2013 IEEE RO-MAN: The 22nd IEEE International Symposium on
Robot and Human Interactive Communication
Gyeongju, Korea, August 26-29, 2013

Invariant Representation for User Independent Motion Recognition

Matteo Saveriano and Dongheui Lee

Abstract— **Human gesture recognition is of importance for smooth and efficient human robot interaction. One of difficulties in gesture recognition is that different actors have different styles in performing even same gestures. In order to move towards more realistic scenarios, a robot is required to handle not only different users, but also different view points and noisy incomplete data from onboard sensors on the robot. Facing these challenges, we propose a new invariant representation of rigid body motions, which is invariant to translation, rotation and scaling factors. For classification, Hidden Markov Models based approach and Dynamic Time Warping based approach are modified by weighting the importances of body parts. The proposed method is tested with two Kinect datasets and it is compared with another invariant representation and a typical non-invariant representation. The experimental results show good recognition performance of our proposed approach.**

I. Introduction

Understanding human intentions is a key aspect in human-robot interaction. If the robot understands what the human is doing, then it can select and adapt its behaviours coherently with human's needs. The first step to understand human intentions is to recognise human actions. In [1], [2], for example, human motion recognition is used to decide which is the best action the robot should perform.

We aim at making the robot capable of recognizing actions, performed by several people, in a daily life scenario, as shown in Fig. 1. Some issues arises from this goal. First, the same gesture can be performed by different people in slightly different manners and from different view points. Thus, it is desirable to have action recognition algorithm which is coordinate-free and invariant to the body sizes of actors. Second desirable property is the capability to handle continuous incoming data, since sensory inputs in real scenario are not segmented a prior (in other words, starting and ending points of each gesture is unknown a prior). Last, although existing commercial motion capturing systems can provide fast and accurate tracking performance, they often requires obtrusive settings in the environment and on the human subject (e.g., markers or special suits). This artificial setting makes hard to change the target person among different users in different environments. Thus, in this work, we use the Microsoft Kinect sensor for human motion tracking, although its data are less accurate and more noisy.

The main focus of this work is given to a new invariant representation for user independent gesture recognition. This problem has been extensively studied during the last decade. In [3], [4] invariants under *affine* and *projective transformations* are proposed. These invariants can be directly computed from the image coordinate, but one has to track the same needed five points for the duration of the motion. 3D invariants under *affine transformations* are proposed in [5], [6]. In this case, the same six points should be tracked in all the frames. However, continuous tracking of six points per each rigid body in a full human body is challenging in a daily life scenario. In our proposed approach we do not need to keep track of such a large number of same points, but we just need the position and the orientation of the rigid body.

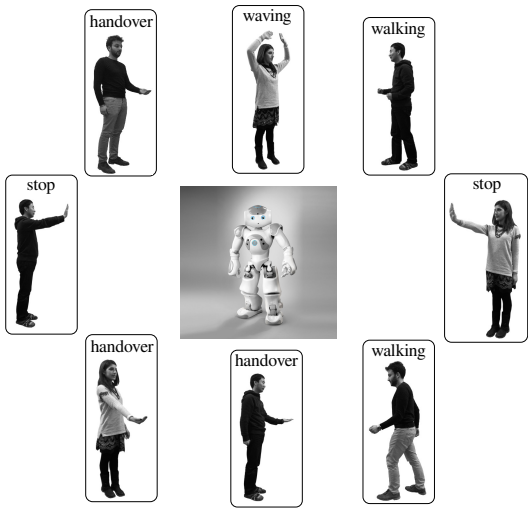


Fig. 1. Gesture recognition in a daily-life scenario.

A complete set (6 elements) of invariants, computed by the meaning of *Instantaneous Screw Axes* is adopted in [7] to recognise the motions of a rigid body. To make this representations invariant to linear and angular scale, one needs to divide the invariants by two scaling factors depending on the duration of the gesture. On the contrary, our representation is linear scale invariant independently on the gesture duration.

In [8] the full-body gestures recognition problem is solved using the *4D Action Feature Model*, consisting of 4D action shape (sequence of 3D shapes) with spatio-temporal features attached. Even if it is possible to recognise actions from a single view, multi-view video sequences are required to construct the *4D Action Feature Model* during the training procedure. In contrast, our work can be applicable with one Kinect sensory data for both training and recognition, which can allow online incremental learning for a robot.

In [9], [10] the authors propose to represent the motion with the *spatio-temporal curvature* of the trajectory. The curvature is useful to detect the so-called *dynamic instants*,

Authors are with Fakultät für Elektrotechnik und Informationstechnik, Technische Universität München, Munich, Germany matteo.saveriano@tum.de, dhlee@tum.de.

