Heterogeneous hand gesture recognition using 3D dynamic skeletal data

Quentin de Smedt, Hazem Wannous, Jean-Philippe Vandeborre

Heterogeneous hand gesture recognition using 3D dynamic skeletal data

Quentin de Smedt, Hazem Wannous, Jean-Philippe Vandeborre

Heterogeneous hand gesture recognition using 3D dynamic skeletal data

Quentin De Smedt[a,*], Hazem Wannous[b], Jean-Philippe Vandeborre[a]

[a] *IMT Lille Douai, Univ. Lille, CNRS, UMR 9189 - CRIStAL - Centre de Recherche en Informatique Signal et Automatique de Lille, F-59000 Lille, France.*
[b] *Univ. Lille, CNRS, Centrale Lille, IMT Lille Douai, UMR 9189 - CRIStAL - Centre de Recherche en Informatique Signal et Automatique de Lille, F-59000 Lille, France.*

ABSTRACT

Hand gestures are the most natural and intuitive non-verbal communication medium while interacting with a computer, and related research efforts have recently boosted interest. Additionally, the identifiable features of the hand pose provided by current commercial inexpensive depth cameras can be exploited in various gesture recognition based systems, especially for Human-Computer Interaction. In this paper, we focus our attention on 3D dynamic gesture recognition systems using the hand pose information. Specifically, we use the natural structure of the hand topology – called later *hand skeletal* data – to extract effective hand kinematic descriptors from the gesture sequence. Descriptors are then encoded in a statistical and temporal representation using respectively a Fisher kernel and a multi-level temporal pyramid. A linear SVM classifier can be applied directly on the feature vector computed over the whole presegmented gesture to perform the recognition. Furthermore, for early recognition from continuous stream, we introduced a prior gesture detection phase achieved using a binary classifier before the final gesture recognition. The proposed approach is evaluated on three hand gesture datasets containing respectively 10, 14 and 25 gestures with specific challenging tasks. Also, we conduct an experiment to assess the influence of depth-based hand pose estimation on our approach. Experimental results demonstrate the potential of the proposed solution in terms of hand gesture recognition and also for a low-latency gesture recognition. Comparative results with state-of-the-art methods are reported.

1. Introduction

Among human body parts, the hand is an effective and intuitive interaction tool in most Human-Computer Interaction (HCI) applications. Consequently, hand gesture recognition is becoming a central key for different types of applications such as virtual game control, sign language recognition, HCI, robot control, etc.

Using the hand gesture as a HCI modality introduces intuitive and easy-to-use interfaces for a wide range of applications in virtual and augmented reality systems, as well as offering support for the hearing-impaired and providing solutions for all environments using touchless interfaces. However, the hand is an object with a complex topology and has endless possibilities to perform the same gesture. For example, Feix *et al.* [8] summarize the grasping taxonomies and found 17 different hand shapes to perform a grasp. The *grasp* is a hand gesture where we need precise information about the hand shape if we want to recognize its type. Other gestures, such as *swipes*, which are more defined by the hand motion than its shape, are already commonly used in tactile HCI. This heterogeneity between useful gestures have to be taken into account in a hand gesture recognition algorithm.

e-mail: hazem.wannous@univ-lille.fr (Hazem Wannous)

To date, the most reliable tools used to capture 3D hand gestures are motion capture devices, which have sensors attached to a glove delivering real-time measurements of the hand. However, they present several drawbacks in terms of the naturalness of hand gesture and cost, in addition to their complex calibration setup process. Recently, effective and inexpensive depth sensors, like the Microsoft Kinect, have been increasingly used in the domain of computer vision. By adding a third dimension into the game, depth images offer new opportunities to many research fields, one of which is the hand gesture recognition area. In recent years, many researchers [52, 41, 14, 44, 33, 2, 42, 31, 13, 50, 19, 6, 24, 17, 18] studied hand gesture recognition challenges using color and/or depth images.

In the field of action recognition, Shotton *et al.* [35] proposed a real-time method to accurately predict the 3D positions of 20 body joints, together called *body skeleton*, from depth images. Hence, several descriptors in the literature proved how the position, the motion, and the orientation of joints could be excellent descriptors for human actions. Following this statement, hand skeletal data could handle precise information of the hand shape that HCI need in order to use the hand as a manipulation tool.

Very recently, new devices, such as the Intel RealSense or the Leap Motion Controller (LMC), provide precise **skeletal data** of the hand in the form of a full 3D skeleton corresponding to 22 joints in $\mathbb{R}^3$ labeled as shown in Fig. 1. Potter *et al.* [29] presented an early exploration of the suitability of using such data from a LMC in order to recognize and classify precise hand gestures of Australian Sign Language. However, hand pose estimation from depth images remains a prominent field of research. Many issues still have to be solved: properly recognizing the skeleton when the hand is either closed or perpendicular to the camera, without an accurate initialization, or when the user performs a quick gesture. The hand contains more joints than there are in the rest of the human body model of Shotton *et al.* [35] and is a smaller object. The hand has also a more complex structure. If an arm, a head or a leg can have different shapes, the hand is composed of a palm and five similar fingers making its pose estimation more difficult.

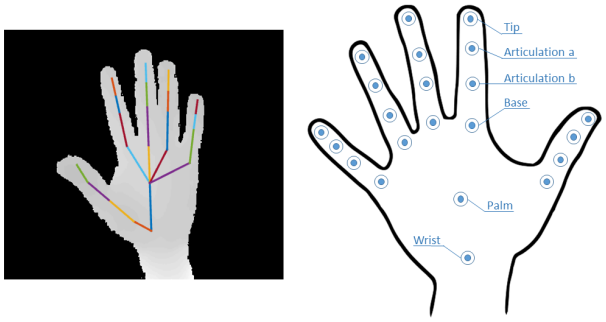


Fig. 1: Depth and hand skeletal data returned by the Intel RealSense camera. The 22 joints include: one for the center of the palm, one for the wrist and four joints for each finger representing the tip, the two articulations and the base. All joints are represented in $\mathbb{R}^3$. The Leap Motion Controller also provides a very similar hand skeleton.

As a new field of study, there are few 3D hand gesture datasets providing skeletal data [15, 3]. We created the latest one, called *Dynamic Hand Gesture*, in a previous work to study the use of hand skeletal data to perform gesture recognition. Volunteers performed each hand gesture using either only one finger or the whole hand. In this paper, we investigate the use of a hand skeleton model in a novel dynamic hand gesture recognition solution. We propose to capture the motion and the hand shape variations based on the skeletal joints from gesture frames. Our overall approach is sketched in Fig. 2.

We emphasize our analysis of the use of skeletal data on hand gesture recognition to meet four main challenges: (1) Classifying hand gestures when the dataset contains gestures defined by the hand shape and/or by the hand motion through the sequence; (2) Performing the same gesture in various ways due to the endless possibilities of using a different number of fingers. This happens mainly because of the complex topology of the hand. (3) Investigating the impact of the hand pose estimator on the recognition process; (4) Evaluating the recognition in terms of latency, enabling early recognition of gestures.

The rest of this paper is structured as follows. Related work on 3D hand gestures in terms of datasets and recognition approaches are reviewed in Section 2. Our recognition approach is described in Section 3. In Section 4, the strengths of our approach in terms of accuracy and latency on three datasets are

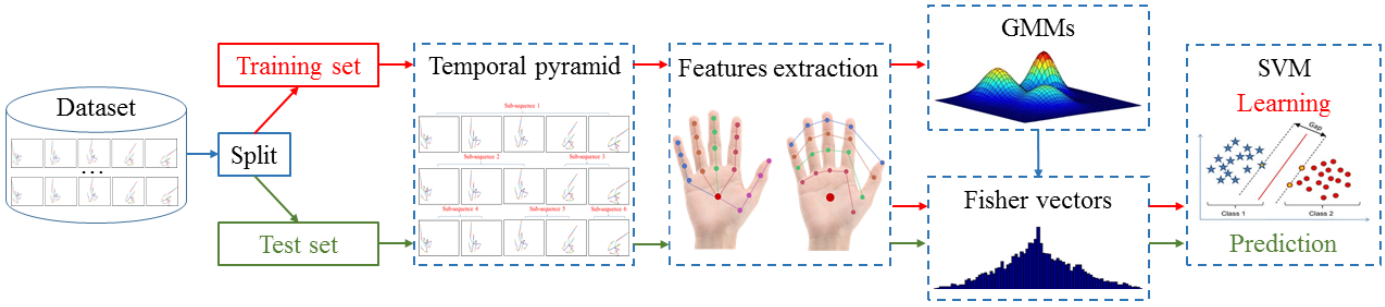


Fig. 2: Overview of the approach. The illustrated pipeline is composed of four modules: (1) features extraction, (2) temporal modeling using a pyramid, (3) statistical representation using a Fisher vector, (4) SVM classifier.

demonstrated before concluding in Section 5.

2. Related Work

Hand gesture recognition has been an active research field for the past 20 years, where a variety of approaches have been proposed. Over the past six years, advances in commercial 3D depth sensors have substantially promoted the research of 3D hand gesture detection and recognition. The approaches reviewed below focus on 3D hand gesture recognition, and can be gathered into three main categories: **static** and **dynamic** hand gesture recognition using depth images and/or hand **skeletal data**.

In most of the **static** approaches, 3D depth information is used to recognize hand silhouettes or simply hand areas in order to extract features from a segmented hand region. Features are usually based on global information as proposed by Kuznetsova *et al.* [14], where an ensemble of shape function is computed on the hand point cloud. Other local descriptors are expressed as the distribution of points in the divided hand region into cells [44]. Instead of using the distribution of points in the region of the hand, Ren *et al.* [33] represented the hand shape as a time-series curve and used a distance metric called Finger-Earth Mover Distance to distinguish one hand gesture from a collected dataset of 10 different gestures. The time-series curve representation is also used by Cheng *et al.* [2], to generate a set of fingerlet representing the hand gesture. Wang et al. [42] proposed a superpixel earth mover distance metric of depth and color images acquired with a Microsoft Kinect, which effectively retains the overall shapes of hand gestures. They tested their method on a dataset which contains 10 different gestures performed by five subjects. Additionally, sign language recognition has been widely investigated. Pugeault and Bowden [31] proposed a method using Gabor filter for hand shape representation and a Random Forest for gesture classification. They applied their method to a collected *ASL Finger Spelling* dataset, containing 48000 samples of RGB-D images labeled according to 24 static gestures of the American Sign Language.

