A fast and accurate motion descriptor for human action recognition applications

Enjie Ghorbel, Rémi Boutteau, Jacques Bonnaert, Xavier Savatier, Stéphane Lecoeuche

A fast and accurate motion descriptor for human action recognition applications

Enjie Ghorbel* ** , Rémi Boutteau**, Jacques Bonnaert*, Xavier Savatier**, Stéphane Lecoeuche*
*Mines Douai, IA, F-59508
Douai, France
Email:name.surname@mines-douai.fr
**Institut de Recherche en Systèmes Electroniques Embarqués (IRSEEM), ESIGELEC
Université de Rouen, EA4353, Rouen, France
Email:name.surname@esigelec.fr

***Abstract*—With the availability of the recent human skeleton extraction algorithm introduced by Shotton et al. [1], an interest for skeleton-based action recognition methods has been renewed. Despite the importance of the low-latency aspect in applications, it can be noted that the majority of recent approaches has not been evaluated in terms of computational cost. In this paper, a novel fast and accurate human action descriptor named Kinematic Spline Curves (KSC) is introduced. This descriptor is built by interpolating the kinematics of joints (position, velocity and acceleration). To overcome the anthropometric and the execution rate variabilities, we respectively propose the use of a skeleton normalization and a temporal normalization. For this purpose, a new temporal normalization method based on the Normalized Accumulated kinetic Energy (NAE) of the human skeleton is suggested. Finally, the classification step is performed using a linear Support Vector Machine (SVM). Experimental results on challenging benchmarks show the efficiency of our approach in terms of recognition accuracy and computational latency.**

I. Introduction

Nowadays, action recognition is increasingly attracting interest of researchers in the field of computer vision due to its several applications in Human Computer Interaction (HCI), e-health, gaming, surveillance, etc. Until now, numerous of popular methods has been designed using RGB (Red Green Blue) videos [2]. Nevertheless, this modality has many drawbacks such as sensitivity to body segmentation, illumination changes, viewpoint changes and occlusions.

For this reason, the emergence of depth cameras has encouraged many scientists to use two other modalities (depth maps and skeleton sequences). Indeed, RGB-D (Red Green Blue Depth) cameras provide an additional modality known as depth maps. Furthermore, with the work of Shotton et al. [1], it became feasible to extract relatively accurate skeletons from depth maps in real-time (around 45ms for skeleton extraction per frame according to [3]). Although motion capture systems provide more accurate skeletons, the RGB-D cameras remain an interesting alternative given their lower cost. Thus, RGB-based action recognition methods can be divided according to the chosen modality: Depth-based descriptors and skeleton-based descriptors. While depth-based descriptors are generally more robust to noise and occlusions and more accurate, skeleton-based descriptors are faster to compute, less sensitive to view-point variation and are therefore more adapted to real-world applications [4][5].

Many recent skeleton-based descriptors have shown their ability to accurately recognize actions [6], [7], [8]. Nonetheless, the low-latency challenge is very often neglected despite its importance in applications. The performance of motion descriptors can be seen as a trade-off between good accuracy of recognition and low latency as formulated in [9]. The latency is defined as the sum of computational latency (the time required for calculation) and observational latency (the time of observation necessary before making a good decision). In this paper, we will mainly focus on computational latency because the actions are assumed to have already been segmented. Many off-line applications still require a quick decision such as medical rehabilitation, coaching, gaming, etc.

Motivated by the challenge of carrying out an accurate action recognition while retaining low computational latency, we introduce a novel human skeleton-based descriptor referred as Kinematic Spline Curves (KSC). To ensure the performance of KSC, we propose the succession of simple but efficient processes. First, a skeleton normalization is used to reduce the negative effect of anthropometric variability. Second, to overcome the temporal variability whilst avoiding an excessive increase of computational cost, a temporal normalization algorithm is introduced. The main idea of this normalization is to interpolate kinematic features considering them as functions of Normalized Accumulated kinetic Energy (NAE) (8). To perform the final classification, a linear SVM is used.