978-1-4799-0509-6/13/$31.00 ©2013 IEEE

points in which there are changes in the direction of motion and/or in the velocity. Their choice is motivated by some psychological evidence stating that humans perceive actions through the *atomic units* of actions, defined as motion events due to the significant changes in motion trajectory [11]. They compute the curvature of 2D trajectories and prove its invariance with respect to the point of view. However the invariance to linear scaling factors is not proved. Compared to that, our representation can handle 3D data and it is invariant to linear scaling factors, translation and rotation.

In [12] a technique to recognise motion from Kinect data is proposed. First, the distances between all the normalised positions of the human parts, are computed. Then, the action is partitioned into a temporal pyramid and the Fourier Transform is computed for each segment at each level. Choosing the low-frequency Fourier coefficient as features, the authors obtain a representation that is invariant to the view and to linear scaling factors, and that is robust to noise and temporal misalignment. An SVM model is trained on these data and is used to recognise gestures. Hereinafter we refer to this approach as *Fourier Temporal Pyramid (FTP) features*.

Our main contribution is to propose a new compact representation of a 3D motion, which is invariant to rotations, translations and linear scaling factors. This invariant representation, computed from the position and the orientation of a rigid body, is used to create a set of features useful to recognise articulated whole-body gestures. The 3D motion representation is tested with two classification algorithms. The classification methods take the weight of body parts into consideration, in order to actively evaluate the motions of relevant and irrelevant body parts. We tested our approach on a continuous dataset in which the gestures segmentation is unknown. Further it is compared with the *FTP features* approach [12] on the *MSR-Action3D* dataset used in [12] and a typical non-invariant representation.

The rest of the paper is organised as follows. Section II describes the set of invariant features proposed in our approach. Section III gives some details about data collection and filtering. Section IV explains how Dynamic Time Warping and Hidden Markov Models are used to classify human gestures. Section V presents the experimental results. Section VI states the conclusions and the future works.

II. Invariant Representation of Rigid Motions

According to [9], [10] curvature is useful to detect the *dynamic instants* that characterise a gesture. We propose a new representation, which is view and linear scale invariant, and capable of capturing changes in the direction of motion and/or in the velocity of a 3D motion. Invariance to roto-translation of the reference frame (or of the user pose) helps us to recognise the motion independently of the initial relative poses of the actor with respect to the robot sensor. Invariance to linear scaling factors helps us to cope with kinematics differences between different users. The proposed representation is also compact, because 3D positions and orientations[1] are represented with two scalar values for each time instant.

A. Invariant representation of translation

Given the position of a rigid body during the time $\boldsymbol{r}(t) = [x(t)\ y(t)\ z(t)]^T$, we define an invariant representation of the translation as

$$\gamma(t) = \frac{\|\dot{\boldsymbol{r}}(t) \times \ddot{\boldsymbol{r}}(t)\|}{\|\dot{\boldsymbol{r}}(t)\|^2} = \kappa(t)\, \|\dot{\boldsymbol{r}}(t)\|\,, \tag{1}$$

where $\dot{\boldsymbol{r}}(t)$ and $\ddot{\boldsymbol{r}}(t)$ represents respectively the first and second order time derivatives of $\boldsymbol{r}(t)$, and κ is the *curvature*. This representation is invariant under constant rotation, translation and linear scale.

Proof: Let's assume that $\boldsymbol{r}(t)$ is transformed using a constant transformation $T = (\alpha\boldsymbol{R}, \boldsymbol{t})$, where $\alpha \neq 0$ is a scaling factor, $\boldsymbol{t} = [t_x\ t_y\ t_z]^T$ is a translation and $\boldsymbol{R}$ is a rotation matrix. We have $\boldsymbol{r}_T(t) = \alpha\boldsymbol{R}\boldsymbol{r}(t) + \boldsymbol{t}$, $\dot{\boldsymbol{r}}_T(t) = \alpha\boldsymbol{R}\dot{\boldsymbol{r}}(t)$ and $\ddot{\boldsymbol{r}}_T(t) = \alpha\boldsymbol{R}\ddot{\boldsymbol{r}}(t)$. Recalling that, given two vectors $\boldsymbol{v}, \boldsymbol{w} \in \mathbb{R}^3$, $(\boldsymbol{R}\boldsymbol{v}) \times (\boldsymbol{R}\boldsymbol{w}) = \boldsymbol{R}(\boldsymbol{v} \times \boldsymbol{w})$, $\|\boldsymbol{R}\boldsymbol{v}\| = \|\boldsymbol{v}\|$ and $\|\boldsymbol{v} \times \boldsymbol{w}\| = \|\boldsymbol{v}\|\, \|\boldsymbol{w}\|\, sin(\theta)$, it is easy to verify that:

$$\gamma_T(t) = \frac{\|\dot{\boldsymbol{r}}_T(t) \times \ddot{\boldsymbol{r}}_T(t)\|}{\|\dot{\boldsymbol{r}}_T(t)\|^2} = \frac{\left\|\alpha^2\boldsymbol{R}[\dot{\boldsymbol{r}}(t) \times \ddot{\boldsymbol{r}}(t)]\right\|}{\|\alpha\boldsymbol{R}\dot{\boldsymbol{r}}(t)\|^2} = \gamma(t).$$

B. Invariant representation of rotation

An invariant representation of the rotation is defined by

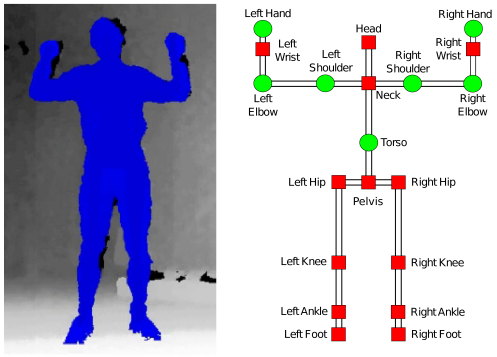
$$\xi(t) = \frac{\|\boldsymbol{\omega}(t) \times \dot{\boldsymbol{\omega}}(t)\|}{\|\boldsymbol{\omega}(t)\|^2}, \tag{2}$$

where $\boldsymbol{\omega}(t) = [\omega_x(t)\ \omega_y(t)\ \omega_z(t)]^T$ represents the angular velocity trajectory of the rigid body, expressed in a fixed frame, and $\dot{\boldsymbol{\omega}}(t)$ its time derivative. We can prove this proposed representation (2) is invariant to roto-translations, since translation does not affect the angular velocity and rotation affects neither the norm of a vector nor the angle between two vectors, similarly to Sec. II-A. However, the invariance to the scaling in the orientation is not proved yet. It is because the mapping between the time derivative of the orientation and the angular velocity is non-linear. For example, the relationship between the Euler angles time derivative $\dot{\boldsymbol{e}} = [\dot{\phi}\ \dot{\theta}\ \dot{\psi}]^T$ and the angular velocity $\boldsymbol{\omega}$ is $\boldsymbol{\omega} = \boldsymbol{T}(\phi, \theta, \psi)\dot{\boldsymbol{e}}$, where the matrix $\boldsymbol{T}$ depends on the set of Euler angles used (see [13] for further details).

Although the invariance of (2) is not proven for orientation scale, it hardly affects the gesture recognition performance since the orientations of the human links are very similar among different people. The scaling to the orientation is almost negligible, unlike the linear scaling which is crucial to cope with kinematic differences between different users.

Since the proposed representation is singular if $\|\dot{\boldsymbol{r}}(t)\| = 0$ or if $\|\boldsymbol{\omega}(t)\| = 0$, in our implementation we simply set the invariant to zero when it is close to singularity.

[1] In our representation we use angular velocity and acceleration vectors that belong to the 3D space.



(a) (b)

Fig. 2. (a) Human shape (blue) extracted from the depth map provided by the Kinect sensor. (b) Human skeleton model in which the circles and the squares represent the tracked human parts. Parts marked with the circles are effectively used in the Continuous Motion dataset of Sec. V-B.

III. Data Collection and Filtering

A. Whole body motion features extraction

As mentioned in Sec. I, we are interested in recognizing gestures from different people and/or poses, using the Microsoft Kinect sensor. By using the *OpenNI* library[2], we can track different people from a depth map. The human body is represented by a skeleton model, consisting of 20 parts. For each body part we get the position and the orientation, with respect to a fixed frame attached to the Kinect sensor, updated at 30 fps. Fig. 2 shows a depth map obtained from Kinect sensor, and the skeleton model adopted.