Unlike the static approaches which work on hand description based on a single image, **dynamic** methods work on the temporal aspect of hand motion, by considering the gesture as a sequence of hand shapes. Kurakin *et al.* [13] presented the MSRGesture3D dataset containing 12 dynamic gesture from the American Sign Language. They recorded 360 sequences of hand depth images from a Microsoft Kinect. Their recognition algorithm is based on a hand depth cell occupancy and a silhouette descriptor. They used an action graph to represent the dynamic aspect of the gestures. Recently, using a histogram of 3D facets to encode 3D hand shape information from depth maps, Zhang *et al.* [50] outperformed latest results obtained on the MSRGesture3D dataset using a dynamic programming-based temporal segmentation. One of the tracks of the *Chalearn 2014* [6] consists of using a multimodal database of 4000 gestures drawn from a vocabulary of 20 dynamic Italian Sign Language gesture categories. They provided sequences of depth images of the whole human body and body skeletons. From

this dataset, Monnier *et al.* [19] employed both body skeleton and Histogram of Oriented Gradients (HOG) features computed on the depth map cropped around the hand to perform a gesture classification using a boosted cascade classifier. Recently, the use of deep learning has changed the paradigm of many research fields in computer vision. Recognition algorithms using specific neural network – like Convolutional Neural Network (CNN) – obtained previously unattainable performance. On the *Chalearn 2014* [6] dataset, Neverova *et al.* [20] used stacked CNNs applied to raw intensity and depth sequences around the hand and a multilayer perceptron on body skeletons.

In order to study hand gesture recognition in a real-time scenario for automotive interfaces, Ohn-Bar and Trivedi [24] made a publicly available dataset of 19 gestures performed in a car captured with the Microsoft Kinect. The initial resolution obtained by such a sensor is 640×480 and the final region of interest is 115×250. Moreover, at some distance from the camera, with the illumination varying in the car, the resulting depth is very noisy, making the challenge of gesture recognition tougher. They compared the accuracy of gesture recognition using several known depth features (HOG, HOG3D, HOG^2). Using stacked 3D CNNs combining multiple spatial scales, Molchanov *et al.* [17] improved earlier results. Very recently, Molchanov *et al.* [18] introduced a new challenging multimodal dynamic hand gesture dataset captured with depth, color and stereo-IR sensors. They acquired 1532 sequences of 25 gesture types intended for Human-Computer Interfaces. They trained a recurrent 3D CNN that performs simultaneous detection and classification of dynamic hand gestures. They used a connectionist temporal cost function in order to predict class labels from in-progress gestures in unsegmented input streams.

In the field of action recognition, using the body skeletal data has shown promising results. Vemulapalli et al. [40] utilize rotations and translations to represent the 3D geometric relationships of body skeleton parts in the Lie group, and then employ Dynamic Time Warping and a Fourier Temporal Pyramid to model the temporal (FTP) dynamics. The FTP is also used by Wang et al. [45] with local occupancy pattern features extracted from depth maps. They applied a data mining framework to discover the most discriminative combinations of body joints. Recently, Devanne et al. [4] represented the spatio-temporal human motions by a full human skeleton trajectory. This motion trajectory are extracted from 3D joints and expressed on a Riemannian manifold.

In the field of hand gesture recognition, the use of hand **skeletal data** is at its beginning. A pioneering work from Keskin et al. [12] is the Random Decision Forests (RDF) based hand skeleton tracking. It performs per-pixel classification, assigns each pixel to a hand part and a mean shift is used to estimate the centers of hand parts, together called a hand skeleton. In addition, this algorithm is enhanced by adding an upstream RDF hand shape classifier which serves as an intermediate layer to bridge pixels to specific RDF hand pose estimators.

Later, Dong *et al.* [5] outperformed the results obtained on the static hand gesture *ASL Finger Spelling* dataset. They went in depth into the hand representation. They proposed a hierarchical mode-seeking method to locate positions of hand joints under kinematic constraints, segmenting the hand region into 11 natural parts (one for the palm and two for each finger). A Random Forest classifier is then built to recognize ASL signs using a feature vector of joint angles.

Thanks to recent devices, such as the Leap Motion Controller (LMC) or the Intel RealSense, we can now employ skeletal information without regard for the hand pose estimation phase in order to create fast and accurate hand gesture recognition algorithms. Potter *et al.* [29] proved the potential to recognize hand gesture with skeletal data obtained by the LMC. Following this statement, Marin *et al.* [16] mixed depth and skeletal data descriptors, respectively using a Microsoft Kinect and a LMC, in order to recognize ASL. They computed distances and angles between the hand joints and also curvatures on the hand depth, coupled with a multi-class SVM for classification. Xu *et al.* [46] captured the information of hand motion trajectory using a LMC to recognize ten simple dynamic hand gestures. In 2016, Lu *et al.* [15] built a dataset called *Handicraft-Gesture* using the LMC. This dataset is made of 10 gestures which originate

from pottery skills. There were 10 volunteers helping to build the dataset and each one performed every gesture three times resulting in 300 sequences of hand skeleton gestures. Using angles and distance-based features coupled with a Hidden Conditional Neural Field classifier, they obtained 95% recognition accuracy. We introduced in a previous work [3] a challenging skeleton based dynamic hand gesture dataset containing 14 gestures, made by 20 volunteers performing the same gesture with two different sets of fingers. It results in 2800 sequences. In this paper, we go further into the hand gesture description using hand skeletal data and study its impact on the different types of gesture.

3. Hand Gesture Recognition approach

Using 3D hand skeletal data depicted in Fig. 1, a dynamic gesture can be seen as a time series of hand skeletons. It describes the motion and the hand shapes along the gesture. For each frame t of the sequence, the position in the camera space of N_j joint which are represented by three coordinates, i.e. $j_i(t) = [\ x_i(t)\ y_i(t)\ z_i(t)\ ]$. N_j is the number of joints which compose the hand skeleton. The skeleton at frame t is then represented by the $3N_j$ dimension row vector:

$$s(t) = [\ x_1(t)\ y_1(t)\ z_1(t)\ ...\ x_{N_j}(t)\ y_{N_j}(t)\ z_{N_j}(t)\ ] \tag{1}$$

With N_f representing the number of frames in the sequence, the final representation of the sequence is a matrix of size $N_f \times 3N_j$ where each line t is the row vector $s(t)$:

$$\mathcal{M} = \begin{bmatrix} s(1) \\ \vdots \\ (N_f) \end{bmatrix} \tag{2}$$

This new type of data handles a lot of information on the motion and the shape of the hand along the sequence. In order to fully represent the gesture, we propose to mainly capture the hand shape variations based on skeleton joints, but also the direction of the movement and the rotation of the hand with three distinct features.

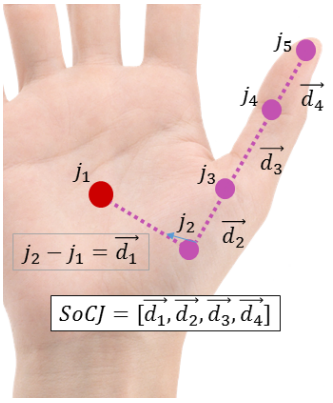


Fig. 3: An example of the SoCJ descriptor constructed around the thumb tuple. Let be $T = (j_1, j_2, j_3, j_4, j_5)$ where $j_i \in \mathbb{R}^3$. We compute the displacements from points to their respective right neighbor resulting in the SoCJ vector $[\vec{d}_1, \vec{d}_2, \vec{d}_3, \vec{d}_4]$.

3.1. Feature Extraction

3.1.1. Motion features

Some gestures are defined almost only by the way the hand moves in space (e.g. *swipes*). To take this characteristic into account, we compute a direction vector in $\mathbb{R}^3$ for each frame t of our sequence using the position of the palm joint noted j_{palm}:

$$\vec{d}_{dir}(t) = \frac{j_{palm}(t) - j_{palm}(t-c)}{\left\| j_{palm}(t) - j_{palm}(t-c) \right\|} \tag{3}$$

where c is a constant value chosen experimentally. We normalize the direction vector by dividing it by its norm.

For a sequence of N_f frames, we have the set $\mathcal{S}_D$:

$$\mathcal{S}_D = \left\{ \vec{d}_{dir}(t) \right\}_{[\ 1 < t < N_f\ ]} \tag{4}$$

The rotation of the wrist during the gesture describes also how the hand is moving. For each frame t, we compute the vector from the wrist node to the palm node to get the rotational information in $\mathbb{R}^3$ of the hand:

$$\vec{d}_{rot}(t) = \frac{j_{palm}(t) - j_{wrist}(t)}{\left\| j_{palm}(t) - j_{wrist}(t) \right\|} \tag{5}$$

For a sequence of N_f frames, we have the set $\mathcal{S}_R$:

$$\mathcal{S}_R = \left\{ \vec{d}_{rot}(t) \right\}_{[\ 1 < t < N_f\ ]} \tag{6}$$

3.1.2. Hand shape feature

To represent the shapes of the hand during the sequence using skeleton data, we propose a descriptor based on sets of joints, denoted as *Shape of Connected Joints* (SoCJ).

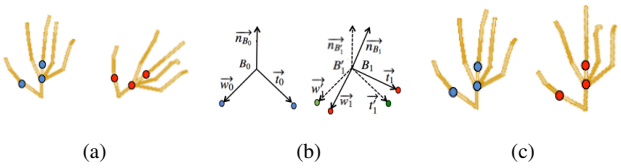
Hand skeletons returned from sensors consists of 3D coordinates of hand joints, represented in the camera coordinate system. Therefore, they vary with the rotation and translation of the hand with respect to the camera. To make our hand shape descriptor invariant to hand geometric transformations, we propose a normalization phase. Firstly, in order to take into account the differences of hand size between performers, we estimate the average size of each bone of the hand skeleton using all hands in the dataset. Secondly, carefully keeping the angles between bones, we change their size by their respective average size found previously.

Indeed, in order to be consistent with the translation and rotation transformations, we create a reference skeleton hand H_f corresponding to an open hand in front of the camera with its palm node at [0 0 0] as the *root joint*. Then, we define a new base with origin in the root joint, which includes the wrist node vector $\overrightarrow{w}$, the base of the thumb node vector $\overrightarrow{t}$, and their cross product $\overrightarrow{n_B} = \overrightarrow{w} \times \overrightarrow{t}$. This new base is then translated and rotated, so as to be aligned with a reference base B_0 computed from H_f. The calculation of the optimal rotation between the two bases B_1 of a current skeleton and B_0 of the reference skeleton H_f, is performed using *Singular Value Decomposition* (SVD). This process results in a new hand which keeps its shape but centered around [0 0 0] with the palm facing the camera. For each gesture sequence, we compute the translation and the rotation of the first hand skeleton with respect to the H_f and then apply the same transformations to all other hand skeletons of the sequence. This guarantees the invariance of the representation to the position and orientation of the hand in the scene. Fig. 4 shows an example of alignment of two different hand skeletons.