This paper is organized as follows: Section II presents an overview of state-of-the-art skeleton-based human action descriptors. Then, the proposed method is detailed in Section III, while in Section IV the experimental results are presented. Finally, conclusions and future work are drawn in Section V.

II. Related work

In this section, we present a brief review of skeleton-based methods for human action recognition. These methods can be categorized as pose-based approaches, geometric approaches and kinematic-based approaches.

A. Pose-based approaches

They count among the first skeleton-based methods for action recognition. Li et al. [6] introduced the 3D bag of points which are used to build an action graph. In [10], a Histogram Oriented of Joints (HOJ) for each posture is built. Then, the classification is done based on Hidden Markov Models (HMM) which describe the evolution of postures. However, these features have shown their limitations because of their sensitivity to anthropometric variability. In this way, many methods began to use relative joint positions. For instance, we can cite Eigenjoint features[11] which contain the information of spatial and temporal distances between joints. A Principal Component Analysis is used to reduce the high dimension of feature vectors.

B. Geometric approaches

Recently, many papers have been inspired from euclidean geometry or differential geometry. Evangelidis et al. [8] introduced skeletal quads which represent quadruples containing the information of similarity transformations between segments. On the other hand, Vemulapalli et al. [7] chose to define transformation matrices of the Special Euclidean group $SE(3)$ between every couple of adjacent segments. Thus, each pose is represented by an element (a point) of $SE^n(3)$, where n represents the number of segment connections. To obtain curves on the Lie group $SE^n(3)$ which are compared via a Dynamic Time Warping algorithm (DTW), an interpolation is done after switching to $se^n(3)$, the Lie algebra associated with $SE^n(3)$. These geometric approaches are theoretically very interesting. However, numerical calculation of these sophisticated methods can lead to an increase in recognition error and in computational latency.

C. Kinematic-based approaches

Skeleton have been widely used in bio-mechanical studies [12]. To describe human motion, Zanfir et al. [13] calculated joint kinematics (position, velocity and acceleration) thanks to the discrete information of joint positions. Each kinematic value is empirically weighted to specify its contribution in the classification. The experiments have shown the efficiency of this method. However, the influence of weight parameters is not well clarified in this paper. Furthermore, it seems that the discontinuity of features can generate limitations in some cases. For example, noisy skeletons or an important execution rate variability can negatively impact the results.

In this article, we propose to exploit kinematic features which do not require an important computational cost. However, instead of using the aperiodic information of joint kinematics, we propose to interpolate and to uniformly sample them in order to obtain periodic features. In addition, a temporal normalization is proposed to overcome the execution rate variability representing an important challenge on action recognition.

III. Kinematic Spline Curves (KSC)

This section presents the novel human action descriptor called KSC. Figure 1 illustrates an overview of the different processes carefully selected in order to perform an accurate and fast action recognition. First, a skeleton normalization is proposed to alleviate the anthropometric variability. Then, Kinematic Features (KF) are computed from the discrete information of normalized joint positions. To overcome the execution rate variability, a temporal normalization based on Normalized Accumulated kinetic Energy (NAE) is introduced. Thus, KF are expressed as functions of NAE (instead of time) and are then interpolated using a cubic spline interpolation algorithm. Finally, to obtain KSC descriptor, a periodic sampling of continuous Kinematic Spline Curves KSC^c is carried out.

A. Spatial Normalization (SN)

An action can be represented by a sequence of N skeletons, while each skeleton is composed of n joints and contains the information of 3D position $p_{\mathrm{j}}(t_k)$ (1) of each joint *j*. t_k refers to the frame index ($t_k \in [t_1, ..., t_N]$).

$$p_{\mathrm{j}}(t_k) = [x_{\mathrm{j}}(t_k), y_{\mathrm{j}}(t_k), z_{\mathrm{j}}(t_k)] \tag{1}$$

Therefore, this skeleton sequence can be seen as a multidimensional time series (2).

$$p(t_k) = [p_1(t_k), p_2(t_k), ..., p_{\mathrm{n}}(t_k)] \tag{2}$$

Inspired by bio-mechanical studies, the hip joint is assumed to be the origin. For this reason, hip joint coordinates are subtracted from each joint coordinates (3).