Linear velocity and acceleration are simply computed using numerical differentiation. The orientation is described using a *unit quaternion* $\mathcal{Q} = (\eta, \boldsymbol{\epsilon})$. The equation relating the angular velocity to the time derivative of the quaternion $\dot{\mathcal{Q}} = (\dot{\eta}, \dot{\boldsymbol{\epsilon}})$ is given by $\boldsymbol{\omega} = 2\boldsymbol{S}(\boldsymbol{\epsilon})\dot{\boldsymbol{\epsilon}} + 2\eta\dot{\boldsymbol{\epsilon}} - 2\dot{\eta}\boldsymbol{\epsilon}$, where $\boldsymbol{S}(\cdot)$ is a *skew-symmetric operator* [13]. Then the angular acceleration is computed using numerical differentiation. Starting from this data, two invariants in eq. (1) and (2) are computed for each time instant and used as input for a classification algorithm.

B. Anisotropic diffusion

The proposed representation needs to compute numerically the first and second order derivatives of positions and orientations. Since it is well known that derivation is highly sensitive to the noise, we filter the noisy Kinect data using the *Anisotropic Diffusion* [14], saving at the same time the peaks in the trajectory.

This method iteratively smooths the data with a Gaussian kernel, but adaptively changes the variance of the Gaussian using the gradient of the signal $\boldsymbol{s}$ at the current time i as follows:

$$s_i^{j+1} = s_i^j + \alpha(c_N^j \nabla_N s_i^j + c_S^j \nabla_S s_i^j),$$

where $0 \leq \alpha \leq \frac{1}{4}$ is the control parameter, j the iteration number, c_N and c_S the *conduction parameters*, $\nabla_N s_i^j = s_{i-1}^j - s_i^j$, and $\nabla_S s_i^j = s_{i+1}^j - s_i^j$. The conduction parameters are updated at each iteration as:

$$c_N^j = e^{-\left(\|\nabla_N s_i^j\|/k\right)^2} \qquad c_S^j = e^{-\left(\|\nabla_S s_i^j\|/k\right)^2},$$

where k is the *noise estimator*. The iterations number, α and k are chosen empirically.

The effectiveness of this technique is shown in Fig. 3, depicting the original Kinect data (blue dashed line) and the smoothed trajectory (red solid line) of the height of the right hand with respect to the ground during the *handover* gesture. Herein, the used parameters are $\alpha = 0.2$, $k = 40$ and 10 iterations of the algorithm.[3]

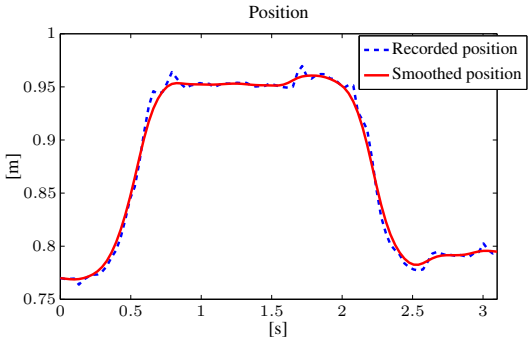


Fig. 3. Model (black solid line) of one of the invariant features obtained from five training features (blue dashed lines).

IV. Classification Algorithms

As stated in [9], an action can be recognised simply counting the number of *dynamic instants*, i.e. the local maxima of the curvature and their sign, however this simple solution is sensitive to noise in the data. In order to cope with uncertainties of different actors' motions, two classification algorithms based on *Dynamic Time Warping (DTW)* and *Hidden Markov Model (HMM)* are adopted.

A. DTW-based recognition

1) Training: Dynamic Time Warping (DTW) [15] is an algorithm often used to non-linearly align two sequences of different lengths. The DTW distance is a measure of the similarity of the two sequences. Since the standard DTW works only on one dimensional vectors, we modified the algorithm for the multi dimensional case for articulated human motion. Given two M dimensional motion sequences A and B, the distance matrix D is computed according to[4] $D(i,j) = \sum_{m=1}^{M}(A(m,i) - B(m,j))^2$, where i and j are the time index of each sequence. Using this distance matrix, the best alignment is found which results from the smallest cumulative distance between two sequences. We refer to this algorithm as M-DTW.

To create a model that abstracts a gesture from multiple demonstrations we follow the iterative procedure in

[2] http://openni.org

[3] The filtering step (with unoptimised C++ implementation) takes about $1ms$ per each frame in the on-line recognition scenario of Sec. V-B.

[4] Because of the different scale it is necessary to normalise each dimension to a zero mean and unit variance.