Let x represent the coordinates of a joint in $\mathbb{R}^3$ and $T = [x_1\ x_2\ x_3\ x_4\ x_5]$ a tuple of five ordered different joints from the hand skeleton s. To describe the shape of the joint connections, we compute the displacement from one point to its right-hand neighbor:

$$SoCJ(T) = [\ x_2 - x_1\ ...\ x_5 - x_4\ ] \tag{7}$$

This results in a descriptor in $\mathbb{R}^{12}$. Fig. 3 shows an example



(a) (b) (c)

Fig. 4: The calculation of the optimal rotation between the two hand skeletons using SVD: (a) Two skeletons with different orientations: the reference one on the left side and on the right side, the first skeleton of a sequence; (b) Bases B_1 and B_2 are built from the two corresponding wrists, the bases of the thumb joint vectors and their cross products. The base B'_2 corresponds to B_2 aligned with respect to B_1; (c) The resulting skeleton (right) is now aligned with respect to the first one (left). The transformations computed between these two bases are applied to all the skeletons of the sequence.

of a particular SoCJ using the palm joint and the thumb's. We remind that the skeleton of the Intel RealSense camera is composed of 22 joints. Theoretically, we can compute $C(22, 5) = 26334$ different SoCJs for the hand skeleton s where C is the binomial coefficient function resulting in the set:

$$s_{socj} = \{\ SoCJ(i)\ \}_{[\ 1 < i < 26334\ ]} \tag{8}$$

For a sequence of N_f frames, we have the set $\mathcal{S}_{socj}$:

$$\mathcal{S}_{socj} = \left\{\ s_{socj}(t)\ \right\}_{[\ 1 < t < N_f\ ]} \tag{9}$$

3.2. Fisher Vector Representation

The *Fisher Vector* (FV) coding method was first introduced for large-scale image classification. Its superiority against the *Bag-Of-Word* (BOW) method has been analyzed in the image classification [34]. It also has been used over the past five years in action recognition [7, 27, 49, 43].

First, we train a K-component *Gaussian Mixture Model* (GMM). Then, can compute our FV which is given by the derivatives of gradient. We normalize the final vector with a $l2$ and power normalization to eliminate the sparseness of the FV and increase its discriminability. We refer the reader to Sanchez *et al.* [28] for more details.

It is interesting to notice that the final size of a FV is $2dK$ where d is the size of the feature data and K the number of clusters in the classification process. This observation is a drawback compared to BOW, which has a size of K, when applied

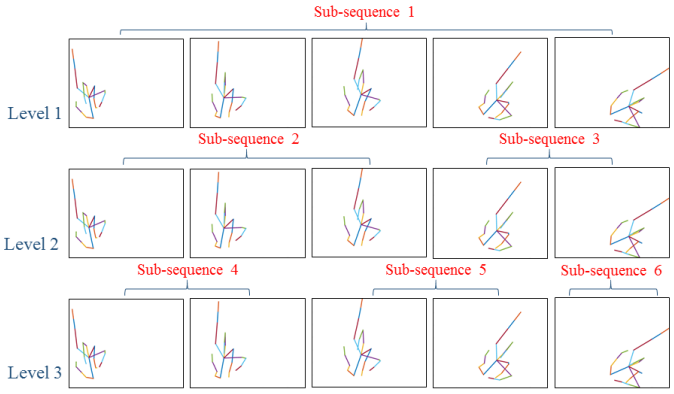


Fig. 5: An example of a Temporal Pyramid of size 3.

to a long descriptor. However, this effect can be ignored in our case where K is relatively small.

At the end of the feature extraction, we represent a sequence of hand skeletons by three sets of different features describing the direction of the hand ($\mathcal{S}_D$), its rotation ($\mathcal{S}_R$) and its shape ($\mathcal{S}_{soc_j}$) during the sequence.

3.3. Temporal Representation and Classification

The descriptors explained previously in section 3.1 describe the hand shape and the motion variation inside the sequence but they do not take into consideration the dynamic nature of a gesture. To add the temporal cue, we use a *Temporal Pyramid* (TP) representation already employed in action and hand gesture recognition approaches [7, 50].

The principle of the TP is to divide the sequence into n sub-sequences at each n^{th} level of the pyramid (shown in Fig. 5). We compute our three descriptors and their statistical representations for each sub-sequence and concatenate them. Adding more levels to the pyramid gives more temporal precision but increases substantially the size of the final descriptor and the computing time.

For gesture classification, we use a supervised learning classifier SVM with a linear kernel as it easily deals with our high-dimensional representation. We employ a *one-vs-rest* strategy resulting in G binary classifiers, where G is the number of different gestures in the experiment. We make use of the implementation contained in the LIBSVM library [1].

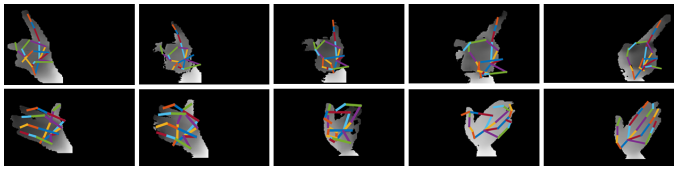


Fig. 6: *Swipe Right* gesture performed (top) with one finger and (bottom) with the whole hand from the DHG-14/28 dataset.

4. Experiments

In this section, we first evaluate our proposed approach on two datasets and compare it with four state-of-the-art methods using depth images and skeletal data. We then explore its capability to reduce the latency of the recognition process by evaluating the trade-off between accuracy and latency. We also study the impact of the hand pose estimation on a third dataset and finally discuss the promising potential of our approach and limitations.

4.1. Datasets

4.1.1. Dynamic Hand Gesture 14-28 dataset

In a preliminary work conducted recently, we presented the Dynamic Hand Gesture (DHG) 14/28[1] dataset [3]. It contains 14 dynamic hand gestures performed in two ways: using one finger or the whole hand (an example is given in Fig. 6). Each gesture is performed five times by 20 volunteers in two ways, resulting in 2800 sequences. Sequences are labeled following their gesture, the number of fingers used and the performer. The Intel RealSense short range depth camera is used to collect the dataset. Each frame contains a depth image and the coordinates of 22 positions of hand joints in the 3D camera space. The depth images and the hand skeletons were captured at 30 frames per second. Depth images have a 640×480 resolution. The length of the sample gesture ranges goes from 20 to 150 frames.

All gestures are listed in Table 1 according to this performing manner, fine or coarse. Fine gestures are usually performed by fingers, and coarse gestures are mainly performed by hand movements (e.g. swipe gestures). All have been chosen to be

[1] Available on: http://www-rech.telecom-lille.fr/DHGdataset

Table 1: List of the gestures included in the DHG-14/28 dataset.

Gesture	Label	Tag name
Grab	Fine	G
Expand	Fine	E
Pinch	Fine	P
Rotation CW	Fine	R-CW
Rotation CCW	Fine	R-CCW
Tap	Coarse	T
Swipe Right	Coarse	S-R
Swipe Left	Coarse	S-L
Swipe Up	Coarse	S-U
Swipe Down	Coarse	S-D
Swipe X	Coarse	S-X
Swipe V	Coarse	S-V
Swipe +	Coarse	S-+
Shake	Coarse	Sh

close to the state-of-the-art, like the VIVA challenges dataset [24]. Nevertheless, we removed the differentiation between normal and scroll swipe, as you can find it in our number-of-fingers approach, and with the pairs of gestures *Pinch/Expand* and *Open/Close*. In addition, we supplemented this base with the gesture *Grab* because of its usefulness in augmented reality applications, but also for its scientific challenges related to the potentially high variation among performers. We also added the gesture *Shake*, as it is interesting for recognition algorithms to be able to differentiate gestures composed from other gestures (the shake gesture is a repetition of opposed swipe gestures).

We focused our dataset on three main challenges: (1) Studying 3D dynamic hand gesture recognition using depth *and* full hand skeleton; (2) Evaluating the effectiveness of the recognition process following the heterogeneity of the hand shape depending on the set of fingers used. (3) Distinguishing between both fine-grained and coarse-grained gestures. Indeed, dividing the gesture sequences into two categories – coarse and fine gestures – contributes to increasing difficulties in the recognition challenge. Gesture categories are given in Table 1.

4.1.2. Handicraft-Gesture dataset

Handicraft-Gesture is a dataset built with a Leap Motion Controller (LMC) [15]. A LMC is a device providing accurate information about the hand skeleton which contains the same 22 joints described in Fig. 1. This dataset is made of 10 gestures which originate from pottery skills: *poke*, *pinch*, *pull*, *scrape*, *slap*, *press*, *cut*, *circle*, *key tap*, *mow*. The data are captured at a rate of 60 frames per second. There were 10 volunteers helping to build the dataset and each one performed every gesture three times. Therefore, the Handicraft-Gesture dataset contains 300 sequences of dynamic hand gestures.

4.1.3. NVIDIA Dynamic Hand Gesture dataset

Recently, Molchanov *et al.* [18] introduced a new challenging multimodal dynamic hand gesture dataset captured with depth, color and stereo-IR sensors in a car simulator. Using multiple sensors, they acquired a total of 1532 gestures of 25 hand gesture class annotated respectively 1 to 25 for: *moving the hand left*, *right*, *up*, or *down*; *moving two fingers left*, *right*, *up*, or *down*; *clicking with the index finger*; *calling someone (beckoning with the hand)*; *opening and shaking the hand*; *showing the index finger*, *two fingers* or *three fingers*; *pushing the hand up*, *down*, *out* or *in*; *rotating two fingers clockwise* or *counter-clockwise*; *pushing forward with two fingers*; *closing the hand twice*; and *showing thumb up* or *OK*. Similarly to the DHG dataset, this set contains coarse and fine gestures. A total of 20 subjects participated in data collection, performing gestures with their right hand. The SoftKinetic DS325 sensor is used to acquire frontal view color and depth videos.

4.2. Experimental settings

4.2.1. Descriptor encoding

We choose the number of levels L_{pyr} of our TP equal to 4 as it provides a satisfactory compromise between the temporal representation of the gestures and the final size of our descriptor. The final size of our descriptor computed from each gesture sequence is then $(\sum_{i=1}^{L_{pyr}} i) \times (size_{\Phi D} + size_{\Phi R} + size_{\Phi SoCJ})$. Note

that $size_\Phi = 2dK$, where K is the number of model trained in the GMM and d the size of the feature. d is the dimensions of descriptors, represented in $\mathbb{R}^3$, $\mathbb{R}^3$, $\mathbb{R}^{12}$ respectively for the direction, the rotation and the SoCJ features. For Fisher Vector encoding, we map our descriptors into a K-component GMM with K equal to 8, 8 and 256 gaussians respectively for the direction, the rotation and the SoCJ features. For all experiments conducted on the previous datasets, we use a *Leave-One-Subject-Out cross-validation* protocol.

