with_right_arm_grasping*, (c) *standing_with_right_arm_up*, (d) *squatting_with_right_arm_up*. The multi-camera system allows to correctly recognized the poses even when the persons are partially occluded.

The outcomes of the experiments have shown that our system produces accurate results in most of the situations. Even visually similar but conceptually different poses, like *right_arm_pointing* and *right_arm_grasping*, can be recognized. The main source of errors in the pose recognition process came from the wrong estimations of the 3D joint locations in particularly challenging conditions.

5 Conclusions and future works

The idea to decouple the recognition of the upper and lower limbs poses has been proven to give promising results. Nevertheless, our pose recognition algorithm is still a work in progress. In the future we plan to develop our system by including more links in both the upper and lower-limbs signatures, thus taking into account also the links connecting the shoulders to the elbows and the hips to the knees. In such case, the new signatures will be:

$$\hat{\boldsymbol{\Sigma}}_{\boldsymbol{u}} = \left\{\hat{\boldsymbol{l}}_2, \hat{\boldsymbol{l}}_3, \hat{\boldsymbol{l}}_5, \hat{\boldsymbol{l}}_6\right\}, \quad \hat{\boldsymbol{\Sigma}}_{\boldsymbol{l}} = \left\{\hat{\boldsymbol{l}}_8, \hat{\boldsymbol{l}}_9, \hat{\boldsymbol{l}}_{11}, \hat{\boldsymbol{l}}_{12}\right\}$$

Increasing the dimension of the signatures will make it possible to recognize a larger variety of poses. With such configuration it will be feasible to register a complex database of poses, more similar to the natural ones that we assume everyday. An evaluation of the differences in robustness before and after the modification will also be necessary. The possibility to implement such system is strongly dependent on the reliability of the skeletal tracker. Since the number of links considered for the recognition is increasing, also the chance to have wrong joint estimations rises. Another challenging aspect is the intra-class variability in the human poses, since "raising the arms" can mean slightly different poses for different persons, i.e., arms straight up, or half-way up, etc. A low number of links for the recognition of the pose behaves like a filter that smooths the intra-class variability. Thus, increasing the number of links is likely to produce a higher number of false negatives.

After finding the best trade-off between the accuracy of the system and the number of evaluated links, and developing a more sophisticated pose recognition algorithm, we plan to use it as input for an action recognition system. The action recognition algorithm will take into account also the time component, by analyzing sequences of poses, and not only single frames.

References

1. Asteriadis, S., Chatzitofis, A., Zarpalas, D., Alexiadis, D.S., Daras, P.: Estimating human motion from multiple kinect sensors. In: Proceedings of the 6th international conference on computer vision/computer graphics collaboration techniques and applications. p. 3. ACM (2013)
2. Cao, Z., Simon, T., Wei, S.E., Sheikh, Y.: Realtime multi-person 2d pose estimation using part affinity fields. In: CVPR. vol. 1, p. 7 (2017)
3. Carraro, M., Munaro, M., Burke, J., Menegatti, E.: Real-time marker-less multi-person 3d pose estimation in rgb-depth camera networks. arXiv preprint arXiv:1710.06235 p. (in press) (2017), in Proceedings of the IAS-15 Conference
4. Hen, Y.W., Paramesran, R.: Single camera 3d human pose estimation: A review of current techniques. In: Technical Postgraduates (TECHPOS), 2009 International Conference for. pp. 1–8. IEEE (2009)
5. Huo, F., Hendriks, E., Paclik, P., Oomes, A.H.: Markerless human motion capture and pose recognition. In: Image Analysis for Multimedia Interactive Services, 2009. WIAMIS'09. 10th Workshop on. pp. 13–16. IEEE (2009)

6. Jain, H.P., Subramanian, A., Das, S., Mittal, A.: Real-time upper-body human pose estimation using a depth camera. In: International Conference on Computer Vision/Computer Graphics Collaboration Techniques and Applications. pp. 227–238. Springer (2011)
7. Kim, H., Lee, S., Lee, D., Choi, S., Ju, J., Myung, H.: Real-time human pose estimation and gesture recognition from depth images using superpixels and svm classifier. Sensors 15(6), 12410–12427 (2015)
8. Li, W., Zhang, Z., Liu, Z.: Action recognition based on a bag of 3d points. In: Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on. pp. 9–14. IEEE (2010)
9. McAtamney, L., Corlett, E.N.: Rula: a survey method for the investigation of work-related upper limb disorders. Applied ergonomics 24(2), 91–99 (1993)
10. Munaro, M., Ballin, G., Michieletto, S., Menegatti, E.: 3d flow estimation for human action recognition from colored point clouds. Biologically Inspired Cognitive Architectures 5, 42–51 (2013)
11. Munaro, M., Basso, F., Menegatti, E.: Openptrack: Open source multi-camera calibration and people tracking for rgb-d camera networks. Robotics and Autonomous Systems 75, 525–538 (2016)
12. Ng, C.W., Ranganath, S.: Gesture recognition via pose classification. In: Pattern Recognition, 2000. Proceedings. 15th International Conference on. vol. 3, pp. 699–704. IEEE (2000)
13. Patras, L., Giosan, I., Nedevschi, S.: Body gesture validation using multi-dimensional dynamic time warping on kinect data. In: Intelligent Computer Communication and Processing (ICCP), 2015 IEEE International Conference on. pp. 301–307. IEEE (2015)
14. Sigalas, M., Baltzakis, H., Trahanias, P.: Gesture recognition based on arm tracking for human-robot interaction. In: Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on. pp. 5424–5429. IEEE (2010)
15. Song, Y., Demirdjian, D., Davis, R.: Tracking body and hands for gesture recognition: Natops aircraft handling signals database. In: Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on. pp. 500–506. IEEE (2011)
16. Utsumi, A., Miyasato, T., Kishino, F.: Multi-camera hand pose recognition system using skeleton image. In: Robot and Human Communication, 1995. RO-MAN'95 TOKYO, Proceedings., 4th IEEE International Workshop on. pp. 219–224. IEEE (1995)
17. Van den Bergh, M., Koller-Meier, E., Van Gool, L.: Real-time body pose recognition using 2d or 3d haarlets. International journal of computer vision 83(1), 72–84 (2009)
18. Wan, K., Sawada, H.: Gesture recognition based on the probability distribution of arm trajectories. SICE Journal of Control, Measurement, and System Integration 2(5), 263–270 (2009)
19. Wang, H., Schmid, C.: Action recognition with improved trajectories. In: Computer Vision (ICCV), 2013 IEEE International Conference on. pp. 3551–3558. IEEE (2013)
20. Weng, E.J., Fu, L.C.: On-line human action recognition by combining joint tracking and key pose recognition. In: Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on. pp. 4112–4117. IEEE (2012)
21. Zhang, L., Sturm, J., Cremers, D., Lee, D.: Real-time human motion tracking using multiple depth cameras. In: Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on. pp. 2389–2395. IEEE (2012)