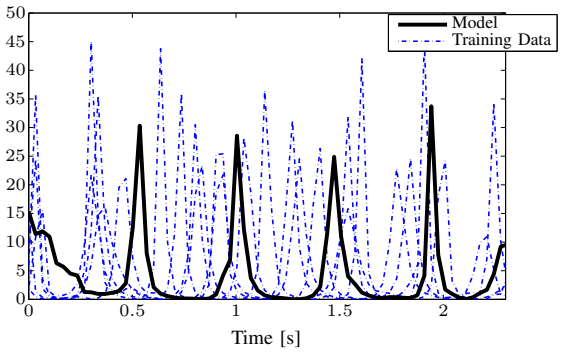


Fig. 4. Trained model (black solid line) of one of the invariant features obtained from five training demonstrations (blue dot-dashed lines).

[7] as follows. Consider N training demonstrations, i.e. N repetitions of the same gesture, each one consisting of M dimensional vectors of features. First, for each training demonstration i, the average M-DTW distance to all other training demonstrations, $d_{av,i}$, is computed. Then an exponential probability density function is fitted through these distances[5]:

$$w_i = P(d_{av,i}|\alpha) = \alpha exp^{-\alpha d_{av,i}}, \quad i = 1, \ldots, N$$

where $\alpha^{-1} = \frac{1}{N}\sum_{i=1}^{N} d_{av,i}$. The N dimensional vector $\boldsymbol{w} = [w_1, ..., w_N]^T$ is used to weight the contribution of each demonstration to the model of each training trial.

First, the two demonstrations with the lowest $d_{av,i}$ are aligned using M-DTW. As a result, the two demonstrations will have the same length: $\boldsymbol{s}_1^* = [s_{11}^*, ..., s_{1K}^*]^T$, and $\boldsymbol{s}_2^* = [s_{21}^*, ..., s_{2K}^*]^T$. Then, these vectors are fused together using a weighted average of the corresponding values in the two training trials:

$$s_k^* = \frac{w_1 s_{1k}^* + w_2 s_{2k}^*}{w_1 + w_2}, \quad k = 1, \ldots, K$$

where w_1 and w_2 are the corresponding elements in the vector $\boldsymbol{w}$. The procedure is repeated between $\boldsymbol{s}^* = [s_1^*, ..., s_K^*]^T$ and the other training trials until all the demonstrations are considered. In this way, the trained model becomes robust to the outliers. For example, a demonstration exhibiting high dissimilarity to the others is weighted differently and affects less to the final model. The result of this procedure is shown in Fig. 4, in which the blue dot-dashed lines represent five training demonstrations[6] and the black solid line the obtained model.

2) Gesture recognition: We define a model c_p for each different motion, using the procedure described in Sec. IV-A.1. Then, for each model p and each feature j, the DTW distances $d_{av,jp}^*$ between the new gesture and each of the gesture models are calculated. The log-likelihood $P(d_{av,jp}^*, j = 1, \ldots, M|c_p)$, where M is the number of features, is computed for each motion model c_p using an exponential distribution:

$$P(d_{av,jp}^*, j = 1, \ldots, M|c_p) = \sum_{j=1}^{M} \theta_{jp} log\left(P(d_{av,jp}^*|c_p)\right),$$

where θ_{jp} is used to select the features really involved in the gesture. The motion is then assigned to the model with the biggest probability if the difference between the first two bigger log-likelihood is over a certain empirically chosen threshold.

The weights θ_{jp} are set automatically from the gesture models. If the maximum of the *j-th* feature vector, in the *p-th* model, is bigger than a threshold, then $\theta_{jp} = 1$, otherwise $\theta_{jp} = 0.1$. This threshold is empirically set to 10.

B. HMM-based recognition

1) Hidden Markov Models: HMMs are widely used in the analysis of temporal sequence in several areas such as speech recognition [16], and gesture recognition [1], [2]. An HMM represents a Markov process which cannot be directly observed. In this paper Left-To-Right continuous HMMs (CHMMs) are adopted. The CHMM is described by the set of parameters $\boldsymbol{\lambda}$ containing:

- The set of the L hidden states $\boldsymbol{S} = \{s_i\}, 1 \leq i \leq L$. Hereinafter q_t denotes the state at time t.
- The set of M observable output symbols $\boldsymbol{O} = \{\boldsymbol{o}_t\}, 1 \leq t \leq T$, where $\boldsymbol{o}_t$ is the observation at time t.
- The initial state probability vector. $\boldsymbol{\pi} = \{\pi_i\}, 1 \leq i \leq L$, where $\pi_i = P(q_1 = s_i)$.
- The state transition probability matrix $\boldsymbol{A} = \{a_{ij}\}, 1 \leq i, j \leq L$, where a_{ij} is $P(q_{t+1} = s_j|q_t = s_i)$.
- The observation symbol probability distribution $\boldsymbol{B} = \{b_i(\boldsymbol{o}_t)\}, 1 \leq i \leq L$, where $b_i(\boldsymbol{o}_t)$ is $P(\boldsymbol{o}_t|q_t = s_i)$.