4.2.2. *Intuitive versus automatic selection of SoCJ descriptors*

On a hand skeleton composed of 22 joints, we can compute 26334 different SoCJs. Using all of them is not mandatory as they provide redundant information and cost computing time. We propose to evaluate two ways to choose our feature set as a combination of the most relevant SoCJs. We first evaluate a SoCJ set chosen intuitively and, second, by using an automatic suboptimal deterministic feature selection algorithm called Sequential Forward Floating Search (SFFS). In this section, the evaluation criterion $\mathbb{J}$ of each feature set is the classification accuracy obtained using only fine gestures of the DHG dataset. We choose this subset of gestures as the SoCJs are meant to describe the hand shape (50% of the dataset is used for test while using the remainder as training observations). Firstly, to represent the hand shape, we intuitively divide the hand skeleton into nine tuples of five joints representing the hand's physical structure as presented in Fig. 7 and from which we will compute our SoCJ descriptor. This subset of nine tuples is chosen as it forms a grid on the hand skeleton and each joint appears at least once. We obtain on this subset a score $\mathbb{J}$ of 73.22%.

Secondly, we use the SFFS algorithm proposed by Pudil *et al.* [30] in order to automatically choose a relevant subset $\mathbb{X}$ following the criterion $\mathbb{J}(\mathbb{X})$ laid down above. Starting with an empty set of features $\mathbb{X}$, this algorithm works in three steps:

1. Inclusion: select the most significant feature with respect to $\mathbb{J}$ and include it in $\mathbb{X}$.
2. Conditional exclusion: find the least significant feature k in $\mathbb{X}$. If it is the feature just added, then keep it and return to step 1. Otherwise, exclude the feature k and continue to step 3.
3. Continuation of conditional exclusion: again find the least significant feature in $\mathbb{X}$. If its removal (a) leave $\mathbb{X}$ with at least 2 features, and (b) the value of $\mathbb{J}(\mathbb{X})$ is greater than the criterion value of the best feature subset of that size found so far, then remove it and repeat step 3. When these two conditions are no longer satisfied, return to step 1.

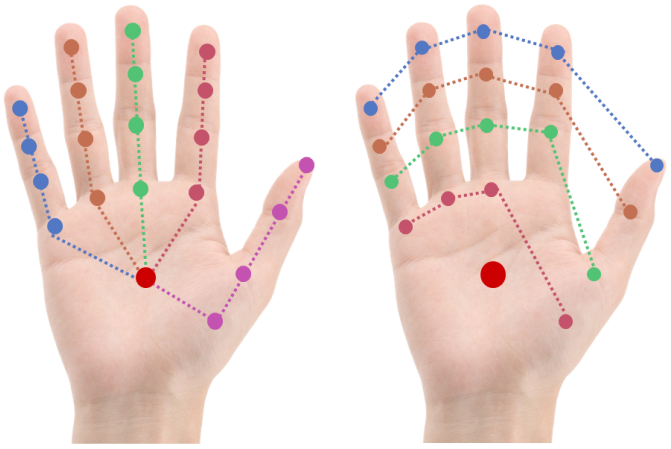


Fig. 7: The nine tuples chosen intuitively to construct the SoCJ descriptors. On the left side are the five constructed with the four joints of each finger plus the palm. On the right side, the one using the five tips, the five first articulations, the five second articulations and the five finger bases.

We conducted several experiments in order to choose the better combinations of SOCJs, using 4, 5, 6 and 7 joints. The better $\mathbb{J}$ score of 75.73% is obtained with a combination of 10 SoCJs using 5 joints for each, while adding more SoCJ seems irrelevant as the accuracy does not increase. This combination provides a good compromise between time complexity and accuracy. Results of the SFFS algorithm are shown in Fig. 8.

Fig. 9 shows the first three SoCJs selected by the SFFS algorithm. It is interesting to see that the first one is composed of joints which belong to the thumb and the index, thus providing the necessary information about the hand "clamp". The second one gathers one joint of each finger and the information of the general shape of the hand (i.e. "open" or "close"). If we use all the 26334 possible combinations (SoCJ is represented by 5 joint), the accuracy decreased to 74.30% as a misclassification due to the lack of precision and the redundancy. We note that

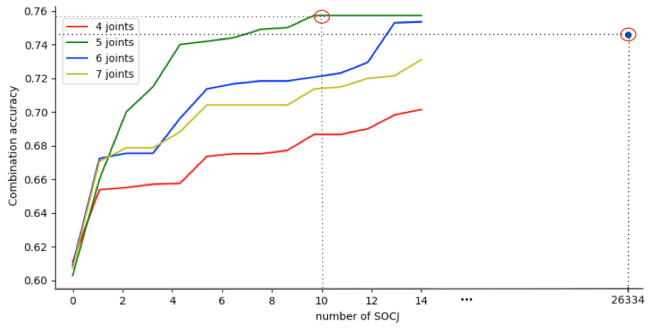


Fig. 8: SoCJ selection using SFFS algorithm on the fine gesture subset of the DHG dataset. The y-axis accuracy is obtained using the number of SOCJ on the x-axis. Best accuracy is obtained with 10 SoCJs, each one represented by 5 joints. Using more that 10 SOCJs is not relevant as the accuracy does not increase.

the computation of all the 26334 SoCJs for a sequence of 35 frames takes 6.24 seconds, and only 0.0022 seconds for the 10 SoCJs chosen by the SFFS. We use these SoCJs in the following experiments as this subset improves the score compared to the one chosen intuitively.

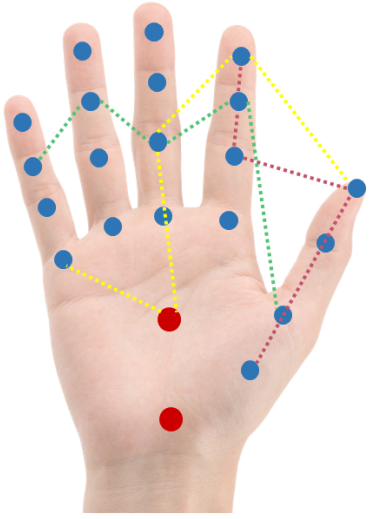


Fig. 9: The first three SoCJ chosen by the Sequential Forward Floating Search algorithm.

4.3. *Influence of hand pose estimation on gesture recognition*

The introduction of commodity depth sensors and the multitude of potential applications have stimulated new advances inside the hand pose estimation community. However, it is still challenging to achieve efficient and robust estimation performance because of large possible variations of hand poses, severe self-occlusions and self-similarities between fingers in the depth image. The current state-of-the-art methods mostly employ deep neural networks to estimate hand pose from a depth image [38, 22, 51, 9, 48]. The availability of a large-scale, accurately annotated dataset is a key factor for advancing this field of research. Consequently, numerous RGB-D datasets have been made publicly available last years. The different hand pose datasets differ in the annotation protocol used, the number of samples, the number of joints in the hand skeleton representation, the view point and the depth image resolution. Currently, the widely used datasets in the literature benchmarking purposes are IVCL [37], NYU [38] and MSRA15 [36]. The IVCL [37] and the MSRA15 [36] datasets are captured using the Intel Creative depth sensor (time-of-light), and composed respectively of 180K and 76.5K ground truth annotated depth images with the 3D joint locations of the hand. The NYU [38] comprises 72K frames of multi-view depth images captured using the Primesense Carmine camera (structured light).

4.3.1. *Estimator evaluation*

In order to measure the effect of pose estimation on gesture recognition, we performed several experiments on the two first datasets as their capture technology corresponds to the used hand gesture datasets in this work. First, we evaluate three hand pose estimators on DHG dataset, using the methods proposed by Oberweger *et al.* [22] and Ge *et al.* [10] in addition to the Intel RealSense estimator [11]. We used in these experiments the region-of-interest of the hand returned by Intel RealSense camera as input to the hand pose estimator algorithms instead of a particular hand extraction algorithm, without any preprocessing step. Both estimators [22, 10] were trained on both datasets to select the best training one. Tests showed an improvement of 4% of the recognition accuracy for Oberweger *et al.*'s estimator using IVCL dataset for training. However, they did not reveal any significant effect of the used dataset for Ge *et al.*'s estimator, used for training the pose estimator, on the gesture recognition result. Thus, we choose the IVCL dataset for all the training phase of the two estimators. Fig. 10 shows

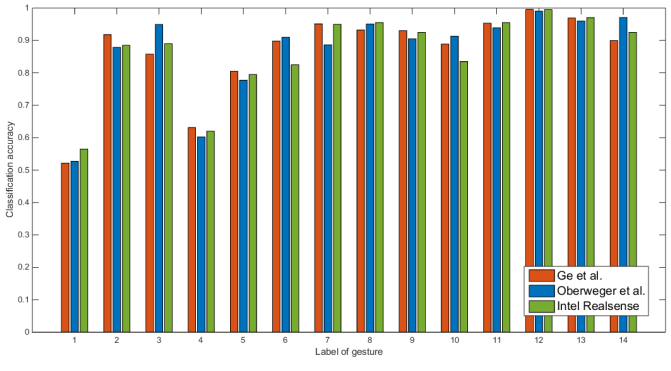


Fig. 10: Recognition accuracies per class of gesture on the DHG-14 dataset following three hand pose estimators.

the recognition accuracies on our DHG-14 dataset per class of gestures. The average accuracies by estimator, available in Table 2, show that our method performs well independently to the pose estimation method.

Table 2: Average recognition accuracies obtained on the DHG-14 dataset using three hand pose estimators.

Hand pose estimation algorithm	Accuracy (%)
Ge *et al.* [10]	86.92
Oberwerger *et al.* [22]	86.24
Intel Realsense [11]	86.86

4.3.2. Assessment of the estimation error impact

To measure how good is the used hand pose estimation for recognizing hand gesture by our approach, we need to swap the estimated 3D hand joints by the ground truth labels in the test set. The only dataset that provides all this information and has significant gestures is MSRA dataset [36]. It contains 9 subjects performing 17 gestures chosen mostly from American Sign Language and each of them varies little in a 500-frame sequence. This dataset, introduced for the hand pose estimation issue, has a large view point coverage, but it has small variations in articulation. However, it is the only available public dataset that contains significant hand gestures with provided annotation. The ground truth of 3D joint hand pose was annotated in a semi-automatic manner, where an optimization method [32] was used and manual refinements were done until convergence. We evaluated our method on the MSRA15 dataset using, both in testing, the hand pose labels and the estimated ones by each hand pose estimator. We used the *Leave-One-Subject-Out* cross-validation protocol for evaluation, where 8 subjects were used for training and the remaining subject for test in turn. Totally 9 experiments are repeated, and the average accuracies is reported in Table 3. For a challenging purpose, the gesture recognition accuracies are computed using the first 100, 200 and all frames of each gesture sequence. Several observations

Table 3: Recognition accuracies obtained by our approach on the MSRA15 dataset using two hand pose estimators compared to the ground truth labels.