$$p^{\mathrm{abs}}(t_k) = [p_1(t_k) - p_{\mathrm{hip}}, p_2(t_k) - p_{\mathrm{hip}}, ..., p_{\mathrm{n}}(t_k) - p_{\mathrm{hip}}] \tag{3}$$

It can be easily shown that anthropometric variability negatively affects the recognition task. To palliate this kind of variability, a skeleton normalization inspired from [13] is employed. However, it remains some differences between our algorithm and the normalization of [13]. Zanfir et al. [13] suggested to learn an average skeleton for each dataset and constrained all skeletons limbs to have the same size of the average skeleton limbs. This normalization approach heavily depends on a specific dataset and makes its adoption in real-world applications difficult. Here, the euclidean normalization of each segment is proposed (without imposing an average length). Hence, we obtain skeletons with unitary segments (4). Each segment is normalized successively starting with the root (hip joint) and moving gradually to the connected segments. This approach has the advantage of preserving the skeleton angles. Its performance will be proved in Section IV.

$$p^{\mathrm{norm}}(t_k) = [p_1^{\mathrm{norm}}(t_k), p_2^{\mathrm{norm}}(t_k), ..., p_{\mathrm{n}}^{\mathrm{norm}}(t_k)] \tag{4}$$

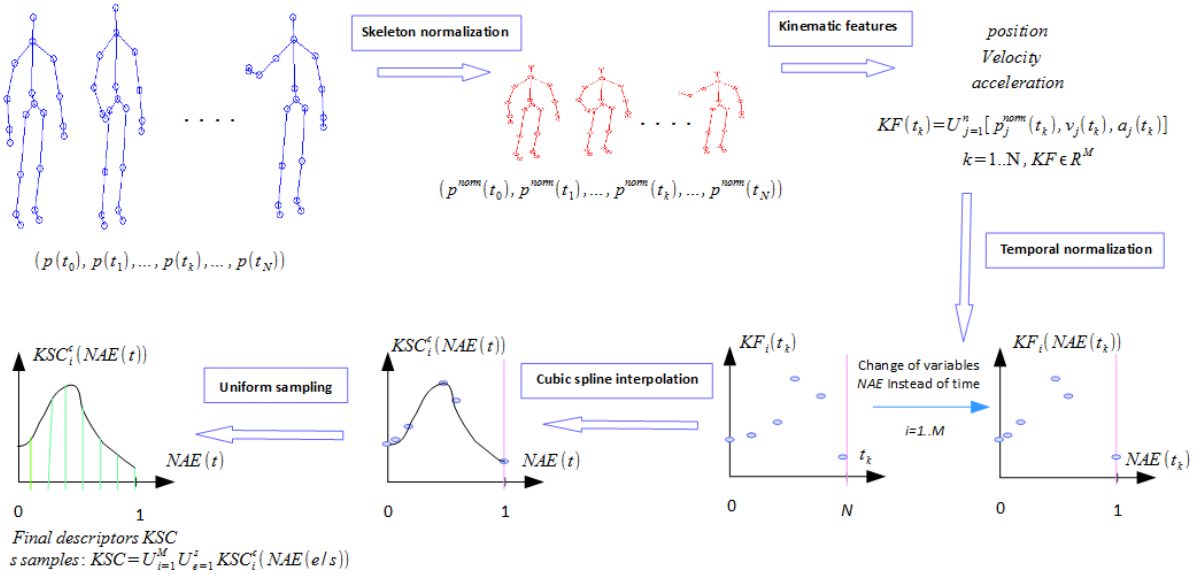

Fig. 1. An overview of our approach: it describes the different processes used to build Kinematic Spline Curves (KSC). The first step (skeleton normalization) allows re-sizing skeleton to reduce the effect of anthropometric variability. Using the obtained normalized joint positions, the skeleton joint velocities and skeleton joint accelerations are calculated. Thus, KF represents the concatenation of the three kinematic values. To make the features invariant to execution rate variability, t is replaced by NAE and each component of KF is interpolated using a cubic spline algorithm. Therefore, M functions KSC^c defined on $[0, 1]$ are obtained. The final step represents the uniform sampling of KSC^c which allows us to construct the final descriptor KSC.