2) Training and recognition: In order to use HMMs to recognise motion we define an HMM for each different motion. The model parameters $\boldsymbol{\lambda}$ are estimated using the Baum-Welch algorithm [16], taking as input N sets of invariant features extracted from N repetitions of the same gesture. Once the parameters of HMMs are trained, gesture recognition for an observation sequence $\boldsymbol{O}_u = \{\boldsymbol{o}_{u1}, \ldots, \boldsymbol{o}_{ut}\}$ can be performed. For each HMM, the probability $P(\boldsymbol{O}_u|\boldsymbol{\lambda})$ is computed by using the Forward-Backward procedure [16]. The motion is then assigned to the model with the biggest probability if the difference between the first two bigger likelihood is over a certain empirically chosen threshold.

To represent $b_i(\boldsymbol{o}_t)$ we used a modified version of the mixture of G Gaussian distributions:

$$b_i(\boldsymbol{o}_t) = \sum_{g=1}^{G} c_{ig} \left[\frac{exp\left(-\frac{1}{2}(\boldsymbol{o}_t - \boldsymbol{\mu}_{ig})^T \boldsymbol{\Theta} \boldsymbol{\Sigma}_{ig}^{-1} (\boldsymbol{o}_t - \boldsymbol{\mu}_{ig})\right)}{(2\pi)^{G/2}|\boldsymbol{\Sigma}|^{1/2}}\right]$$

where c_{ig} is the mixture coefficient, $\boldsymbol{\mu}_{ig}$ and $\boldsymbol{\Sigma}_{jg}$ are respectively the mean vector and the covariance matrix for the g^{th} Gaussian component in state s_i. The diagonal matrix $\boldsymbol{\Theta} = diag(\theta_1, \theta_2, \ldots, \theta_M)$ weights the contribution of each feature. Its entries are chosen as described in Sec. IV-A.2.

In the experiments, we used 12 states and 1 Gaussian for each state. The increase of the number of states, or the number of Gaussian, does not affect the classification.

[5]Notice that it is possible to choose other distributions to fit through $d_{av,i}$.

[6]The invariants of the right hand positions during the *come* gesture.

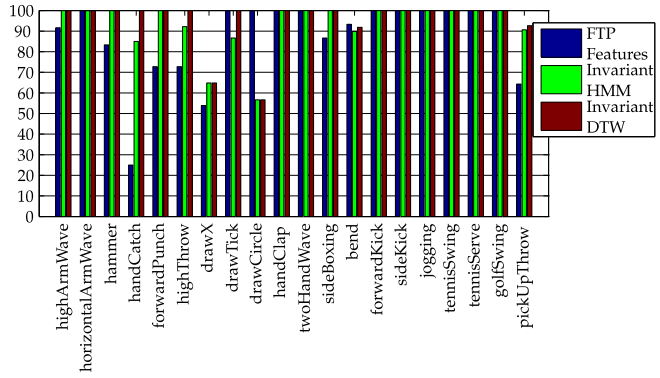


Fig. 5. The comparison between the recognition rates for MSR-Action3D dataset.

V. Experimental Results

The proposed approach is tested on two different datasets: the MSR-Action3D dataset[7] [12] and a *Continuous Motion (CM)* dataset. Both datasets are challenging: the first contains many similar actions, the second is a continuous stream of data in which the starting and the ending points of each gesture are unknown. The classification algorithms take as input the proposed invariants of human motion in Sec. II. We call them respectively *Invariant DTW* and *Invariant HMM*.

A. MSR-Action3D dataset

MSR-Action3D dataset contains twenty actions captured by a depth camera: *highArmWave*, *horizontalArmWave*, *hammer*, *handCatch*, *forwardPunch*, *highThrow*, *drawX*, *drawTick*, *drawCircle*, *handClap*, *twoHandWave*, *sideBoxing*, *bend*, *forwardKick*, *sideKick*, *jogging*, *tennisSwing*, *tennisServe*, *golfSwing*, *pickUp&Throw*. Each action is performed by ten subjects three times. The frame rate is 15 fps. The skeleton model is that in Fig. 2(b).

In order to compare our results with the *FTP features* approach [12], we consider only the positions of the twenty skeleton parts. Furthermore, we test our algorithm on the cross-subject test setting [12], using half of the samples as a training set, and the rest as a test set.