Hand pose estimator	100F	200F	Whole sequence
Ground truth	76.4%	96.9%	98%
Ge *et al.* [10]	71.2%	92.8%	98%
Oberwerger *et al.* [22]	72.3%	93%	98%

may be highlighted. First, the recognition accuracies obtained by the two estimation algorithms are very close. Certainly, a larger estimation error lead to worse gesture recognition results. However, a difference of average 3D estimation error lower than 20mm, measured on subsets of IVCL and MSRA15 datasets as reported in [22, 10], obviously has no important effect on our gesture recognition results. Second, a significant difference in hand gesture recognition is observed between using the hand pose labels (ground truth) and using the estimated ones in test set for small subsequence of the gesture. This difference decreases with using more frames until getting the same result when the whole sequence of 500 frames is used. Furthermore, the results obtained show the effectiveness of our approach to recognize gestures from American Sign Language, without ignoring that the gestures in MSRA15 dataset have small variations in articulation and do not show any heterogeneity or strong similarity between classes. We note the absence of results from the state-of-the-art methods in terms of gesture recognition on this dataset since it was introduced for pose estimation issues.

4.4. Hand Gesture Recognition Analysis

4.4.1. DHG 14-28 dataset

To assess the effectiveness of our algorithm to classify the gestures of the DHG dataset into 14 classes, we compare the results obtained by the hand shape and motion descriptors separately. Table 4 presents the accuracies of our approach obtained using each of our descriptors independently and by combining them. For clarity, we divide the results by coarse and fine gestures according to the labels from Table 1, allowing us to analyze the impact of each descriptor on each gesture category.

Table 4: Accuracy comparison fine / coarse / both gesture for the DHG-14 dataset.

Features	Fine (%)	Coarse (%)	Both (%)
Direction	44.60	88.50	72.79
Rotation	50.30	50.61	50.50
SoCJ	67.84	63.12	64.88
SoCJ + Dir. + Rot.	74.43	93.77	**86.86**

Using all descriptors (direction + rotation + SoCJ) presented in Section 3, the final accuracy of our algorithm on the DHG-14 is 86.86%. It rises to 93.77 % recognition for the coarse gestures, but for the fine ones the accuracy drops below 75%. Using only the direction, a large difference can be observed between accuracies obtained for the fine and the coarse gestures, respectively 44.60% and 88.50%. The analysis of the results obtained using only the SoCJ descriptor shows that the hand shape is the most effective feature for the fine gestures with an accuracy of 67.84%. On the other hand, this result shows that the hand shape is also a way to describe coarse gestures with a fair accuracy of 63.12%. If the rotation descriptor shows a low mean accuracy of 50.50% for both fine and coarse gestures, it is a valuable feature for pairs of similar gestures such as *Rotation CW* and *Rotation CCW*. These results confirm the interest of using several descriptors in order to completely describe the hand gestures.

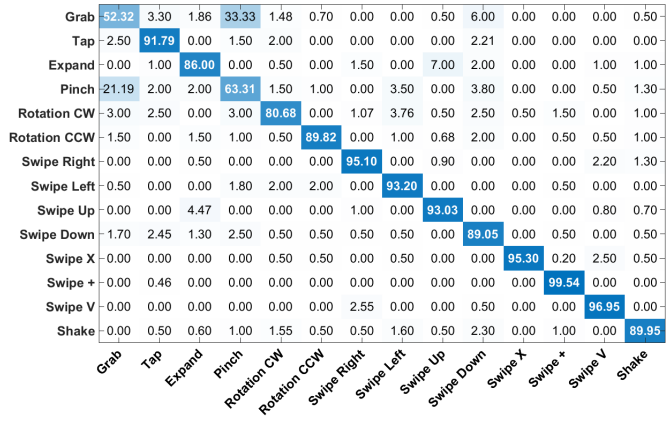


Fig. 11: The confusion matrix of the proposed approach for the DHG-14 dataset.

To better understand the behavior of our approach according to the recognition per class, the confusion matrix is illustrated in Fig. 11. The first observation is that 11 gestures out of 14 have scored higher than 85.00%. The second observation is the low accuracy obtained for certain gestures such as *Grab* is mainly due to the great confusion with *Pinch* gesture. By analyzing their sequences, we find that the algorithms of the hand pose perform well the 3D join estimation. However, we observe that these gestures are very similar and difficult to distinguish even by the human eye. The main difference between them is the hand movement amplitude and our approach does not take this characteristic into account.

With a final accuracy of 86.86% obtained on DHG-14 dataset, we noticed that the recognition of dynamic hand gestures is still challenging. The recognition system has to deal with the considerable differences between gestures performed by different people, resulting in a challenging heterogeneity of the gestures.

Finally, in order to meet the challenge of gesture recognition when performed with different numbers of fingers existing in the DHG-28 dataset, we consider the hand gestures to belong to 28 classes related to the gesture and the way it has been performed (with one finger or the whole hand). The resulting confusion matrix is shown in Fig. 12. Using our approach, we obtain an accuracy of 84.22%. As shown in Table 8, by multiplying the number of classes by two, we lose 2.64% accuracy.

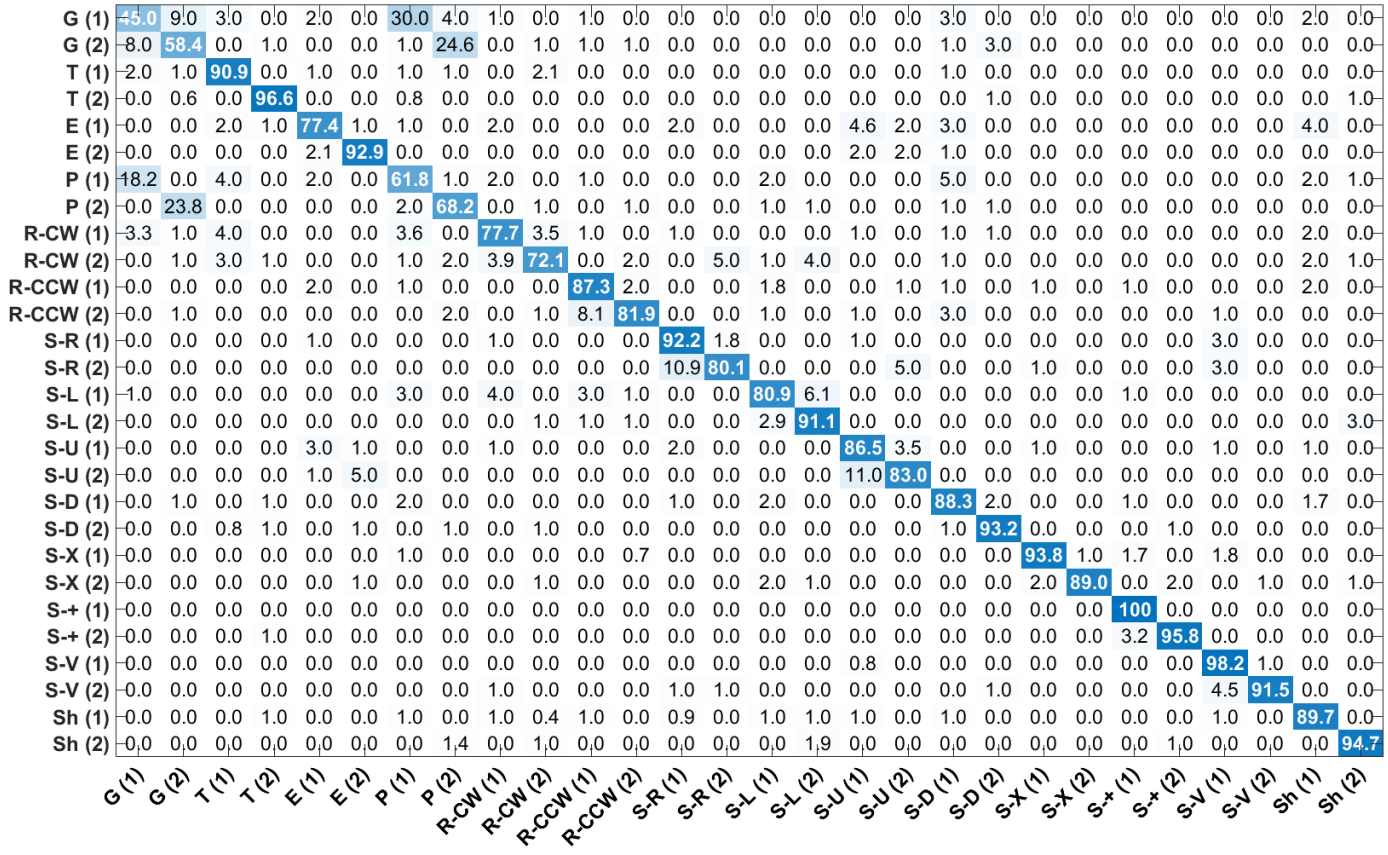


Fig. 12: The confusion matrix obtained by the proposed approach for the DHG-28 dataset. The gestures annotated (1) and (2) were performed respectively using one finger and with the whole hand.

The impact of using a Temporal Pyramid, a Fisher Vector and their combinations in our processing pipeline are also measured and evaluation results of the DHG-14 dataset are shown in Table 5. We note that using SVM directly on features requires the descriptor to have a fixed size. Indeed, we use a linear interpolation to resize the sequence dimension to 35 frames which is the average number of frame of the sequences in the DHG dataset. Obtained results show that adding the FV representation increases the accuracy by 2.87%. The temporal information is very important because it encodes the dynamics of the gesture. Taking this into account in our pipeline – by adding the Temporal Pyramid step – increases the final accuracy by 7.10%.

Table 5: Recognition accuracies obtained on the DHG-14 gesture dataset following different pipeline configurations.

Pipeline	Accuracy (%)
Features + SVM	76.89
Features + FV + SVM	79.76
Features + TP + FV + SVM	86.86

4.4.2. Handicraft-Gesture dataset

To evaluate our approach on the Handicraft-Gesture dataset, we follow the experimental protocol proposed by Lu *et al.* [15], i.e. *Leave-One-Subject-Out* cross-validation. They compute several features based on palm direction, palm normal, fingertip positions, and palm center position. For the classification of temporal sequences they use a Hidden Conditional Neural Field classifier. In Table 6 shows an increase in hand gesture recognition accuracy by 2.11% using our approach.