B. Kinematic Features (KF)

In Section II, it has been mentioned that kinematic values such as joint positions, joint velocities and joint accelerations represent an interesting way to describe human motion. Equations (5) and (6) respectively describe the computation of joint velocities and joint accelerations from the discrete information of joint positions [13].

$$v(t_k) = p^{\text{norm}}(t_k + 1) - p^{\text{norm}}(t_k - 1) \tag{5}$$

$$a(t_k) = p^{\text{norm}}(t_k + 2) + p^{\text{norm}}(t_k - 2) - 2 \times p^{\text{norm}}(t_k) \tag{6}$$

Thus, the Kinematic Features (KF) result from the concatenation of Normalized position $p^{\text{norm}}(t)$, velocity $v(t)$ and acceleration $a(t)$ (7).

$$KF(t_k) = [p^{\text{norm}}(t_k), v(t_k), a(t_k))] \tag{7}$$

For each frame, KF are computed. KF vector dimension is equal to $M = 9 \times n$.

C. Temporal Normalization (TN): a novel NAE-based approach

Temporal variability is mainly due to execution rate variability. Indeed, changeable action duration as well as different distribution of motion make action recognition a very challenging task. Actions are performed in different time slices with different velocity variations and are consequently difficult to compare. This is why temporal normalization represents an important step to include. We introduce a fast temporal normalization based on Normalized Accumulated kinetic Energy (NAE). We define the NAE at an instant t as the ratio $\Sigma(t)$ between the kinetic energy $E^{\text{acc}}(t)$ consumed by the human body until t and the total kinetic energy E^{total} consumed by the human body on the whole video composed of N frames. Equation (8) depicts this NAE term where $E(t)$ represents the kinetic energy consumed by the human body at an instant t.

$$\Sigma(t) = \frac{E^{\text{acc}}(t)}{E^{\text{total}}} = \frac{\sum_{c1=1}^{t} E(c1)}{\sum_{c2=1}^{N} E(c2)} \tag{8}$$

The idea is to interpolate the KF components according to NAE, instead of time. In opposition to the time variable, the NAE variable increases when the velocity of joints increases and consequently when the displacement quantity increases as well. If there is no motion, NAE does not increase. We use the normalization and the accumulation of energy for two essential reasons. First, the normalization of the energy guarantees that actions are expressed in the same range (varying between 0 and 1). Second, the accumulation allows obtaining a growing variable which ensures a coherent interpolation. Figure 2 illustrates the interest of the NAE-based temporal normalization.

Skeletons can be considered as a set of n points where each point corresponds to a joint. Many previous papers [14], [15], [16] proposed to express the kinetic energy of the human skeleton $E(t)$, at an instant t, as described by equation (9) where n represents the number of joints, m_i the mass of the joint i and V_i its velocity. Since the skeleton joints are fictive, they are assumed to have a unitary mass in this paper.

$$E(t) = \sum_{i=1}^{n} \frac{1}{2} m_i V_i^2(t) = \sum_{i=1}^{n} \frac{1}{2} V_i^2(t) \tag{9}$$

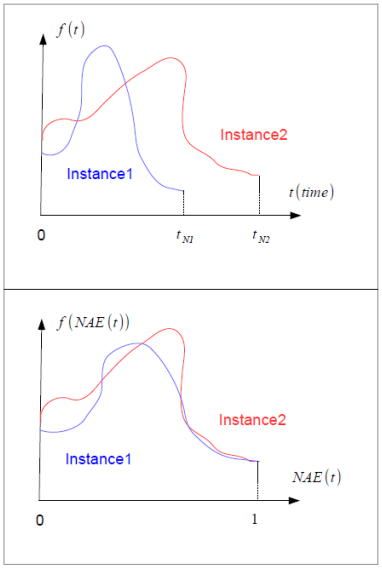

Fig. 2. Illustration of the temporal normalization role by visualizing a joint component trajectory $f(t)$. $f(t)$ represents the x-coordinates of a joint. Here, two instances of a same action are considered (Instance1 and Instance2). Top: the joint component trajectories are plotted as functions of time. We notice that the two trajectories are expressed in different time slices ($t_{N1} \neq t_{N2}$). Bottom: After the NAE-based normalization, we notice that both trajectories are defined in the same range $[0, 1]$. Also, it is important to notice that the two trajectories representing the same action type are more similar.