As shown in Fig. 5, the proposed invariant approach outperforms *FTP features* based method. In fact, the average recognition rate with *FTP features* is 88.2%, with *Invariant HMM* is 93.2% and with *Invariant DTW* is 95.3%.

B. Continuous Motion dataset

The CM dataset consists of a continuous sequence of six actions performed by four different users: *come*, *handover left*, *handover right*, *pick up*, *stop* and *rest*. Each gesture is performed twice by each person. Altogether, the dataset consists of 48 actions. Every time the user arbitrarily changes his pose with respect to the Kinect sensor. All the users perform *come*, *handover*, *stop* with the right hand, and *handover* with the left hand. Like in an on-line recognition scenario, the gesture segmentation is unknown. This dataset was created during the AUTOMATICA fair in 2012 [8].

Data are collected at 30 fps using the *OpenNI 1.3.2.1* library. The skeleton model adopted is shown in Fig. 2(b). The tracking algorithm, implemented in *OpenNI 1.3.2.1*, frequently fails in tracking the lower part of the human body. Moreover, the orientation estimation of both hands was not provided by the tracking algorithm. Since the purpose of this paper is not to realise a robust tracking technique, we simply ignore the contribution of these parts. The complete set of invariant features, consisting of 12 elements, is shown in Tab. I, while the skeleton parts used to create this set are represented as green circles in Fig. 2(b). We also decided to ignore the contributions of the *Head* and *Neck* motion, since they do not affect the performed gestures.

TABLE I
Set of invariant features used in the CM dataset

Body Part	Inv. Position (γ)	Inv. Orientation (ξ)
Hand (Left and Right)	yes	no
Elbow (Left and Right)	yes	yes
Shoulder (Left and Right)	yes	yes
Torso	yes	yes

In this experiment, training procedure was done offline. A user who is not included in the CM dataset performed five demonstrations for each gesture from arbitrary poses. This data was used for training models for DTW and HMM. With the trained models, we tested online recognition with the CM dataset. As mentioned above, in this case the data segmentation is unknown. This becomes challenging for recognition because the starting or ending parts of some gestures can be similar, for example *come*, *stop*, and *handover right* gestures. During the online recognition, we analyse data using a sliding window of length V. The recognition procedure starts after the arrival of the first V frames of poses. When a new data arrives every time frame, the window slides one frame and the recognition procedure is performed with the observation within the sliding window. If the same gesture is found F times, not necessarily consecutively, then the gesture is classified.

The results, obtained choosing $L = 50$ and $F = 15$, are shown in Fig. 6. As expected, errors occur among *come*, *stop*, and *handover right* gestures. The average recognition rate using the *Invariant DTW* algorithm is 91.67% and the minimum rate is 75% (*stop* gesture). Using *Invariant HMM* the average recognition rate is 87.5% and the minimum is 62.5% (*handover right* gesture). It is clear that the unknown segmentation significantly reduces the recognition rate. The adoption of a sophisticated segmentation technique certainly will improve the performance.

C. Discussion

Experiments in Sec. V-A and Sec. V-B show the effectiveness of our approach both on segmented and continuous

[7] http://research.microsoft.com/en-us/um/people/zliu/actionrecorsrc/default.htm

[8] http://www.automatica-munich.com
The proposed algorithm was tested further with more than 4 subjects during the whole fair period and showed reliable recognition rates.

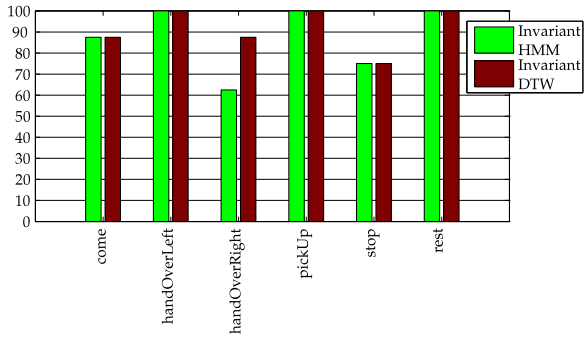


Fig. 6. The comparison between the recognition rates for the CM dataset.

motions, stating the importance of having an invariant representation of motion. In support of this, we tested another motion representation, which is the link positions in the torso frame. This representation is invariant to torso rotation and translation but not invariant to scaling factors. When using this representation, the average recognition rates drop to 65% for DTW and to 63% for HMM with the CM dataset.