In order to investigate the impact of each of our descriptors (direction, rotation and SoCJ) on the Handicraft-Gesture

Table 6: Recognition accuracies obtained on the Handicraft-Gesture dataset.

Method	Accuracy (%)
Lu *et al.* [15]	95.00
Ours	**97.11**

Table 7: Recognition accuracies obtained on the Handicraft-Gesture for each descriptor of our approach.

Features	Accuracy (%)
Direction	71.66
Rotation	62.67
SoCJ	92.35
SoCJ + Direction + Rotation	97.11

dataset, Table 7 presents the accuracy obtained using each of those descriptors independently and by concatenating them in only one descriptor. The direction and the rotation of the hand through the movement gave us fair results, respectively 71.66% and 62.67%. However, the score increases to 92.35% using only the SoCJ descriptor. In particular, pottery skills require fine hand gestures which do not contain a lot of motion information, such as the fine gestures of the DHG dataset. These gestures are better described by the hand shape variation, so, that is the reason why the SoCJ will be considered as the most effective feature.

We also note that, as shown in Table 4, combining the descriptors leads to a significant gain in performance. This combination is more useful for the DHG dataset than for the Handicraft-Gesture dataset where adding the motion features improves the recognition rate by only 4.76%. That is explained by the nature of the gestures included in the datasets. The DHG dataset is more heterogeneous as it also contains coarse gestures for which our motion features are important. In fact, both coarse and fine gestures are useful in a human-computer interface. Future gesture recognition algorithms will have to take this difference into account.

4.4.3. Pre-segmented sequences of NVIDIA dataset

To evaluate our approach on the challenging NVIDIA dataset [18], we use the Ge *et al.* hand pose estimator [10] which gives the best recognition accuracy on DHG-14 dataset (see Table 2). We performed the hand region-of-interested extraction step using the same algorithm proposed by [22, 10]. The extracted 3D joint positions of hand from depth images are used as input for our gesture recognition method. Following the same protocol proposed in [18], we randomly split the data by subject into training (70%) and test (30%) sets, resulting in 1050 training and 482 test videos. When considering the pre-segmented sequences of the dataset, our approach obtain an accuracy of 74%. First, with such a recognition accuracy, we went beyond the two handcrafted methods [24, 47] which extract descriptors on the sequence of depth images and obtained respectively 36.3% and 70.7%. Second, deep learning methods outperformed recent results in many domains in computer vision. Following this statement, 3D convolutional layers presented in [39, 18] show particularly reliable accuracies on the task of 3D hand gesture recognition, obtaining, respectively, 78.8% and 80.3% accuracy. Finally, in addition to the recognition challenges, the NVIDIA dataset [18] has been created to study the detection of gestures. Indeed, an unsegmented stream of gestures contains a lot of unwanted and meaningless hand motions that do not belong to none of the gesture categories. A prior gesture detection is required before the recognition process.

4.4.4. Comparison with state-of-the-art methods

We compare our approach with four state-of-the-art methods on the DHG dataset. We chose two depth-based descriptors: HOG^2 proposed by Ohn-Bar *et al.* [23] and HON4D proposed by Oreifej *et al.* [26]. We also compare our approach to a skeleton-based method proposed by Devanne *et al.* [4] showing a good accuracy for human action recognition. Devanne's approach is based on a similarity metric of human trajectories using the shape of 3D body skeleton in a Riemannian manifold. Finally, we compare the hand shape descriptor SoCJ with a similar state-of-the-art feature called Skeletal Quad defined by Evangelidis *et al.* [7]. The publicly available source codes

of these methods are used in our experiments.

For the two depth-based descriptors [23, 26], pre-processing steps on the depth sequences are needed. First, using a suitable threshold, we clean the image by removing the background and the body of the subject keeping only the region-of-interest of the hand. Then, we crop the size of the images by removing all regions where the hand does not appear along the sequence. For the HON4D method, we choose a spatio-temporal grid of size $5 \times 5 \times 3$ since it gives the best accuracy. For Evangelidis *et al.* method [7], in order to properly compare the hand shape descriptors, we use our approach by swapping our SoCJ descriptor with the Skeletal Quad one while keeping the rotation and direction features.

Table 8 analyzes the results obtained by the methods cited previously using 14 and 28 gestures on the DHG dataset. We note that our approach outperformed, with an accuracy of 86.86%, the two depth-based descriptors showing the promising direction of using skeletal data for hand gesture recognition. The accuracy of the action recognition method [4] applied for 3D hand joints trajectories obtained a score of 76.61% of recognition. It shows that an action recognition approach is unsuitable for hand gesture recognition as hand trajectories are not distinctive features for different gestures.

Table 8: accuracy comparison 14 / 28 gestures for the DHG dataset.

Method	14 gestures (%)	28 gestures (%)
Ohn-Bar *et al.* [23]	81.85	76.53
Oreifej *et al.* [26]	75.53	74.03
Devanne *et al.* [4]	76.61	62.00
Evangelidis *et al.* [7]	84.50	79.43
Xu *et al.* [46]	50.32	30.85
Ours	**86.86**	**84.22**

When we apply these methods on 28 classes, the HOG^2 descriptor [23], which had a good result on 14 gestures, obtains 76.53% of accuracy. The depth-based methods do not handle enough hand shape information to deal with the challenge of classifying hand gestures performed with different numbers of fingers. We note that Devanne's approach loses 14.61% of recognition rate on this experiment showing that the method, if it gives good result on action recognition dataset, it is unsuitable for fine and dynamic hand gesture recognition.

Evangelidis *et al.* [7] propose a local body skeleton descriptor that encodes the relative position of joint quadruples. It requires a *Similarity Normalization Transform* (SNT) that leads to a compact (6D) view-invariant skeletal feature, referred to as Skeletal Quad. Because of the SNT, their descriptor takes more computation time and is less suitable for hand shape description as they lost information about the distance between joints. The accuracy of the DHG-28 dataset using their hand shape descriptor decreases by 4% compared to the SoCJ descriptor on this task.

Xu *et al.* [46] propose a method to recognize dynamic hand gestures (Arabic numbers from 0 to 9) using Leap Motion controller. They use only the three-dimensional space trajectories of the palm position, from which they construct their features as a vector of 81 items representing the orientation angles, the relative direction angles and the distance between the starting and the end point of the gesture sequence. The elements of the feature vector are sorted according to two criteria: the amplitude of the variation and the appearance of the features. A multi-class SVM classifier with RBF kernel is then used to perform the recognition after a training phase. The results obtained by this method on the DHG-14 dataset show the insufficiency of 3D-trajectory of palm joint to characterize our heterogeneous gestures. Only 50.32% of 14 gestures are correctly recognized, with an accuracy of 63.17% for coarse gestures versus 27.20% for fine gestures. This insufficiency is due to the type of the gestures performed with important variations of both hand motion and shape. It is interesting to note that with an accuracy around 63%, Xu *et al.*'s descriptor [46] shows a performance comparable to our SOCJ descriptor alone for coarse gestures, but it fails with fine gestures giving only an accuracy of 30% compared to 67.84% obtained with SOCJ. Finally, their approach encounters difficulties to discriminate the same gestures made by a finger or by the whole hand, where the overall accuracy drops from 50% for 14 gestures to 30% for 28 gestures.

Table 9: accuracy comparison coarse / fine gestures for the DHG-14 dataset.

Method	Coarses (%)	Fines (%)
Ohn-Bar *et al.* [23]	86.00	71.60
Oreifej *et al.* [26]	83.88	60.50
Devanne *et al.* [4]	86.61	58.60
Evangelidis *et al.* [7]	92.22	70.62
Xu *et al.* [46]	63.17	27.20
Our approach	**93.77**	**74.43**

In Table 9, we investigate the impact of the different methods on the fine and coarse gestures separately. We remind that coarse gestures are defined by the motion of the hand in space and fine gestures are more perceptible by the variation of the hand shape along the sequence. The statement of a need of precision in the field of dynamic hand gesture recognition is also shown in this experiment. Except for the HOG^2 descriptor [23], if [26] and [4] give honorable results in the task of coarse gesture classification, they show a lack of precision generating a recognition rate below 61% when trying to classify fine gestures. If our approach gives the best results with 74.43% of correctly labeled fine gestures, we note it also needs improvements.

4.5. Latency Analysis and Computation Time

For many applications, making a potentially unreliable forced decision based on partial available frames is a real challenge. The goal of the following experiments is to automatically determine when a sufficient number of frames are observed to provide a reliable recognition of the occurring gesture, hence the term *low-latency* recognition. The *latency* can be defined as the time lapse between the moment when a sequence is given to the algorithm and the instant when the system recognizes the performed gesture. We will study two characteristics: computational and observational latency. The computational latency is the time the system takes to perform the recognition process. The observational latency represents the percentage of a continuous gesture needed by a system in order to perform its recognition.

4.5.1. Computational latency

The computational time is a very important characteristic of a hand gesture recognition algorithm as it should be working in real time for some HCI applications. We evaluate the computational latency of our approach on the DHG-14 dataset, using a MATLAB implementation with an Intel Xeon CPU E3 3.40 GHz and 8 GB RAM. Since the proposed approach is based only on skeletal joint coordinates, it is simple to calculate and it needs only a small computation time. Table 10 reports the minimum, average and maximum computation time for each step of our approach. For the whole recognition process, the average computation time is 0.2502s for a sequence of 35 frames. This time makes our approach suitable for real-time recognition. We note that 88.49% of this time is taken by the classification process.

4.5.2. Observational latency

To analyze the observational latency of our approach, we show how the accuracy depends on the percentage of the sequence we currently have. New recognition rates are computed by processing only a percentage of the sequence. In each case, we cut the training sequences into shorter ones to create a new training set. During the classification step, we also cut the test sequences to the corresponding length and apply our method with the same learning protocol *Leave-One-Subject-Out* cross validation. Fig. 13 shows the observational latency of our approach on the DHG-14 and the Handycraft dataset. We see that accuracy close to the maximum is obtained using 60% on the sequences on both datasets. In other words, the evaluations in terms of latency have revealed the efficiency of our approach for rapid gesture recognition. It is possible to recognize a gesture from DHG-14 dataset composed of 50 frames up to 80.82% seeing only 30 frames (versus 86.86% using all the frames). Thus, our approach can be used for interactive systems, notably, in entertainment applications to resolve the problem of lag and improve some gesture-based games. This shows that the computational latency can be masked by the observational latency in the cases where sequences are nearly twice as long as the computational latency.

Table 10: Computation time in second for each step of our approach on the DHG-14 dataset. We note that some steps are dependent of the size of the sequence. We report the time for the smallest sequence (N_f = 8), the mean size over all the sequence (N_f = 35) and the biggest sequence (N_f = 150).