To calculate the kinetic energy, the instantaneous velocity v is generally used ($V = v$ in (5)). Nevertheless, joint trajectories include sometimes slight oscillations due to undesired motion or to noise (caused by the RGB-D cameras or by the subject himself). These oscillations participate wrongly to increase the accumulated energy term, and make the energy calculation biased. The smooth filtering of joints is not considered relevant because of its parametric nature which makes it hardly adaptable to real-world applications. Therefore, we propose to use a second kind of velocity, the average velocity $V = v^{\mathrm{m}}$ (10). This term reduces the influence of oscillations thanks to the algebraic sum of instantaneous velocity.

$$v^{\mathrm{m}}(t_k) = \frac{1}{t_k}\sum_{l=1}^{k} v(t_l) \tag{10}$$

Thus, the kinetic energy is computed following the equation (11).

$$E(t_k) = \sum_{j=1}^{n} \frac{1}{2}(v_{\mathrm{j}}^{m}(t_k))^2 \tag{11}$$

Finally, the kinetic energy E described in Equation (11) allows the calculation of the NAE term Σ(t) as described in the introduction by the equation (8). Thanks to a change of variables (NAE instead of time), the KF are expressed as variables depending on NAE (12).

$$KF(\Sigma(t_k)) = [p^{\mathrm{norm}}(\Sigma(t_k)), v(\Sigma(t_k)), a(\Sigma(t_k))] \tag{12}$$

Hence, the KF extracted from any action vary in a known range $[0, 1]$. Nonetheless, the discrete KF of each instance are associated to different amounts of NAE. Thus, an interpolation is needed to make actions comparable.

D. Cubic spline interpolation of Kinematic Features (KF)

Assuming continuity of human action kinematics, we propose to interpolate KF components depending on NAE as described in (12). For this purpose, the cubic spline interpolation have been chosen because it connects the points using polynomials of third degree. Third degree polynomials present a maximum of one inflexion point and allow obtaining realistic curves (enough variations, contrary to first or second degree polynomials, with limited oscillations, contrary to polynomials with more than three degree). Indeed, oscillations increase with the polynomial range. Using the discrete information of each KF component, we obtain continuous functions depending on NAE, as described in equation (13), where $Spline$ refers to the cubic spline operator. We recall that M represents the dimension of KF.

$$KSC_i^c(\Sigma(t)) = Spline(KF_i(\Sigma(t_k))_{k=1..N}) \quad \forall i = 1...M \tag{13}$$

E. Uniform sampling

Lastly, a periodic sampling is performed in order to obtain same-size descriptors and to express their components according to the same amounts of NAE. The choice of the number of samples s will be discussed in Section IV. Hence, Equation (14) depicts the calculation of the final descriptor KSC. The size of KSC descriptor is equal to $9 * n * s$.

$$KSC = \cup_{i=1..M} \cup_{e=1..s} KSC_i^c(\Sigma(\frac{e}{s})) \tag{14}$$

Algorithm 1 summarizes the different steps of KSC descriptor computing.

Algorithm 1: Computation of KSC

Input : skeleton sequence $(p_j(t_k))_{1\leq j\leq n, 1\leq k\leq N}$
Output: KSC

Normalize Skeleton $(p_j^{norm}(t_k))_{1\leq j\leq n, 1\leq k\leq N}$ (4)
Compute Kinematic Features $(KF_i(t_k))_{1\leq i\leq M}$ (12)
Compute $(\Sigma(t_k))_{1\leq k\leq N}$ (8)
for $i \leftarrow 1$ **to** M **do**
  Interpolation: $KSC_i^c(t) := Spline(KF_i(t_k))_{1\leq k\leq N}$
end
Uniform sampling with a sampling rate s:
$KSC := \cup_{i=1..M} \cup_{e=1..s} KSC_i^c(\Sigma(\frac{e}{s}))$

F. Action recognition via linear Support Vector Machine

To recognize actions, a linear SVM classifier provided by libSVM library [17] is trained using the KSC descriptors. Our choice has been motivated by the low computational cost of linear kernel classifiers compared to non-linear ones [18].