The comparison with the *FTP features* shows that our method achieved good results with very similar gestures such as *bend* and *pickUp&Throw*. This is probably due to the fact that *FTP features* only use position information together with a technique to select involved features. On the other hand, our approach combines velocity and acceleration informations with a features selection technique. This combination makes gestures more distinctive in many cases. The worst result with our technique is the 57% of correct recognition on the *drawCircle* gesture. This is due to the way in which the *drawX* and the *drawCircle* are performed. The right hand trajectory in the *drawX* is in practise drawing a letter '8'. The right hand trajectory in the *drawCircle* consists of: a straight line, a circle, and another straight line to come back in the initial position. Since the gestures are performed continuously, the velocity profiles are similar, and misclassification occur.

The *Invariant DTW* algorithm usually outperforms the HMM approach. We believe that it is caused mainly from two aspects. First, the technique used to construct the DTW models weight differently trials that have high dissimilarities. So the resulting models are less sensitive to outliers in the training set. Second, as shown in Fig. 4, the invariant representation results in a very spiky signal. Some of the states in the HMM have a big variance in order to model this spikes. As a result, when two gestures are very similar, some states in the HMMs are partially overlapped, increasing the number of misclassified or unclassified gestures.[9]

VI. Conclusion

In this paper we present a new invariant representation in order to recognise motions performed by different people from different point of view. The proposed 3D human motion representation is invariant to rotations, translations and linear scaling factors. This representation is combined with two classification algorithms: modified HMM- and DTW- based classification algorithms. The proposed approach is tested with segmented and unsegmented datasets, and compared with another invariant representation and with non-invariant representation. Experimental results show the effectiveness of our approach. As future research we will focus on understanding human complex behaviours by integrating gesture recognition with other sensory information, which come from speech recognition and environment reconstruction.

[9]For example, *stop* and *handOverRight* in the CM dataset are very similar, especially considering that the used *OpenNI* library did not provide the hand orientation.

Acknowledgements

This work has been partially supported by the DFG excellence initiative research cluster "Cognition for Technical System CoTeSys" and the European Community within the FP7 ICT-287513 SAPHARI project.

References

[1] D. Aarno and D. Kragic, "Motion intention recognition in robot assisted applications," *Robotics and Autonomous Systems*, pp. 692–705, 2008.

[2] D. Lee, C. Ott, and Y. Nakamura, "Mimetic communication model with compliant physical contact in human-humanoid interaction," *Int. Journal of Robotics Research*, vol. 29, no. 13, pp. 1684–1704, 2010.

[3] Y. Piao, K. Hayakawa, and J. Sato, "Space-time invariants and video motion extraction from arbitrary viewpoints," in *Proceedings of the IEEE International Conference on Pattern Recognition*, 2002, pp. 56–59.

[4] ——, "Space-time invariants for recognizing 3d motions from arbitrary viewpoints under perspective projection," in *Proceedings of the IEEE International Conference on Image and Graphics*, 2004, pp. 200–203.

[5] A. Zisserman and S. Maybank, "A case against epipolar geometry," in *Applications of Invariance in Computer Vision*. Springer Berlin / Heidelberg, 1994, pp. 69–88.

[6] I. Weiss, "Geometric invariants and object recognition," *International Journal of Computer Vision*, pp. 207–231, 1993.

[7] J. De Schutter, E. Di Lello, J. De Schutter, R. Matthysen, T. Benoit, and T. De Laet, "Recognition of 6 dof rigid body motion trajectories using a coordinate-free representation," in *IEEE International Conference on Robotics and Automation*, 2011, pp. 2071–2078.

[8] P. Yan, S. M. Khan, and M. Shah, "Learning 4d action feature models for arbitrary view action recognition," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–7.

[9] C. Rao, A. Yilmaz, and M. Shah, "View-invariant representation and recognition of actions," *International Journal of Computer Vision*, pp. 203–226, 2002.

[10] C. Rao, M. Shah, and T. S. Mahmood, "Action recognition based on view invariant spatio-temporal analysis," in *ACM Multimedia*, 2003.

[11] R. J. Jagacinski, W. W. Johnson, and R. A. Miller, "Quantifying the cognitive trajectories of extrapolated movements," *Journal of Experimental Psychology: Human Perception and Performance*, pp. 43–57, 1983.

[12] J. Wang, Z. Liu, Y. Wu, and J. Yuan, "Mining actionlet ensemble for action recognition with depth cameras," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 1290–1297.

[13] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics - Modelling, Planning and Control*. Springer, 2009.

[14] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 629–639, 1990.

[15] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, pp. 43–49, 1978.

[16] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," in *Proceedings of the IEEE*, 1989, pp. 257–286.