Step	Mins (sec)	Averages (sec)	Maxs (sec)
Normalization of hand size	0.0038	0.0154	0.0640
Direction descriptor	0.0002	0.0011	0.0045
Rotation descriptor	0.0001	0.0005	0.0026
Registration of the hand	0.0009	0.0038	0.0157
SoCJ descriptor	0.0005	0.0022	0.0089
FV and TP construction	0.0033	0.0058	0.0188
Classification	0.1905	0.2214	0.2150
Total	**0.1993**	**0.2502**	**0.3295**

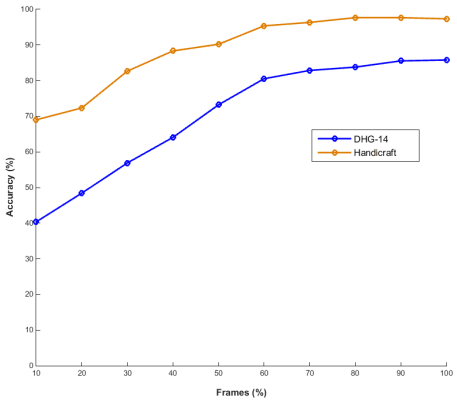


Fig. 13: Observationnal latency analysis on the DHG-14 and Handicraft-Gesture datasets. The accuracy of the y-axis is obtained by processing only the percentage of the sequence shown in the x-axis.

4.6. Online recognition of continuous streams

An unsegmented stream of gestures contains a lot of unwanted and meaningless hand motions that do not belong to none of the gesture categories. First, hand gesture movements are often composed of three phases: (1) the *pre-stroke* phase which occurs before the relevant gesture when the user needs to put its hand in a starting position. For example, it is the movement the user performs to move the hand from its restful position to a place where the camera can see the hand. (2) The *nucleus* phase, where the hand gesture is performed and have meanings. (3) The *post-stroke* phase, which occurs after the relevant gesture when the user wants to move back its hand to a restful position.

Additionally, a stream of gestures contains motions between the gestures.For example, in a human-computer interface based on hand gestures in a car scenario, while the user is not performing a gesture, his hands are still moving to control the vehicle and, so, contains a lot of parasitical hands motions. A challenge of online hand gesture recognition is to detect and extract only hand motions from nucleus phases in order to improve the gesture recognition accuracy.

4.6.1. Extension of our approach to the online scenario

In real applications, we do not have information about when and where the hand gesture is going to be performed. Neverova *et al.* [21] added a binary classification phase before the classification process using {*gesture*, *no_gesture*} labels. Following this reasoning, we propose to learn a prior light binary classifier trained with the same feature vector proposed in our approach, to perform a gesture localization task before the final classification. Contrary to Neverova *et al.*'s frame-based method, we employ a temporal sliding window on which our feature vector will be computed to feed the binary classifier. If the localization stage classifies a current sliding window as *gesture*, it is fed to the final classification phase, it is rejected otherwise.

4.6.2. Gesture detection from NVIDIA dataset

The NVIDIA dataset [18] has been captured following a human-computer interaction based on hand gestures in a car scenario. While the user is not performing a gesture, its

hands still move to control the vehicle and, so, is highly suitable to study gesture detection. The ground-truth annotations of the NVIDIA dataset provide additional informations about the nucleus locations. We use the nucleus annotations as a ground truth following a binary categorical variable {*gesture*, *no_gesture*}, to learn our prior binary classifier, chosen here as a linear SVM. In practice, we experimented candidate windows of different size 20, 30 and 40 frames, by sliding the window in step of 50% of its size. To measure the performance, we compute the average precision score as in [25] considering a detection as correct when it overlaps by at least 20% with a ground truth annotation. On the task of window-wise binary classification following labels {*gesture*, *no_gesture*}, our localization phase has obtained an accuracy of 85.45% with windows of size 20 sliding in step of 10 frames, following the same protocol as in [18]. This accuracy reveals that some sequence windows, which are *nucleus* phases in certain gesture sequences are detected as *No-gesture*, which are then completely rejected before the gesture recognition phase. Furthermore, the detected sequences as *Gesture* may contain false positive ones

4.6.3. Online recognition from NVIDIA dataset

The gesture localization phase allows the elimination of the *no_gesture*} portions, and consequently to detect the isolated gestures which will form the entry of the final multi-class classifier. We compare in Table 11 our approach to two handcrafted methods (HOG+HOG^2 [24] and Super Normal Vector (SNV) [47]) and two deep learning methods (C3D [39] and R3DCNN [18]). Molchanov et al. [18] used a recurrent layer after a 3D convolution in order to model the global temporal information. To tackle the detection challenges, they used a Connectionist Temporal Classification (CTC) loss function to distinguish unwanted and meaningful hand motions and, so, to detect gestures along the stream. It must be noticed that we show results obtained by the state-of-the-art methods using only the depth information. Despite a moderate gesture detection phases (85.45%), we went through the entire recognition process. Compared to our previous results in Section 4.4.3, we obtain an overall recognition accuracy of 78.0% comparable to those obtained by the deep learning method C3D [39], on the NVIDIA dataset. The recognition accuracies per class of gesture obtained by our approach and by the R3DCNN method [18] are also presented in Fig. 14. The first observation is that static gestures (e.g. *Show up one, two or three fingers* (12 - 14), or yet *closing the hand twice* (23), *Ok sign* (24) and *Thumb up* (25)) show accuracy higher than 80%. An other interesting statement is that *Swipe up, Swipe 2 Fingers Left and Right* gestures (3, 5 and 6) reach accuracies higher than 80%. While *Swipe Right and Down* gestures (2 and 4) show weak results, respectively, 65% and 49%. Different phases of inverse gestures contain high similarities, which leads to false positive detections. Indeed, some sequence windows, which belong to either *No-gesture* or *pre-stroke* (or *post-stroke*) phases in certain gesture sequences, were detected in this case as *nucleus* phase, before being sent to the gesture recognition phase. For example, the pre-stroke phase of a *Swipe left* gesture consists in moving the hand to the right so that the camera is able to see the entire gesture. However, this movement to the right can be seen as a *Swipe right* gesture nucleus by the localization algorithm and not as a pre-stroke phase of a *Swipe left* gesture. This may partly explains the low accuracy of 65% obtained for *Swipe Right* gesture. Despite the overall superiority of the R3DCNN method, our approach provided more accurate recognition accuracy for eight gestures: hand up (69% to 85%), two fingers left (68% to

Table 11: Comparison of our method to the state-of-the-art methods on depth images of the NVIDIA Dynamic Hand Gestures dataset. HC: Hand-Crafted, DL: Deep Learning

Method	Type	Data	Acc. (%)
HOG+HOG^2 [24]	HC	Depth	36.3
SNV [47]	HC	Depth	70.7
C3D [39]	DL	Depth	78.8
R3DCNN [18]	DL	Depth	80.3
Ours (online)	HC	3D Hand Skeletal	78.0
Ours + manual detection	HC	3D Hand Skeletal	83.3

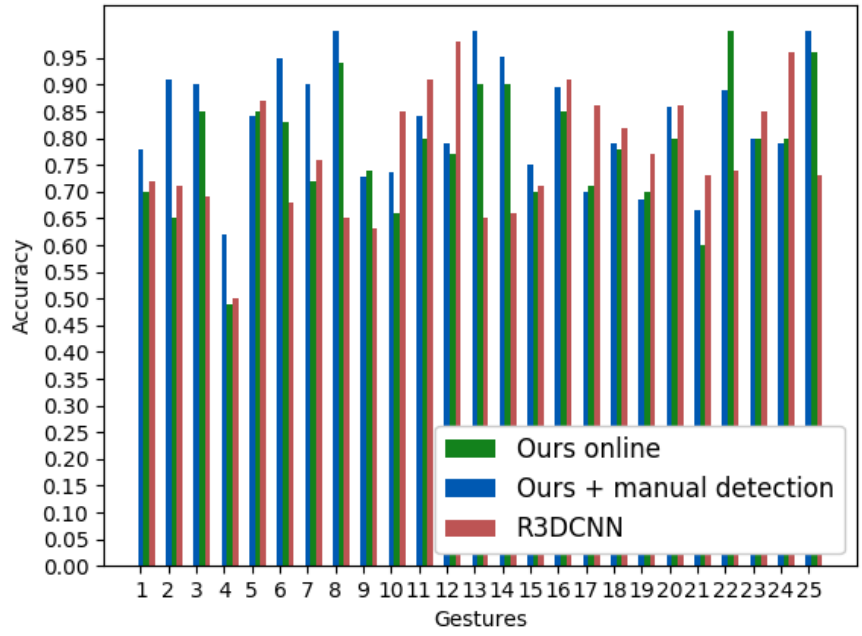


Fig. 14: Comparison of recognition accuracies per class of gestures on the NVIDIA Dynamic Hand Gestures dataset. The corresponding class labels are given in the Subsection 4.1.3

83%), two fingers down (65% to 94%), showing one (65% to 90%) or two (66% to 90%) fingers, one (63% to 74%) or two (74% to 100%) fingers forward , "OK" sign (73% to 96%). It is interesting to note that the R3DCNN method outperformed our approach mostly on gestures with an open hand and where the gesture can be described mainly by the movement in space. In contrast, our method surpasses the recognition of same gesture types performed with a different number of fingers. In Fig. 14, we show accuracies obtained by our method with a prior manual gesture detection step compared to manual detection in addition to R3DCNN methods. The manual detection experiment consists of providing the ground truth of nucleus to our multi-class classifier. This experiment reveals that the detection step for hand gesture recognition is essential as it improves our previous result on presegmented streams presented in Fig. 9 by 3.57%. However, there is room to improve the effectiveness of gesture detection phase as shown from Fig. 14, where a recognition of 83.3% can be reached with a manual detection of gesture. Moreover, the R3DCNN presented in [39] is composed of a large neural network with more than 70 million parameters. They use a NVIDIA DIGITS DevBox with four Titan X GPUs to train and predict a new incoming gesture. Currently, this configuration, imposing constraints in terms of computational resources, is not always available and therefore not necessarily suitable for real applications. Comparatively, with a single stream of skeletal data, minimal preprocessing, limited memory, little parameter tuning and without any data augmentation, we have obtained promising results with our approach.

5. Conclusion

In this work, we explore a way to classify dynamic hand gestures using hand skeletal representation. We proposed a method using three gestural features representing the hand shape and the motion information computed on these new data in addition to a temporal encoding of the gesture dynamics. The evaluation of our approach shows a promising way to perform hand gesture recognition with a skeletal-based approach. Experiments are carried out on three hand gesture datasets, contain-

ing a set of fine and coarse heterogeneous gestures captured in different scenarios. Furthermore, the evaluation of our approach in terms of latency demonstrates improvements for a low-latency hand gesture recognition systems, where an early recognition is needed. Comparative results with state-of-the-art methods demonstrate that our approach outperforms existing handcrafted depth-based approaches. In the future, we plan to focus on neural networks (LSTM, HCNF, CNN, etc.) in order to better represent the complex dynamic and temporal information of a hand gesture. Gesture detection phase in on-line scenario could also be improved for more efficient early gesture recognition.