Descriptor	MSRAction3D (%)
HOJ3D [10]	78.97
EigenJoints [11]	82.33
Actionlet [19]	88.20
FV skeletal quads [8]	89.86
LARP [7]	**92.46**

TABLE I
ACCURACY OF RECOGNITION ON MSRACTION3D: THE VALUES OF EARLIER METHODS ARE TAKEN FROM THE STATE-OF-THE-ART (DIFFERENT CROSS-SPLITTINGS ARE USED)

Descriptor	AS1(%)	AS2(%)	AS3(%)	Overall(%)	Time(s)
JP [7]	82.86	68.75	83.73	78.44	0.58
RJP [7]	81.90	71.43	88.29	80.53	2.15
Q [7]	66.67	59.82	71.48	67.99	1.33
LARP [7]	83.81	84.82	92.73	87.14	17.61
KSC(ours)	**86.67**	**89.29**	**96.40**	**90.78**	**0.092**

TABLE II
ACCURACY OF RECOGNITION AND EXECUTION TIME PER DESCRIPTOR(S) ON MSRACTION3D: AS1, AS2 AND AS3 REPRESENTS THE THREE GROUPS PROPOSED IN THE EXPERIMENTATION PROTOCOL OF [6]

Descriptor	Accuracy (%)
Random Forrest [21]	87.90
LARP [7]	**97.08**
KSC (ours)	95.00

TABLE III
ACCURACY RECOGNITION ON UTKINECT DATASET: THE VALUES OF EARLIER METHODS ARE TAKEN FROM THE STATE-OF-THE-ART

IV. EXPERIMENTAL EVALUATION

We evaluate our method on two well known RGB-D based human action recognition benchmarks, namely MSRAction3D [6] and UTKinect [10] . To ensure fair comparison, we report only the methods based on skeleton representations.

A. MSRAction3D Dataset

MSRAction3D dataset includes 20 different actions, performed by 10 different subjects, 2 or 3 times. It provides 2 modalities: skeleton joints and depth maps. This dataset is challenging because of its very similar actions.

Table I reports the recognition accuracy of state-of-the art methods. However, as mentioned in [20], the experimental settings are different from a paper to another making a fair comparison difficult. In addition to that, numerous of earlier papers do not evaluate their methods in terms of computational latency. For these reasons, we propose to evaluate some available descriptors such as Joint Positions (JP) [7] , Relative Joint Positions (RJP) [7] , Quaternions (Q) [7] and finally Lie Algebra Relative Pairs (LARP) [7] on MSRAction3D in terms of accuracy and computational latency with the respect of the same experimental parameters.

To compare between methods in terms of computational latency, the mean execution time per descriptor is reported. It represents the average time necessary to compute a descriptor (the descriptor represents the feature vector which describes an action in a whole video). It is important to specify that calculations were run on the same computer (*Dell Inspiron N5010* with *intel Core i7*, *Windows 7* and *4GB RAM*).

For all experiments on MSRAction3D, the same parameters of [6] are used. The dataset is divided into three groups (AS1, AS2, AS3). Thus, the classification is realized on each group separately. A cross-splitting is carried out to separate the data in training and testing samples. The actions performed by the subjects 1, 3, 5, 7, 9 are used for training, while the actions performed by the other subjects are used for testing. Table II shows that our method outperforms other approaches in terms of recognition accuracy and computational latency.

B. UTKinect Dataset

UTKinect dataset contains 10 actions which are performed twice by 10 different subjects. RGB images, depth images and skeleton joints are provided. A significant intra-class variation makes this dataset very challenging.

We chose the same settings used in [7], where training and testing data are also separated following a cross-splitting approach. The actions generated by the subjects 1,3,5,7,9 are used for training while the rest of actions are used for testing.