References

[1] Chang, C.C., Lin, C.J., 2011. LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology 2, 27:1–27:27.

[2] Cheng, H., Dai, Z., Liu, Z., 2013. Image-to-class dynamic time warping for 3d hand gesture recognition. 2013 IEEE International Conference on Multimedia and Expo (ICME) , 1–6doi:10.1109/ICME.2013.6607524.

[3] De Smedt, Q., Wannous, H., Vandeborre, J.P., 2016. Skeleton-based dynamic hand gesture recognition. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops , 1–9.

[4] Devanne, M., Wannous, H., Berretti, S., Pala, P., Daoudi, M., Bimbo, A.D., 2015. 3-d human action recognition by shape analysis of motion trajectories on riemannian manifold. IEEE Transactions on Cybernetics 45, 1340–1352. doi:10.1109/TCYB.2014.2350774.

[5] Dong, C., Leu, M.C., Yin, Z., 2015. American sign language alphabet recognition using microsoft kinect. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR) Workshops , 44–52doi:10.1109/CVPRW.2015.7301347.

[6] Escalera, S., Baró, X., Gonzalez, J., Bautista, M.A., Madadi, M., Reyes, M., Ponce-López, V., Escalante, H.J., Shotton, J., Guyon, I., 2014. Chalearn looking at people challenge 2014: Dataset and results. Computer Vision - ECCV Workshops , 459–473.

[7] Evangelidis, G., Singh, G., Horaud, R., 2014. Skeletal quads: Human action recognition using joint quadruples. IEEE International Conference on Pattern Recognition (ICPR) , 4513–4518doi:10.1109/ICPR.2014.772.

[8] Feix, T., Pawlik, R., Schmiedmayer, H.B., Romero, J., Kragic, D., 2009. A comprehensive grasp taxonomy. Robotics, Science and Systems: Workshop on Understanding the Human Hand for Advancing Robotic Manipulation , 2–3.

[9] Ge, L., Liang, H., Yuan, J., Thalmann, D., 2016a. Robust 3d hand pose estimation in single depth images: from single-view cnn to multi-view cnns, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3593–3601.

[10] Ge, L., Liang, H., Yuan, J., Thalmann, D., 2016b. Robust 3d hand pose estimation in single depth images: from single-view cnn to multi-view cnns. IEEE Conference on Computer Vision and Pattern Recognition , 3593–3601.

[11] Intel SR300, . URL: https://click.intel.com/intelrealsense-developer-kit-featuring-sr300.html.

[12] Keskin, C., Kıraç, F., Kara, Y.E., Akarun, L., 2012. Hand pose estimation and hand shape classification using multi-layered randomized decision forests, in: European Conference on Computer Vision, Springer. pp. 852–863.

[13] Kurakin, A., Zhang, Z., Liu, Z., 2012. A real time system for dynamic hand gesture recognition with a depth sensor. 20th European Signal Processing Conference (EUSIPCO) , 1975–1979.

[14] Kuznetsova, A., Leal-Taix, L., Rosenhahn, B., 2013. Real-time sign language recognition using a consumer depth camera. IEEE International Conference on Computer Vision Workshops (ICCVW) , 83–90doi:10.1109/ICCVW.2013.18.

[15] Lu, W., Tong, Z., Chu, J., 2016. Dynamic hand gesture recognition with leap motion controller. IEEE Signal Processing Letters 23, 1188–1192.

[16] Marin, G., Dominio, F., Zanuttigh, P., 2015. Hand gesture recognition with jointly calibrated leap motion and depth sensor. Multimedia Tools and Applications , 1–25.

[17] Molchanov, P., Gupta, S., Kim, K., Kautz, J., 2015. Hand gesture recognition with 3d convolutional neural networks. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR) Workshops , 1–7.

[18] Molchanov, P., Yang, X., Gupta, S., Kim, K., Tyree, S., Kautz, J., 2016. Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural network. IEEE Conference on Computer Vision and Pattern Recognition , 4207–4215.

[19] Monnier, C., German, S., Ost, A., 2014. A multi-scale boosted detector for efficient and robust gesture recognition. Computer Vision - ECCV Workshops , 491–502.

[20] Neverova, N., Wolf, C., Taylor, G., Nebout, F., 2016a. Moddrop: adaptive multi-modal gesture recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 38, 1692–1706.

[21] Neverova, N., Wolf, C., Taylor, G., Nebout, F., 2016b. Moddrop: adaptive multi-modal gesture recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 38, 1692–1706.

[22] Oberweger, M., Wohlhart, P., Lepetit, V., 2015. Hands deep in deep learning for hand pose estimation. Computer Vision Winter Workshop (CVWW) .

[23] Ohn-Bar, E., Trivedi, M.M., 2013. Joint angles similarities and hog2 for action recognition. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR) Workshops - HAU3D .

[24] Ohn-Bar, E., Trivedi, M.M., 2014. Hand gesture recognition in real time

for automotive interfaces: A multimodal vision-based approach and evaluations. IEEE Transactions on Intelligent Transportation Systems 15, 2368–2377.

[25] Oneata, D., Verbeek, J.J., Schmid, C., 2013. Action and event recognition with fisher vectors on a compact feature set, in: ICCV, IEEE Computer Society. pp. 1817–1824.

[26] Oreifej, O., Liu, Z., 2013. Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR) .

[27] Peng, X., Zou, C., Qiao, Y., Peng, Q., 2014. Action recognition with stacked fisher vectors. European Conference on Computer Vision - ECCV , 581–595.

[28] Perronnin, F., Sánchez, J., Mensink, T., 2010. Improving the fisher kernel for large-scale image classification. Computer Vision - ECCV , 143–156.

[29] Potter, L.E., Araullo, J., Carter, L., 2013. The leap motion controller: a view on sign language. Australian computer-human interaction conference: augmentation, application, innovation, collaboration , 175–178.

[30] Pudil, P., Novovičová, J., Kittler, J., 1994. Floating search methods in feature selection. Pattern recognition letters 15, 1119–1125.

[31] Pugeault, N., Bowden, R., 2011. Spelling it out: Real-time asl fingerspelling recognition. IEEE International Conference on Computer Vision Workshops (ICCV Workshops) , 1114–1119.

[32] Qian, C., Sun, X., Wei, Y., Tang, X., Sun, J., 2014. Realtime and robust hand tracking from depth, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1106–1113.

[33] Ren, Z., Yuan, J., Meng, J., Zhang, Z., 2013. Robust part-based hand gesture recognition using kinect sensor. IEEE Transactions on Multimedia 15, 1110–1120.

[34] Sánchez, J., Perronnin, F., Mensink, T., Verbeek, J., 2013. Image classification with the fisher vector: Theory and practice. International Journal of Computer Vision 105, 222–245.

[35] Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., Blake, A., 2011. Real-time human pose recognition in parts from a single depth image. IEEE International Conference on Computer Vision on Pattern Recognition (CVPR) .

[36] Sun, X., Wei, Y., Liang, S., Tang, X., Sun, J., 2015. Cascaded hand pose regression. IEEE Conference on Computer Vision and Pattern Recognition , 824–832.

[37] Tang, D., Chang, H.J., Tejani, A., Kim, T.K., 2014. Latent regression forest: Structured estimation of 3d articulated hand posture. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR) , 3786–3793doi:10.1109/CVPR.2014.490.

[38] Tompson, J., Stein, M., Lecun, Y., Perlin, K., 2014. Real-time continuous pose recovery of human hands using convolutional networks. ACM Transactions on Graphics (ToG) 33, 169.

[39] Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M., 2015. Learning spatiotemporal features with 3d convolutional networks. The IEEE International Conference on Computer Vision (ICCV) .

[40] Vemulapalli, R., Arrate, F., Chellappa, R., 2014. Human action recognition by representing 3d skeletons as points in a lie group, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 588–595.

[41] Vogler, C., Metaxas, D., 2001. A framework for recognizing the simultaneous aspects of american sign language. Computer Vision and Image Understanding 81, 358 – 384. URL: http://www.sciencedirect.com/science/article/pii/S1077314200908956, doi:https://doi.org/10.1006/cviu.2000.0895.

[42] Wang, C., Liu, Z., Chan, S.C., 2015. Superpixel-based hand gesture recognition with kinect depth camera. IEEE Transactions on Multimedia 17, 29–39.

[43] Wang, H., Schmid, C., 2013. Action recognition with improved trajectories. IEEE International Conference on Computer Vision , 3551–3558.

[44] Wang, H., Wang, Q., Chen, X., 2012. Hand posture recognition from disparity cost map. Asian Conference on Computer Vision (ACCV - part II) 7725, 722–733.

[45] Wang, J., Liu, Z., Wu, Y., 2014. Learning actionlet ensemble for 3d human action recognition, in: Human Action Recognition with Depth Cameras. Springer, pp. 11–40.

[46] Xu, Y., Wang, Q., Bai, X., Chen, Y.L., Wu, X., 2014. A novel feature extracting method for dynamic gesture recognition based on support vector machine. IEEE International Conference on Information and Automation (ICIA) , 437–441.

[47] Yang, X., Tian, Y., 2014. Super normal vector for activity recognition using depth sequences. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR) .

[48] Ye, Q., Yuan, S., Kim, T.K., 2016. Spatial attention deep net with partial pso for hierarchical hybrid hand pose estimation, in: European Conference on Computer Vision, Springer. pp. 346–361.

[49] Zen, G., Porzi, L., Sangineto, E., Ricci, E., Sebe, N., 2016. Learning personalized models for facial expression analysis and gesture recognition. IEEE Transactions on Multimedia 18, 775–788.

[50] Zhang, C., Yang, X., Tian, Y., 2013. Histogram of 3d facets: A characteristic descriptor for hand gesture recognition. IEEE Conference and Workshops on Automatic Face and Gesture Recognition (FG) , 1–8doi:10.1109/FG.2013.6553754.

[51] Zhou, X., Wan, Q., Zhang, W., Xue, X., Wei, Y., 2016. Model-based deep hand pose estimation. arXiv preprint arXiv:1606.06854 .

[52] Zhu, Y., Xu, G., Kriegman, D.J., 2002. A real-time approach to the spotting, representation, and recognition of hand gestures for humancomputer interaction. Computer Vision and Image Understanding 85, 189 – 208. URL: http://www.sciencedirect.com/science/article/pii/S1077314202909677.