Table IV and Table I show that our method presents good performances on UTKinect. Nevertheless, we notice that LARP[7] presents a better accuracy on UTKinect. This is why it is important to mention that according to Table II, LARP is 191 times slower to compute which makes it unsuitable for applications requiring low computational latency.

C. Effectiveness of Spatial Normalization (SN) and Temporal Normalization (TN)

In this subsection, the effectiveness of Spatial Normalization (SN) and Temporal Normalization (TN) is shown. Each line of Table V reports the accuracy of recognition without the use of the proposed SN or TN.

SN contributes to improve the accuracy with an increase of around 6% for MSRAction3D and of around 10% for UTKinect. Also, the superiority of unitary euclidean normalization is proved by combining our method with the SN of [13]. The accuracy becomes lower with only 83.26% on MSRAction3D and 85% on UTKinect.

According to Table V, the contribution of TN is fundamental. Without the use of TN, the accuracy decreases from 90.78% to 81.26% on MSRAction 3D and from 95% to 81% on UTKinect . On the other hand, it is important to highlight

Process	MSRAction3D	UTKinect
Spatial Normalization	0.022	0.016
Descriptor computing	0.07	0.064
Classification	0.008	0.002
Total	0.1	0.082

TABLE IV
EXECUTION TIME (S) OF EACH PROCESS PER DESCRIPTOR ON THE THREE BENCHMARKS

Deleted Process	MSRAction3D (%)	UTKinect (%)
Nothing	**90.78**	**95.00**
without S.N.	83.83	85.00
without T.N.	81.26	81.00

TABLE V
EFFECT OF EACH PROCESS ON THE ACCURACY OF RECOGNITION

Kinematics	MSRAction3D (%)	UTKinect (%)
P+V+A	**90.78**	**95.00**
P+V	87.28	90.00
P	86.63	91.00
V	83.90	81.00
A	81.47	82.00

TABLE VI
EFFECT OF EACH KINEMATIC COMPONENT ON THE ACCURACY OF RECOGNITION

s	5 (%)	10 (%)	15 (%)	20 (%)	25 (%)
MSRAction3D	87.22	86.9	88.74	**90.78**	88.74
UTKinect	92.00	93.00	**95.00**	93.00	93.00

TABLE VII
EFFECT OF s THE NUMBER OF SAMPLES s ON THE ACCURACY OF RECOGNITION

the predominant role of the average velocity term. Indeed, the accuracy decreases to 67.58% on MSRAction3D and to 86% on UTKinect with the use of instantaneous velocity.

D. Benefits of kinematic features

Table VI reports the importance of each kinematic value. It demonstrates that position is the most discriminative term. It could be due to an error increase caused by the derivations. However, the combination of the three kinematic values presents the most accurate results.

E. The influence of the parameter s (number of samples)

Table VII reports the influence of the parameter s on the accuracy. For each dataset, the parameter s is fixed according to the best amount of accuracy ($s = 20$ for MSRAction3D and $s = 15$ for UTKinect). Nevertheless, we notice that our method is robust to parameter variation. For the values tested in Table VII, we observe a decrease up to 3% compared with the highest score of accuracy.

V. CONCLUSION AND FUTURE WORK

In this paper, a novel descriptor for action recognition has been introduced. This descriptor is computed thanks to the interpolation of joint kinematics. To overcome the anthropometric variability, a skeleton normalization is extended inspired by the work of zanfir et al. [13]. On the other hand, a novel NAE-based temporal normalization is proposed in order to alleviate the effect of execution rate variability. According to the experiments, our method outperforms other skeleton-based approaches in terms of accuracy of recognition and computational latency. Therefore, this fast and accurate human motion representation may be very useful in real-world applications.

However, our method presents some limitations. For the time being, it is still not suited for online settings which limits the use of KSC to offline systems. The extension of a partial alignment algorithm such as DTW might be an interesting possibility making an early recognition possible. In future work, we are planning to raise the issue of observational latency and online mode by extending our method. This extension will explore the possibility of putting in place a real-time action recognition system.

REFERENCES

[1] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore, "Real-time human pose recognition in parts from single depth images," *Communications of the ACM*, vol. 56, no. 1, pp. 116–124, 2013.

[2] R. Poppe, "A survey on vision-based human action recognition," *Image and vision computing*, vol. 28, no. 6, pp. 976–990, 2010.

[3] G. T. Papadopoulos, A. Axenopoulos, and P. Daras, "Real-time skeleton-tracking-based human action recognition using kinect data," in *MultiMedia Modeling*. Springer, 2014, pp. 473–483.

[4] E. Ghorbel, R. Boutteau, J. Boonaert, X. Savatier, and S. Lecoeuche, "3D real-time human action recognition using a spline interpolation approach," in *Image Processing Theory Tools and Applications (IPTA), 2015 5th International Conference on*. IEEE, 2015.

[5] M. Hammouche, E. Ghorbel, A. Fleury, and S. Ambellouis, "Toward a real time view-invariant 3d action recognition," in *VISAPP*, 2016.

[6] W. Li, Z. Zhang, and Z. Liu, "Action recognition based on a bag of 3D points," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*. IEEE, 2010, pp. 9–14.

[7] R. Vemulapalli, F. Arrate, and R. Chellappa, "Human action recognition by representing 3D skeletons as points in a Lie group," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*. IEEE, 2014, pp. 588–595.

[8] G. Evangelidis, G. Singh, and R. Horaud, "Skeletal quads: Human action recognition using joint quadruples," in *ICPR 2014-International Conference on Pattern Recognition*, 2014.

[9] C. Ellis, S. Z. Masood, M. F. Tappen, J. J. Laviola Jr, and R. Sukthankar, "Exploring the trade-off between accuracy and observational latency in action recognition," *International Journal of Computer Vision*, vol. 101, no. 3, pp. 420–436, 2013.

[10] L. Xia, C.-C. Chen, and J. Aggarwal, "View invariant human action recognition using histograms of 3D joints," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*. IEEE, 2012, pp. 20–27.

[11] X. Yang and Y. Tian, "Eigenjoints-based action recognition using naive-bayes-nearest-neighbor," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*. IEEE, 2012, pp. 14–19.

[12] G. Johansson, "Visual perception of biological motion and a model for its analysis," *Attention, Perception, & Psychophysics*, vol. 14, no. 2, pp. 201–211, 1973.

[13] M. Zanfir, M. Leordeanu, and C. Sminchisescu, "The moving pose: An efficient 3D kinematics descriptor for low-latency action recognition and detection," in *Computer Vision (ICCV), 2013 IEEE International Conference on*. IEEE, 2013, pp. 2752–2759.

[14] K. Onuma, C. Faloutsos, and J. K. Hodgins, "FMdistance: A fast and effective distance function for motion capture data," *Short Papers Proceedings of EUROGRAPHICS*, vol. 2, 2008.

[15] J. Shan and S. Akella, "3D human action segmentation and recognition using pose kinetic energy," in *Advanced Robotics and its Social Impacts (ARSO), 2014 IEEE Workshop on*. IEEE, 2014, pp. 69–75.

[16] Y. Shi and Y. Wang, "A local feature descriptor based on energy information for human activity recognition," in *Advanced Intelligent Computing Theories and Applications*. Springer, 2015, pp. 311–317.

[17] C.-C. Chang and C.-J. Lin, "Libsvm: A library for Support Vector Machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, p. 27, 2011.

[18] S. R. Fanello, I. Gori, G. Metta, and F. Odone, "Keep it simple and sparse: Real-time action recognition," *The Journal of Machine Learning Research*, vol. 14, no. 1, pp. 2617–2640, 2013.

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

[20] J. R. Padilla-López, A. A. Chaaraoui, and F. Flórez-Revuelta, "A discussion on the validation tests employed to compare human action recognition methods using the msr action3d dataset," *arXiv preprint arXiv:1407.7390*, 2014.

[21] Y. Zhu, W. Chen, and G. Guo, "Fusing spatiotemporal features and joints for 3D action recognition," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2013 IEEE Conference on*. IEEE, 2013, pp. 486–491.